ENDPOINTS AND INTERDEPENDENCIES IN INTERNET OF THINGS RESIDUAL ARTIFACTS: MEASUREMENTS, ANALYSES, AND INSIGHTS INTO DEFENSES

by

JINCHUN CHOI
B.S. INHA University, 2011
M.S. INHA University, 2014

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2020

Major Professor: David Mohaisen

ABSTRACT

The usage of Internet of Things (IoT) devices is growing fast. Moreover, the lack of security measures among the IoT devices and their persistent online connection give adversaries an opportunity to exploit them for multiple types of attacks, such as distributed denial-of-service (DDoS). To understand the risks of IoT devices, we analyze IoT malware from an endpoint standpoint. We investigate the relationship between endpoints infected and attacked by IoT malware, and gain insights into the underlying dynamics in the malware ecosystem. We observe the affinities and different patterns among endpoints. Towards this, we reverse-engineer 2,423 IoT malware samples and extract IP addresses from them. We further gather information about these endpoints from Internet-wide scans. For masked IP addresses, we examine their network distribution, with networks accumulating more than 100 million endpoints. Moreover, we conduct a network penetration analysis, leveraging information such as active ports, vulnerabilities, and organizations. We discover the possibility of ports being an entry point of attack and observe the low presence of vulnerable services in dropzones. Our analysis shows the tolerance of organizations towards endpoints with malicious intent. To understand the dependencies among malware, we highlight dropzone characteristics including spatial, network, and organizational affinities. Towards the analysis of dropzones' interdependencies and dynamics, we identify dropzones chains. In particular, we identify 56 unique chains, which unveil coordination among different malware families. Our further analysis of chains suggests a centrality-based defense and monitoring mechanism to limit malware propagation. Finally, we propose a defense based on the observed measures, such as the blocked/blacklisted IP addresses or ports. In particular, we investigate network-level and country-level defenses, by blocking a list of ports that are not commonly used by benign applications, and study the underlying issues and possible solutions of such a defense.

This dissertation is dedicated to my wife Eunah, and my dear parents who supported and encouraged me to complete this achievement.

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my advisor and doctoral committee chair, Dr. David Mohaisen, who advised and encouraged me during my research journey. I would also like to thank the members of my dissertation committee for their valuable feedback and support: Dr. Amro Awad, Dr. Wei Zhang, and Dr. Sung Choi Yoo. Also, I want to thank my colleagues in the Security Analytics Lab (SEAL), Dr. DaeHun Nyang, who is my co-advisor in the Information Security Research Lab (ISRL) at Inha University, and my teammates at Inha for the support. Last but not least, I appreciate the Global Research Laboratory (GRL) project for supporting my study; this work was supported in part by NRF (National Research Foundation of Korea) under Grant 2016K1A1A2912757.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

IoT has reshaped the way in which people, businesses, and even cities interact with their environment through Internet-connected devices. Whether it is remotely unlocking a smart door lock [86] or accessing your home appliances from anywhere in the world. There is no doubt that IoT devices have benefited the global economy and made our lives more efficient. The number of IoT devices soaring into the tens of billions [28, 56], and 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 [68]. However, potential adversaries have set their sights on these devices knowing that they are always connected and online. The majority of these IoT devices are at a high risk to the new threats due to the lack of security awareness among consumers and the lack of consensus on security standards among the IoT industry [62]. Also, given the low computational power of the IoT devices, a single device would not be sufficient to carry out an attack [22]. However, exploiting IoT's scale, adversaries form a network of bots or intermediary targets, large enough to launch an attack of substantial magnitude. 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 [52]. 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 [17, 50, 63, 81].

With the vulnerability of the IoT devices against adversaries, such devices can be easily exploited and controlled. For instance, Bashlite (also known as Gafgyt, LizardStresser, Lizkebab, Qbot, and Torlus) is a malware family that exploits default login IDs and passwords to propagate and infect targets [24]. Based on this infection, the Bashlite malware family made a large botnet that is capable of launching large DDoS attacks [55]. 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 [53]. 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 [22]. Among other incidents, Mirai's attacks temporarily disrupted "Krebs on Security" (security blog) [51], OVH (cloud computing company) [42], and Dyn (domain service company) [48]. 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 [64].

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 [53]. Understanding a phenomenon in behavior is important for multiple reasons. First, such 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 competition would necessarily require appropriate analysis modalities, and such 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 the majority of infected hosts in an IoT botnet makes an excellent candidate for a botnet takedown.

Considering the landscape and the risks IoT devices possess, the malware authors can exploit the vulnerability to either attack them or hire them as intermediary targets for a future large scale attack. Reckoning that the malware sources, command and control (C2) servers, the intermediary targets, and the victim must be connected to the Internet in the attack scenario, which makes it important to study these endpoints.

1.1 Motivation

The focus of this work is to conduct a comprehensive analysis of endpoints in IoT malware, including sources, C2's, intermediary targets, and victims. In particular, we formulate questions that we answer through a data-driven analysis to find the correlations between the endpoints of the dropzones and targets, addressing the following:

Dropzone-Target Inter-Relationships. Since malware associated with certain dropzones point to specific target IP addresses, could these IP addresses be similar or identical to the addresses of targets in other dropzones? To answer this, we reverse-engineered and analyzed the malware's disassembly to extract all target IP addresses for each dropzone.

Geographical Analysis. What are the characteristics of the areas where the dropzones are located? How does this affect the distribution of dropzones and targets? For that, we analyze the distributions of the distance between the dropzones and their targets, and look at these distributions from various perspectives at the country and state level.

Attack Exposure. How exposed are the IP addresses in the target's network address space? Towards this, we analyze the targets and look for vulnerabilities in the services that they use. For the masked targets, we analyze the network space and examine their up-to-date susceptibility.

Network Penetration Analysis. What are the vulnerable services being used for both dropzones and target IP addresses? Which organizations own these IP addresses and how do they affect the dropzone-target relationships? We analyze the attributes of dropzone and target IP addresses such as their active network ports, owning organization, and known vulnerabilities from Internet-wide scanners; Shodan and Censys [3].

To this end, we analyze the characteristics of the IP addresses, using a search engine that retrieves the characteristics of the IP addresses. The features we collect are the ports that are active on the endpoint devices, known vulnerabilities related to the addresses, and organizations that operate the IP addresses. In this work, we set questions that we answer through conducting experiments and network and endpoint analysis.

1.2 Approaches

Analyze IoT Malware Ecosystem. In this work, we extracted endpoints from IoT malware samples by reverse-engineering those samples and perform a data-driven study to analyze their different traces, such as geographical affinities, organizations, open ports, and their susceptibility to attacks. We also try to understand the pattern shared among victims by different malware.

Network Penetration Analysis. We focused on the network penetration analysis, which is the characteristics contained in the IP address, leveraged the information gathered from a search engine for Internet-connected devices, such as active ports, vulnerabilities and organizational information.

The vulnerabilities in the dropzones reflect that the attackers override their authentication status and then utilize the OpenSSH vulnerabilities to gain access to the device. Our analysis also sheds light on the tolerance of organizations towards the endpoints with malicious intent. The tug of war between the profit and the loss of trust among the users (prospective domain buyers) deserve the community's focus.

Malware Behavior Analysis. 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 IP addresses 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.

1.3 Roadmap

The rest of this dissertation is organized as follows. In chapter 2, we introduce a unified related work based borrowing contents from several studies by the dissertation author. In chapter 3, we introduce our analysis of the IoT ecosystem. In chapter 4, we introduce our analysis of the IoT malware network penetration analysis. Both of these chapters are based on our work published in ACM HoTWot 2019 [26]. In chapter 5, we introduce our analysis of the dynamics and interdepenendeices in IoT malware through the lenses of dropzone-target chains, which is based on our work published in IEEE IDSC 2019 [25]. In chapter 6, we introduce our insight into defenses. A brief discussion is introduced in chapter 7, followed by concluding remarks in chapter 8.

CHAPTER 2: RELATED WORK

2.1 IoT Malware Ecosystem Analysis

Recent studies related to IoT malware in the past few years have primarily focused on classifying IoT Malware. Su *et al.* [78] proposed a light-weight classification system based on image recognition that was tested on real IoT malware samples collected by IoTPOT [66], one of the first honeypots specifically for IoT threats. Using a dataset of 500 malware samples comprised of multiple families (including the Mirai botnet and Linux.Gafgyt) and benign samples from Ubuntu 16.04.3 system files, they converted each sample into 64x64 gray-scale images that were fed into a convolutional neural network achieving an average accuracy of 94%. Abusnaina *et al.* [15] presented Graph Embedding and Augmentation (GEA), a method to generate adversarial IoT software. With their approach, they successfully achieved a high misclassification rate in Control Flow Graph (CFG)-based features and deep learning network detection method, while ensuring that the generated software is executable.

Cozzi *et al.* [27] investigated the different patterns and trends among the Linux malware in depth. Sivanathan *et al.* [74] analyzed the traffic of smart IoT environments gathered over a period of 3 weeks to characterize different traffic attributes. They differentiate IoT traffic from other traffic as well as identifying IoT devices with an accuracy of 95%. Alasmary *et al.* [18–20] studied methods of malware detection based on graph-based features from CFGs. They show that CFGs, even with smaller size than similar software, can be powerful in identifying IoT applications, including distinguishing between benign and malicious ones. Related work on analyzing malware of other systems and evaluating the accuracy of their detection using various modalities are explored in [59, 60, 72].

West and Mohaisen [83] used 28,000 expert-labeled endpoints extracted from ≈100K malware binaries for binary threat classification with an accuracy of 99.4%. The endpoints were extracted using dynamic execution of malware in a sandboxed environment. Ouellette *et al.* [65] used deep learning to detect malicious endpoints. They used the features from obfuscated malware samples to feed to the classifier that performing classification on the cloud. Rafique and Caballero [70] used the network signatures from executing malware binaries to cluster them into families. Although limited, prior works have looked into investigating Linux malware, and the malware endpoints have not received the attention. Antonakakis *et al.* [22] analyzed the relationship between domains that were extracted by reverse-engineering the Mirai malware. With this work, we push towards filling the gap.

2.2 Network Penetration Analysis

Studies have also utilized a combination of Shodan, an Internet-wide search engine for IoT devices [12], and known vulnerability databases to realize the potential risks inherent to Internet-connected devices. For example, Genge and Enachescu [41] proposed ShoVAT (Shodan-based Vulnerability Assessment Tool) and collected IoT device information such as open ports, when they were scanned, banner data, and their operating system through the Shodan API. They then used this information to confirm their identities in the NVD and revealed 3,922 known vulnerabilities among 1,501 services in 12 different institutions. Formby *et al.* [40] security challenges in the existing Industrial Control Systems (ICS) and address them by leveraging fingerprinting methods. Feng *et al.* [39] proposed a rule to discover and annotate IoT devices.

2.3 Dynamics and Interdependencies in IoT Botnets

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 [84]. 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. Kendall *et al.* [49] 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.* [61] examined the limitations of the static analysis in the detection of malicious code. Soliman *et al.* [75] 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.

Several IoT malware have the capability to launch DDoS attacks [30]. Mirai is one of the notorious IoT malware that is targeting vulnerable IoT devices such as Digital Video Recorders (DVRs), security cameras, routers [22]. 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 [82]. 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 [73]. Similarly, Ceron *et al.* [24] 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.* [31] 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.

To the best of our knowledge, there is no recent work that analyzes the relationships between the endpoints of IoT malware dropzones and their target devices. With that said, the closest study to our work is by Holz *et al.* [45] who presented one of the first empirical studies of malware and dropzones. Specifically, they focused on keyloggers and harvested data from dropzones which contained stolen credentials. Since keyloggers typically contact dropzones upon execution (to obtain a configuration file), the authors managed to successfully obtain the locations of several dropzones from several Autonomous Systems and countries, shown to be Russia and the US, among others.

CHAPTER 3: IOT MALWARE ECOSYSTEM IN THE WILD: ANALYSIS AND EXPOSURES[1]

3.1 Motivation

We perform IP centric analysis and network centric analysis. In the IP centric analysis, we analyze the dropzone-target inter-relationships. Specifically, we investigate the target IP addresses among different dropzone IP addresses. Also, we perform a geographical analysis of the dropzones and targets. Towards this, we analyze the locations of dropzones and their target IP addresses. In the network centric analysis, we analyze the attack exposure of networks and IP addresses. For masked target endpoints, we examine the entire network and study the network devices and their exposure to risk.

The goal of this work is to analyze the affinities between IoT malware endpoints. Towards this goal, we make the following contributions:

- We analyze the dropzone-target inter-relationships. Specifically, we investigate the target IP addresses among different dropzone IP addresses.
- We perform a geographical analysis of the dropzones and targets. Towards this, we analyze the locations of dropzones and their target IP addresses.
- We analyze the attack exposure of networks and IP addresses. For the masked target endpoints, we examine the entire network and study the network devices and their exposure to risk.

[1] This content was reproduced from the following article: J. Choi*, A. Anwar*, H. Alasmary, J. Spaulding, D. Nyang and A. Mohaisen, "IoT Malware Ecosystem in the Wild: A Glimpse into Analysis and Exposures". The Second ACM/IEEE Workshop on Hot Topics on Web of Things, HotWoT, pp. 413-418, Arlington, VA, USA, 2019. *Equal Contributors. The copyright form for this article is included in the appendix.

Table 3.1: Distribution of malware by family. DZ - Dropzone.

Target family	Count	Pct.	DZ family	Count	Pct.
Gafgyt	930	95.58	Gafgyt	2,294	98.96
Tsunami	39	4.01	Tsunami	24	1.04
SINGLETON	3	0.31	-	-	-
Lightaidra	1	0.10	-	-	-

3.2 Dataset

We describe our dataset and its augmentation towards our goal. We then describe the aims and the objectives of this work. Specifically, we use an IoT malware dataset, perform static analysis on them, and finally use the strings to extract endpoints from them. We obtain our dataset from IoT-POT [66], a honeypot that emulates the Telnet services (later improved to include other services). We obtained a total of 2,423 IoT malware samples, which were graciously given to us by the authors of IoTPOT. The dataset represents four different malware families, labelled by augmenting the results from VirusTotal (VT) and by using AVClass [71]. For malware samples that do not have a decisive family label from the VT results, those malware samples are labeled as SINGLETON. The distribution of malware families can be seen in Table. 3.1.

We reverse-engineer and analyze the malware samples using Radare2 [11], an open-source malware analysis framework. We find strings in the malware binary, especially IP addresses, and classify those addresses by their association with special keywords into two classes: `dropzone` and `target` IP addresses, defined as follows:

- **Dropzone IP.** Adversaries often keep malware binaries in remote servers to distribute them after gaining access to victim devices. These remote servers are identified by `dropzone` IP addresses, controlled and managed by the adversary and used for malware propagation and management. As such, the dropzone IP addresses are associated with wget, HTTP, TFTP, GET or FTP in the residual strings obtained from the malware analysis.

- **Target IP.** To infect victim hosts, malware uses a list of IP addresses, including target devices. We refer to these IP addresses as `targets`. We note that a large number of those target addresses in our analysis are masked. For example, 123.17.%d.%d is one of the target IP address that is masked at /16; the attacker can utilize this address targeting all IPs in the network address space.

Also, we found the internal network addresses (*e.g.,* 192.168.%d.%d), loopback address (*e.g.,* 127.0.0.1) from our target dataset and remove them, since they are irrelevant to our analysis. Also, we note that the Mirai source code contained a list of "don't scan" addresses, including various U.S. Department of Defense (DoD) address blocks, as well as internal addresses [44], which we exclude. Fig. 3.1 shows the dropzone and target in the malware life-cycle, including dropzone setting, victim host (target) compromise, and download of malware from the dropzone to the target.

3.2.1 Data Augmentation

We group the target and dropzone addresses by malware. Since a dropzone can be used by multiple malware, and to help analyze the overall sample-space a dropzone caters to, we cluster the target IPs by each dropzone. Using *UltraTools* [13], a free DNS and domain lookup tool, and *Censys* [3], a search engine for Internet-connected devices, each of the targets and dropzones is augmented with the following information: country, ASN (Autonomous System Number), and location (*e.g.,* latitude and longitude), open ports, etc. We observe some dropzone addresses have no current information, *e.g.,* they are no longer connected to the Internet. This confirms that the dropzones are short-lived—long enough to carry out an attack and short not to be detected. As such, we leverage historical data of those IP addresses from Shodan [12] to determine the necessary data points associated with them.

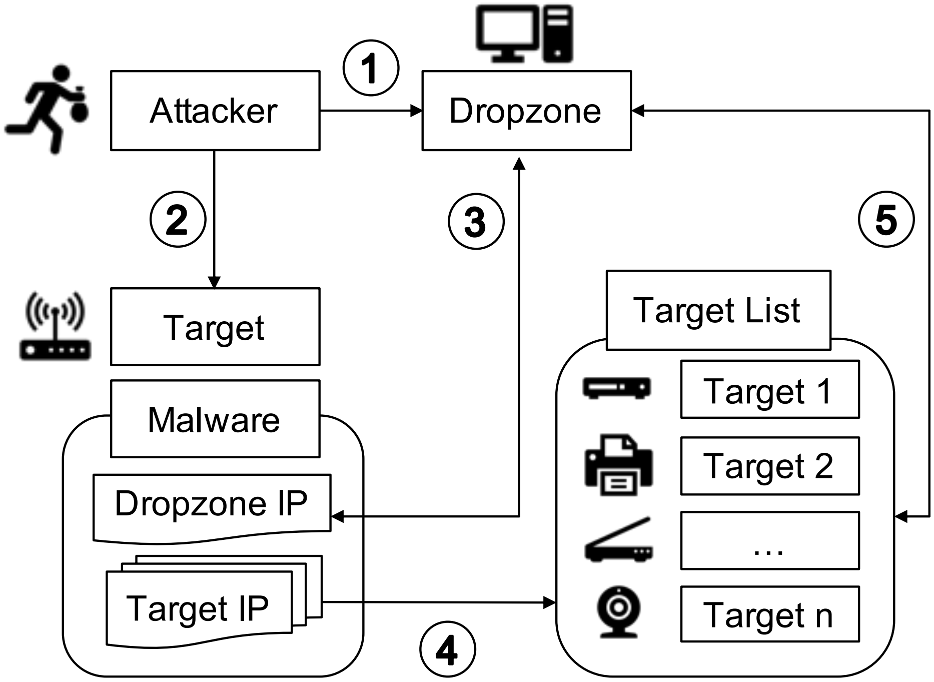


Figure 3.1: An overview of the dropzone and target in the malware life-cycle. ①: The attacker uses pre-configured or infected server (dropzone) for attack propagation. ②: The attacker try to propagate to targets by exploiting vulnerabilities. ③: The malware accesses the dropzone and download scripts and payload to the target. ④: The malware targets potential victim devices in a target IP list. ⑤: The infected target devices repeat the process.

3.2.2 Preliminary Results

The distribution of the target addresses exposes family-level affinities by highlighting what set of addresses is being targeted by different malware. In particular, we observe a total of 106,428 target IP addresses, resulting in 2,211 *unique target* IP addresses associated with 973 malware samples. This makes the analysis of affinities, by understanding what makes these target IP addresses the favorite among malware authors, of paramount importance. The use of dropzone IP addresses by the malware exhibits that the malware shares dropzones among themselves, with some contacting multiple dropzones for commands. In particular, for the dropzones, we find that 877 *unique dropzone* IP addresses are being shared by 2,318 malware samples with 2,407 occurrences. Moreover,

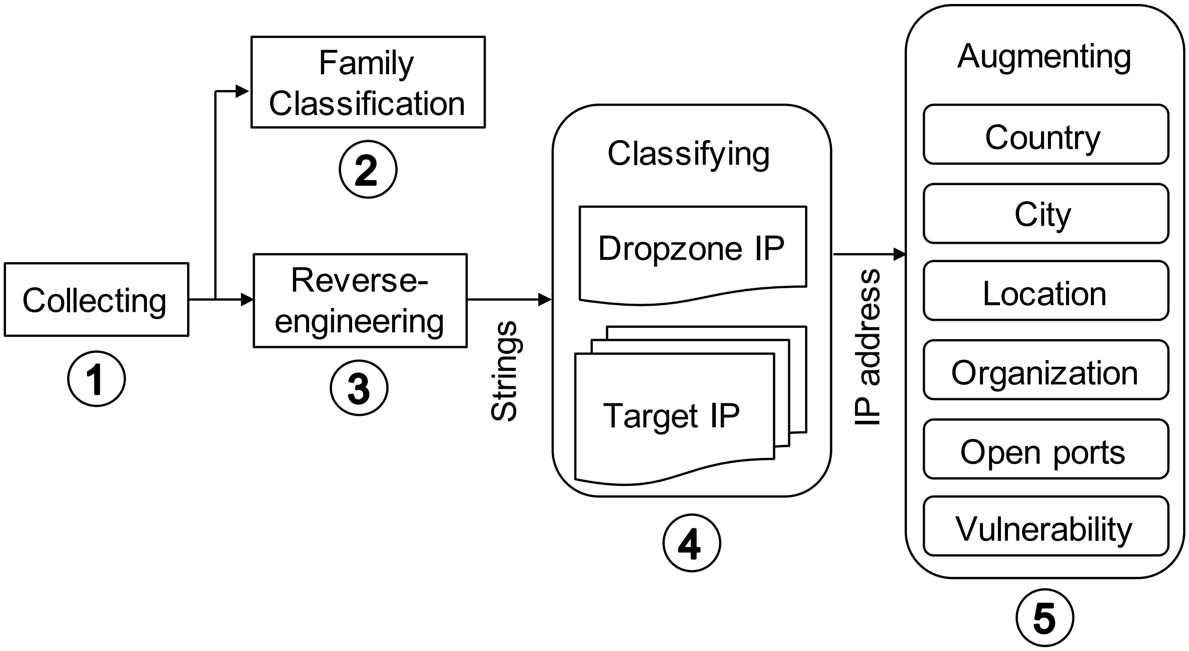


Figure 3.2: Data collection and extraction system. ①: Collecting malware binaries from IoTPOT. ②: Assigning known malware family names to the dataset. ③: Reverse-engineering and extracting strings from binaries. ④: Separating IP addresses based on nearby keywords. ⑤: Augmenting IP addresses with Censys and Shodan data.

while we successfully extract dropzone IP addresses from the majority of malware samples, we can find target IP addresses only in fewer malware samples. This shows a thought pattern of malware authors, *i.e.,* while they share the dropzone IP addresses in the static code, however, they do not reveal the target IP addresses. This can be because they obfuscate this part of the code, employ domain generation algorithms, or use a custom list of IP addresses in the downloaded binary file (*i.e.,* DNS.txt) from a dropzone at run-time, as shown in Fig. 3.3.

We also notice that 40% of the malware samples contain target IP addresses, while 95.66% of them contain dropzones in their strings. We also observe disassembled codes of malware samples that have dropzone but no targets, which is explained either: (i) code-based generation of IP addresses, rather than static IP address listing [21, 23], and (ii) packing.

```
wget \%s −q –O DNS.txt
     || busybox wget \%s –O DNS.txt
     || /bin/busybox wget \%s –O DNS.txt
     || /usr/busybox wget \%s –O DNS.txt
```

Figure 3.3: Retrieving a list of target hosts.

Table 3.2: Top 5 dropzone IPs per the number of targets. Countries include: France (FR), United Kingdom (GB), Canada (CA) and United States (US).

Rk.	Dropzone IP	Country	#Malware	#Targets
1	163.172.104.150	FR	35	9,529
2	145.239.72.250	FR	22	5,632
3	45.76.131.35	GB	17	4,352
4	64.137.253.50	CA	26	3,066
5	198.175.126.89	US	11	2,816

3.3 Dropzone-Target Inter-Relationship

We inspect the dropzone-target relationship, we examine the affinity between the dropzone and the target IP addresses. While ≈77% of the unique target IPs received less than 10 attacks, one unique target IP received 72 attacks. We found one dropzone IP (`50.115.166.193`) that was only associated with 1 malware sample. This malware sample pointed to 1,265 network addresses, which was significantly larger than the average of 121 target IP addresses for a typical malware sample. Also, they are masked network addresses, which means that one target network address can be larger dynamically. Conversely, the dropzone IP (`5.189.171.210`) has 86 associated malware samples, but each of those point to a single target IP address.

Dropzones can be found distributed mainly in North America and Europe. Moreover, through our further analysis we found that the first IP address (`163.172.104.150`) (Table 3.2) is associated with 35 malware samples affecting 9,529 target IPs.

3.3.1 Shared Targets Between Dropzones

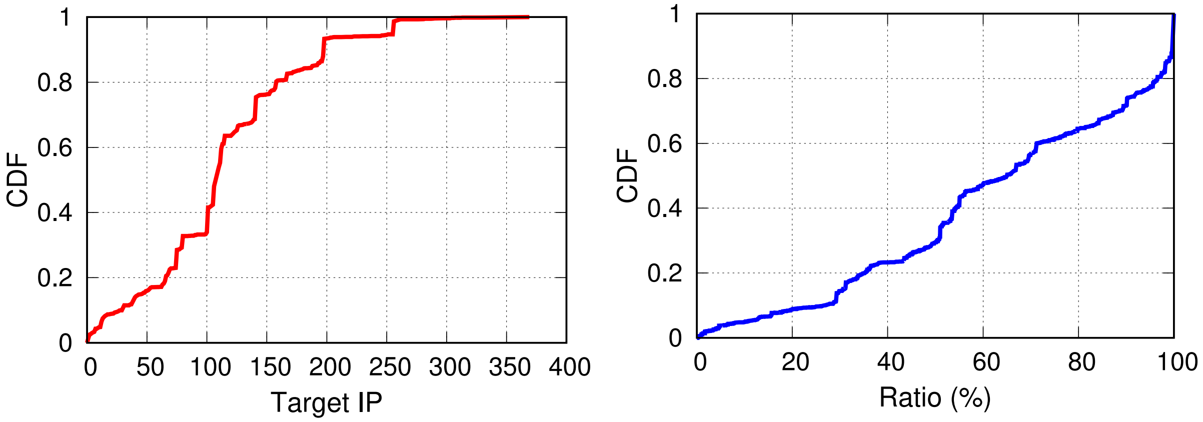
To inspect the shared targets between dropzone IP addresses, we group the dropzone IP addresses and capture the common (overlapping) targets among the dropzones. Since dropzones can be associated with multiple instances of malware, each malware can have its own list of target IP addresses. If we assume that a dropzone has a union of target IPs for each malware belonging to that particular dropzone, we can aggregate all of their target IPs into a larger set of target IPs. We denote U_{dz} as the union of all target IPs for a particular dropzone. To analyze the overlapping target IP addresses and to understand the criteria for choosing target addresses, we compare U_{dz} of each combination of dropzone addresses from a dataset of 877 unique dropzone addresses. Using combinations, we found combinations of

$$\binom{877}{2} = \frac{877!}{2!(877-2)!} = 384,126.$$

Upon removing 365,968 cases that do not have common target IP between them, we reduce the combination to 18,158 dropzone IP pairs. This dataset of 18,158 dropzone IP pairs is a combination of only 247 unique dropzone IP addresses, from the dataset of 877 unique dropzone IP addresses. Also, we find 71 cases that had more than 300 overlapped target IPs, and there are 2,199 cases (12.11%) which are 100% overlapped between dropzones. Overall, we find 6,451 cases (35.53%) in which the overlap is more than 80%.

3.3.2 Summary

It is evident from the results of the above analysis that a large number of targets are being shared between dropzones. If the target IP addresses between different dropzones are matched 100%, it is possible that the attacker obtained the same targets through similar vulnerability analysis (*i.e.,* Shodan) or shared the target list from other attackers through underground communities.



(a) CDF graph of the number of overlapped target IPs. (b) Ratio of overlapped target IPs to total target IPs in a dropzones.

Figure 3.4: CDF graphs showing the distribution of the number of overlapped target IP addresses and their ratio.

3.4 Geographical Analysis

In this section, we focus on the distribution of the distances between the dropzones and their target IPs. It will be apparent that a large number of dropzone-target pairs have a certain range of distances, which is related to the distribution of dropzones and their targets in each country. For example, we noticed that there are several target IPs located in Vietnam, Brazil, and China. To visualize the flow of attacks in a holistic sense, we plotted circular areas whose sizes are proportional to the number of targets and are placed according to their location on a world map with geodesic lines originating from various dropzone locations (see Fig. 3.6).

3.4.1 Distance Between Dropzone and Target

As mentioned previously, a dropzone IP can be associated with several malware instances where each malware can point to one or more target IPs. Knowing the locations of these IPs, we calculate

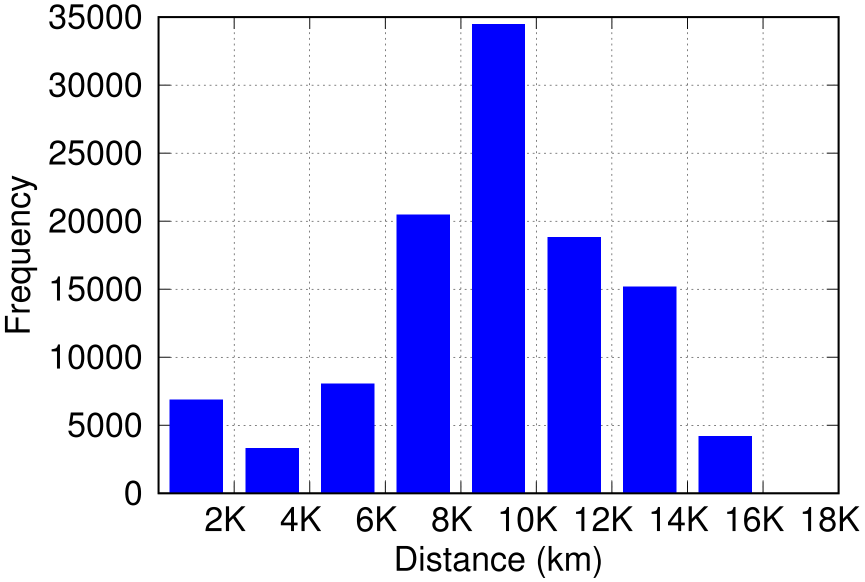


Figure 3.5: Histogram graph of distances between dropzone and target IPs. One bar represents the number of dropzone-target pairs with a distance that has a value within the range of x-axis.

Table 3.3: Top 5 countries by the number of target and dropzone IPs. Countries include: United States (US), Netherlands (NL), France (FR), United Kingdom (GB), Italy (IT), Vietnam (VN), Brazil (BR), China (CN), India (IN) and Pakistan (PK).

Rk.	Country	Dropzones	Pct.	Rk.	Country	Targets	Pct.
1	US	1,041	43.25	1	VN	26,290	24.70
2	NL	278	11.55	2	BR	20,572	19.33
3	FR	188	7.81	3	CN	15,799	14.84
4	GB	183	7.60	4	IN	5,598	5.26
5	IT	177	7.35	5	PK	5,076	4.77

the distance between the dropzone and its target if they are related to the same malware instance. Each distance shows the locality of the attack. The total number of calculated distance cases is 111,480. Fig. 3.5 presents an alternate view with a histogram plot of the distances between the dropzones and their target IPs.

Our result of the majority of the distance shows the 8K-10K km range had the most frequent

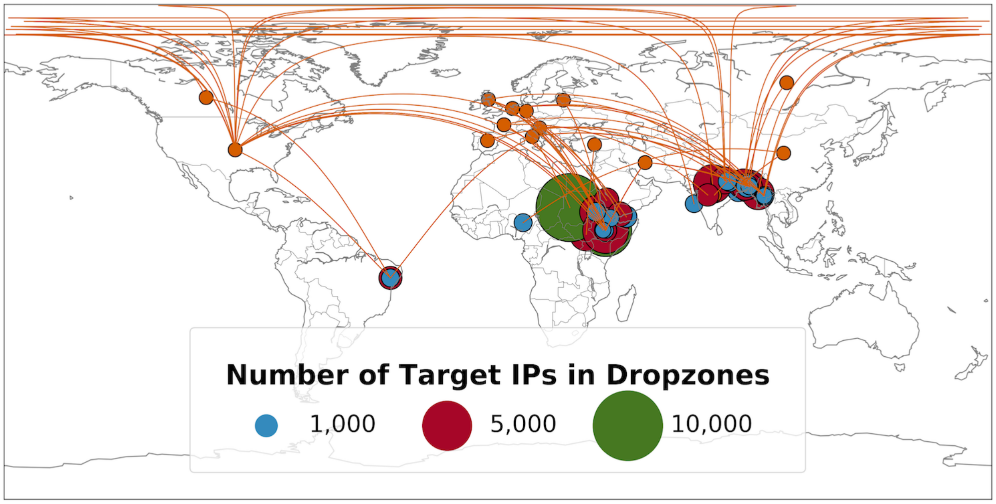


Figure 3.6: Attack trends between dropzones and target IPs. We only plot attacks that have over 500 target IPs. The orange circle represents dropzones, and blue, red, and green circles stand for target areas.

number of cases totaling 34,479 (30.93% of all dropzone-target distance cases). In this range, countries with the most target IPs are Brazil, Vietnam, and China, in order; while the dropzones are in European countries, including Italy, France, and the Netherlands. According to Table 3.3, a large number of dropzones exist in the US, but they also have target IPs in Brazil, Vietnam, and China, with distance between dropzone and target in the range of 12K-14K km and 10K-12K km.

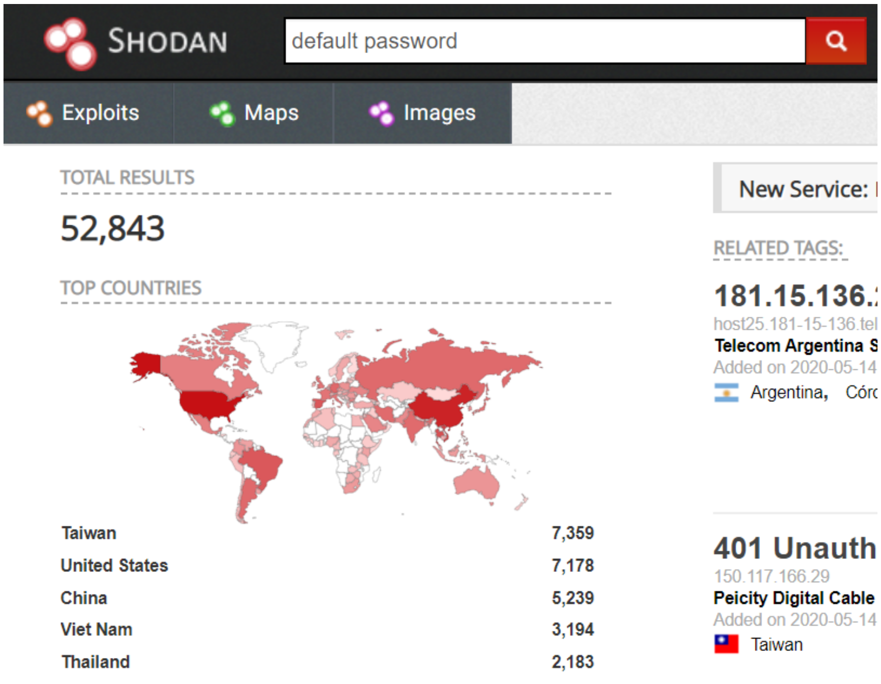
3.4.2 Country-level Analysis

In this part, we look at the overall attack trend between dropzones and their targets on a world-scale. For each dropzone, we collect all of the target IP addresses and extract location information (*e.g.,* latitude, longitude) to display the *average* position of the target area (not the exact position). The target areas are scaled according to the number of target IP addresses they contain. Fig. 3.6 shows the results of our country-level analysis, where we limit to only plotting dropzones with more than 500 target IP addresses. The locations of the dropzones (depicted in orange) are spread around various countries, but we highlight that there is a large concentration of target areas focused in Central Asia.

Table. 3.3 lists top 5 countries by the number of dropzone and target IPs. Note that the US has a large distribution of dropzones pointing to targets in Asian countries such as Vietnam. Additionally, China and Brazil contain a large number of target IP addresses originating from European countries. Imperva Incapsula (a global content delivery network and DDoS mitigation company) confirms that Vietnam (12.8%), Brazil (11.8%) and China (8.8%) were the countries with the most-infected devices (from the Mirai botnet) [44]. Moreover, intuitively, these countries should contain the highest representation of vulnerable devices, such as devices with default credentials or known vulnerabilities. To validate the former, we query "default password" in Shodan. Fig. 3.7 shows the result of searching in Shodan. We find Taiwan, the United States, China, Vietnam, and Thailand in the top five countries, which is partially counter-intuitive. In Vietnam, the Deputy Director-General of Ministry of Information and Communication's Authority of Information Security indicated the number of cyber threats and criminals increasing. Moreover, there are a large number of no-name IoT devices used in the country, such as cameras, which pose a threat to cybersecurity. That lead the Vietnamese government to develop a national strategy, including specific policies and regulations for IoT products [36]. As reported by the National Law Review [67], the EU and the US are the most developed regions in regards with establishing IoT Regulations and ethical frameworks. They have enacted standards and regulations for the devices connected, the network and their security; and the data associated with the devices.

3.4.3 Region-level Analysis

Using regional information from IPinfo [8], we plot a heatmap representing the distribution of dropzones and targets for the entire United States. In Fig. 3.8(a), we see that Washington state and New Jersey contain a high concentration of dropzones. Interestingly, the Washington and New



TOTAL RESULTS

52,843

TOP COUNTRIES

Taiwan	7,359
United States	7,178
China	5,239
Viet Nam	3,194
Thailand	2,183

New Service: I

RELATED TAGS:

181.15.136.:
host25.181-15-136.tel
Telecom Argentina S
Added on 2020-05-14
Argentina, Córc

401 Unauth
150.117.166.29
Peicity Digital Cable
Added on 2020-05-14
Taiwan

Figure 3.7: Result of searching for the “default password” in Shodan. This result indicates East Asia countries are exposed threat of malware.

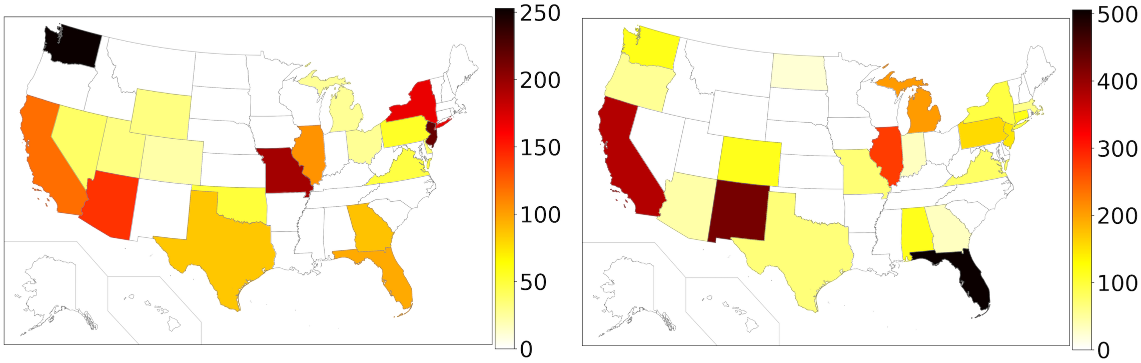
York states have lots of data centers as reported by Statista, an online portal for statistics [57]. We discuss this issue further in §4. Likewise, we see in Fig. 3.8(b) that a high number of target IPs reside in Florida and New Mexico. Table. 3.4 lists the detailed breakdown of the top 5 dropzone and target IPs according to their US State. In all, we had 1,037 dropzone IPs distributed over 20 US States and 1,650 target IPs spread over 22 US States.

3.4.4 City-level Analysis

We utilize the search engine Shodan [12] to look up the actual city in which the given dropzone IPs are, and use IPinfo to find the city information of the target IPs (whenever available). We note that in our data, city information may not exist for every IP, so our region-level and city-level analysis

Table 3.4: Top 5 US states by the number of target IPs and dropzone IPs. States include: Washington (WA), New Jersey (NJ), Missouri (MO), New York (NY), Arizona (AZ), Florida (FL), New Mexico (NM), California (CA), Illinois (IL), and Michigan (MI).

Rk.	State	Dropzones	Pct.	Rk.	State	Targets	Pct.
1	WA	253	24.40	1	FL	506	30.67
2	NJ	188	18.13	2	NM	356	21.58
3	MO	151	14.56	3	CA	283	17.15
4	NY	112	10.80	4	IL	151	9.15
5	AZ	79	7.62	5	MI	83	5.03



(a) Distribution of dropzones by US State.

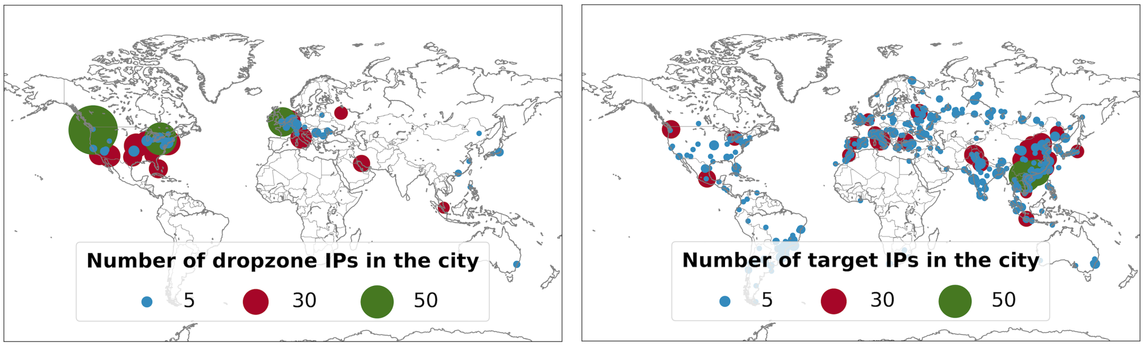
(b) Distribution of targets by US State.

Figure 3.8: Distribution of dropzones and target IPs in the United states. This figure shows dropzone and target mainly exist in which state in the US.

show different distributions. Overall, we had 541 dropzone IPs distributed among 75 cities, and 1,003 target IPs spread over 364 cities. In the Table 3.5, we list the top 5 cities per the number of dropzone and target IPs. We can see the US cities top the rank for the dropzone, and China and Vietnamese cities top the ranks for targets. In Fig. 3.9(a), we note that the blue circles represent the number of dropzone IPs in the range (0, 5] with red and green circles representing dropzone IPs in ranges (5, 30] and (30, 120), respectively. Similarly, Fig. 3.9(b) has the blue circle representing target IPs in the range (0, 5] with red circles as (5, 30] and green circles as (30, 50).

Table 3.5: Top 5 cities per dropzone and target IPs. Cities in the US occupy top ranks with respect to dropzone IPs and cities in Vietnam and China have a lot of target IPs, which demonstrate a case similar to the countrty-level analysis.

Rk.	City	Dropzones	Pct.	Rk.	City	Targets	Pct.
1	Seattle	113	20.89	1	Hanoi	48	4.79
2	Buffalo	49	9.06	2	Guangzhou	32	3.19
3	London	39	7.21	3	Beijing	21	2.09
4	Clifton	29	5.36	4	Rome	19	1.89
5	Kansas City	27	4.99	5	Islamabad	18	1.79


Number of dropzone IPs in the city
5
30
50
Number of target IPs in the city
5
30
50

(a) Distribution of dropzones by city.

(b) Distribution of targets by city.

Figure 3.9: Distribution of dropzones and target IPs throughout the world. In this figure, we can see that a large number of dropzones are distributed in the US and Europe, and targets are mainly distributed in Asia countries (Vietnam, China).

3.5 Network Centric Analysis

Malware specifically aimed at IoT devices tend to recruit a large number of intermediary targets to launch attacks on high-profile targets ultimately. To do this, the malware typically identify the intermediary targets using their IP addresses which are either mentioned in their source code or downloaded via dropzone. Additionally, these IP addresses could be masked IP addresses, showing only a prefix (*e.g.,* `123.17.%d.%d`). For further analysis, we performed masked IP address mapping to IP addresses through Censys.

Table 3.6: Composition of target IPs for masked and not-masked networks. “In Total” means the total number of target IPs, “In Unique” means the composition of non-duplicated target IPs.

Address	In Total	Pct.	In Unique	Pct.
/24	137	0.13%	27	1.22%
/16	104,369	98.07%	1,869	84.53%
/8	776	0.73%	126	5.70%
Not-masked	1,146	1.08%	189	8.55%
Total	106,428	100.00%	2,211	100.00%

3.5.1 IP Address Mapping

In the previous sections, we analyzed the IP addresses explicitly stated in the malware code base. For the masked IP addresses, malware typically uses functions to hide the targets from the malware analysts and determine the targets dynamically. This functions invoked during run time to determine the remaining of the masked octets. Malware authors seldom obfuscate these functions – we, therefore, in this section, examine the entire /16, /24, or /8 network to probe their susceptibility.

Using Classless Inter-Domain Routing (CIDR) notation, Table. 3.6 shows that 98.92% of the target endpoints are masked, mapping to 126 unique /8 networks and 1,869 unique /16 networks and 27 unique /24 networks. Removing the /16 networks covered in /8 and /24 networks, we have 125 /8 networks and 435 /16 networks. These 560 networks are then searched on Censys [38] which map to 100,793,403 active IP addresses, which also allows us to analyze their active ports.

Towards this, we made a request to Censys a research access to query all of the Censys data through Google BigQuery. With a research access permission, we sent a query with the masked IP addresses to map active IP addresses. We did not only collected IP addresses but also device and active ports information for each IP addresses which exists in the dataset. Using the collected information, we can statistically classify ports, whether they are used or not, by devices.

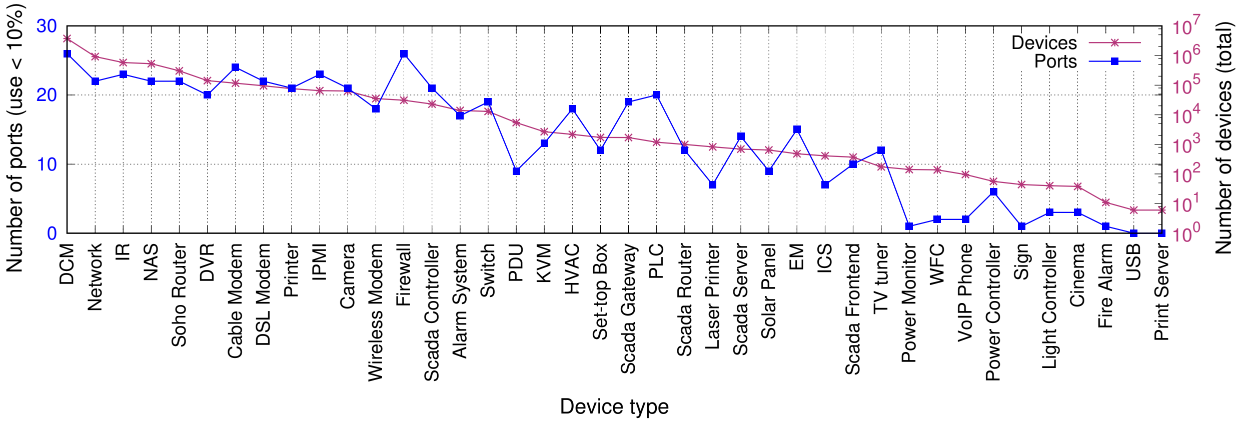


Figure 3.10: Total number of devices and the number of ports that used less than 10% of devices. The left Y-axis belongs to the number of ports (blue), the right Y-axis belongs to the total number of devices (red), and the X-axis is device types. Device types include: DSL/cable Modem (DCM), Infrastructure Router (IR), Network Attached Storage (NAS), Digital Video Recorder (DVR), Intelligent Platform Management Interface, (IPMI) Power Distribution Unit (PDU), Kernel-based Virtual Machine (KVM), Heating, ventilation, and Air Conditioning (HVAC), Programmable Logic Controller (PLC), Environment Monitor (EM), Industrial Control System (ICS), and Water Flow Controller (WFC).

3.5.2 Active Ports and Suspicious Ports

As different devices use different services to operate, we clustered the IP addresses by their device types and studied which ports were being used by the devices. Considering that open ports lead to increased security risks, we look for ports that are necessary for a device to operate without any misfire. Taking a conservative approach, we suggest that if a port is being used by less than 10% of devices in a given device type, it should be closed to reduce its exposure to risk. We observe that except for *VoIP phone* (over 77% of them used 5 ports), more than 75% of the devices among all the other device types have only two or less port being used. Fig. 3.10 shows the number of devices within a device type in log scale and the number of ports being used by less than 10% of the devices. In this figure, the two graphs show a similar pattern. We speculate this result is due to

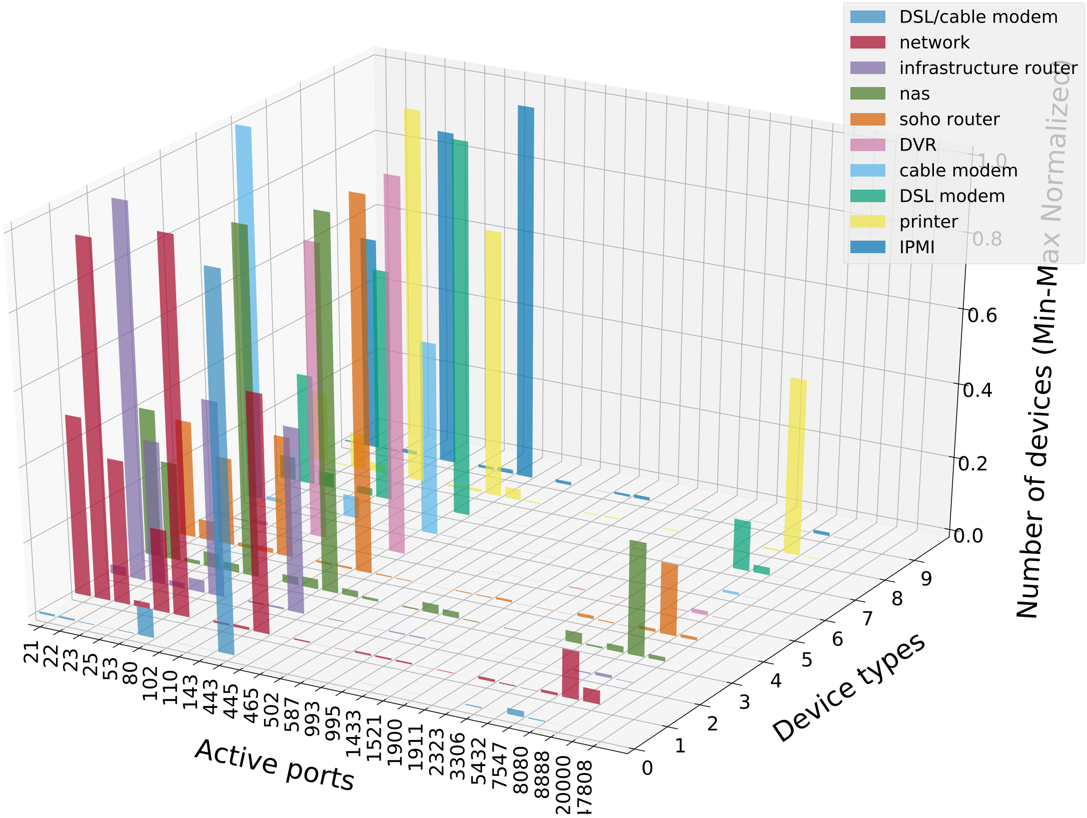


Figure 3.11: Number of top 10 devices by active ports and device types.

more attack taking place on the popular devices (*e.g.,* target devices of the Mirai consist of security cameras, DVRs, and consumer routers [22]). Additionally, the susceptibility of high-wattage IoT devices, such as heating, ventilation, and air conditioning (HVAC), power distribution units (PDU), etc., can be abused by the attackers to launch large-scale coordinated attacks, *e.g.,* power grids, as has been demonstrated by Soltan *et al.* [76].

3.5.3 Summary

The division of the endpoints by devices and then determining their exposure to the attackers represent the chances of an endpoint being compromised. Based on our analysis we suggest the users close the ports that aren't necessary for the uninterrupted execution of their devices. These endpoints need to be further examined in-depth to understand the pattern that could predict an endpoints chances of being compromised. The suggestions could be finally narrowed, with specific device centred recommendations, and by probing them individually by performing an offensive penetration testing. However, in this work, we understand the data-centric landscape and put forward the suggestions with a conservative approach, and without carrying out any offensive analysis undermining ethics.

3.6 Summary

In this chapter, we analyze the $\approx$78.2% of total responsive public IPv4 endpoints among dropzones and their targets as extracted from IoT malware and spread across the globe from diverse perspectives. First, we analyze the dropzone-target inter-relationship and their affinity. We observe that the list of targets is shared between attackers, or are compiled by abusing shared susceptibilities. We visualize the target areas representing dropzone locations and their size scaled by the number of associated targets.

Our distributed analysis shows the exposure of endpoints which we correlate to the risk they possess. These endpoints need to be carefully and individually analyzed to extract patterns for predicting the chances of them being compromised.

CHAPTER 4: IOT MALWARE NETWORK PENETRATION ANALYSIS

4.1 Motivation

In this chapter, we focus on the additional attributes contained in the IP address, leveraging the information gathered from Shodan and Censys [3] such as active ports, vulnerabilities and organizational information.

- We perform a network penetration analysis of the targets and dropzone IP addresses. To do this, we analyze the risk associated with the IP addresses through insights gained from Shodan.
- We analyze characteristics in IP addresses, which is active ports, vulnerabilities, and organizations.

4.2 Active Ports Analysis

For each dropzone and target IP address, we issue an API query call to Shodan and Censys to obtain a list of active ports. We extracted 5,745 active ports from 716 of the 877 dropzone IPs and 1,114 active ports from 129 of 189 the non-masked target IPs. It is to be noted that malware authors also mask the octets of the target IP addresses, which they determine dynamically during execution. In this analysis, we only use the IP the not masked addresses.

Each port number is typically associated with a computer or network service, such as port 80 for HTTP traffic. In Table. 4.1 and Table. 4.2, we list the top 10 active ports among dropzone IPs and target IPs, respectively. As shown in both tables, the largest portion of active ports is common services like SSH (port 22), HTTP (port 80), and HTTPS (port 443). However, we would like to point out some of the other active ports among target IPs appearing in Table. 4.2, like the SUN

Table 4.1: Top 10 active ports in dropzone IPs. The most of ports are used for general purposes.

Rk.	Port	Count	Pct.	Service	Description
1	22	641	32.57%	SSH	The Secure Shell (SSH) Protocol
2	80	600	30.49%	HTTP	World Wide Web HTTP
3	443	350	17.78%	HTTPS	HTTP protocol over TLS/SSL
4	25	276	14.02%	SMTP	Simple Mail Transfer
5	21	275	13.97%	FTP	File Transfer Protocol [Control]
6	3306	224	11.38%	MySQL	MySQL database system
7	53	187	9.50%	DNS	Domain Name Server
8	110	175	8.89%	POP3	Post Office Protocol - Version 3
9	143	171	8.69%	IMAP	Internet Message Access Protocol
10	993	165	8.38%	IMAPS	IMAP over TLS protocol

Table 4.2: Top 10 active ports in target IPs. With the exception of a few, most shown ports are used for common services.

Rk.	Port	Count	Pct.	Service	Description
1	80	111	17.85%	HTTP	World Wide Web HTTP
2	22	106	17.04%	SSH	The Secure Shell (SSH) Protocol
3	443	67	10.77%	HTTPS	HTTP protocol over TLS/SSL
4	21	51	8.20%	FTP	File Transfer Protocol [Control]
5	25	49	7.88%	SMTP	Simple Mail Transfer
6	3306	40	6.43%	MySQL	MySQL database system
7	53	29	4.66%	DNS	Domain Name Server
8	8080	29	4.66%	HTTP-alt	HTTP Alternate (see port 80)
9	111	28	4.50%	SunRPC	SUN Remote Procedure Call
10	123	26	4.18%	NTP	Network Time Protocol

Remote Procedure Call (RPC) on port 111 and the Network Time Protocol (NTP) on port 123.

Port 111. is a well-known port used by the *Port Mapper* service over the TCP and UDP protocols [69], which essentially is a port lookup service for the Open Network Computing Remote Procedure Call (ONC RPC) system designed by Sun Microsystems in the 1980s for their Network File System [10]. As described in RFC 1833 [69], the port numbers for RPC programs and services are determined dynamically on startup so if a client wishes to make ONC RPC calls they will query the Port Mapper on port 111 to obtain the appropriate RPC service port. Reported by L3 Communications in August of 2015 [1], the Port Mapper service became a new attack vector for adversaries seeking to amplify their DDoS attacks. This has to due to the fact that when Port Mapper is queried, the response size varies significantly depending on which RPC services are available on the host. In their examples, L3 Communications show that a 68-byte query results in a 486-byte response for an amplification factor of 7.1x with responses as large as 1,930 bytes for amplification of 28.4x. If adversaries "spoof" the victim's source IP for these UDP packets directed towards several vulnerable devices with port 111 open, they will ultimately be redirected back *en masse* towards the victim (*i.e.,* a UDP flood attack).

Port 123. is reserved for the Network Time Protocol (NTP), which was proposed in 1985 by D.L. Mills [37] to synchronize network clocks using a set of distributed clients and servers. Strangely enough, the content delivery network (CDN) and DDoS mitigation company called CloudFlare predicted in late 2013 that the NTP protocol could possibly be abused in DDoS amplification attacks [43]. This prediction came true in early 2014 [47] because much like the exploit for the Port Mapper service described above, NTP is also UDP-based and can be prone to "IP spoofing" for DDoS attacks. As emphasized in [80], exploiting NTP has a great potential for amplification attacks due to the "monlist" command that a typical attacker sends to an NTP server. The "monlist" command returns the last 600 IP addresses previously synchronized with the NTP server which is spread over 30 separate UDP packets, each of which is 448 bytes. The overall size varies depending on the server, but the data volume is almost 1,000x larger than the packet originally sent by the attacker.

4.3 Vulnerabilities Analysis

We then explore the susceptibility of IP addresses. In particular, we examine the vulnerabilities present in IP addresses. Towards this, we analyze the use of vulnerable services running on endpoints. For the vulnerable endpoints, we gather the Common Vulnerabilities and Exposures (CVE) identifier, *e.g.,* “CVE-2017-15906”. CVE, maintained by MITRE [29], which standardizes security vulnerabilities using their own naming schemes.

We analyze the vulnerabilities in the dropzones towards understanding their dynamics. Table. 4.3 depicts the top six vulnerabilities by the number of dropzone IPs. We further analyze the vulnerabilities to understand the root cause behind their existence. We found that CVE-2017-15906 is the most frequent vulnerability which is related to 203 dropzone IPs and 448 instances of malware. According to the National Vulnerability Database (NVD) [9], CVE-2017-15906 is a “medium” severity vulnerability where versions of OpenSSH before 7.6 do not properly prevent write operations in *readonly* mode. This allows attackers to create several zero-length files which could possibly exhaust disk space. For the second most common CVE among the dropzones (*i.e.,* CVE-2014-1692), the NVD reports the severity of this vulnerability as “high” since it might allow remote attackers to launch a DoS through memory corruption due to uninitialized data structures from the `hash_buffer` function in OpenSSH. These vulnerabilities allow attackers to launch a denial of service (*e.g.,* memory corruption or disk saturation) of target devices. These vulnerabilities do not say much about the dropzones. Additionally, we observe that 98.61% of dropzone IPs, which have the vulnerability of CVE-2014-1692, also possess at least one of the vulnerabilities associated with unauthorized authentication resulting in access to the device; this is, remote unauthorized authentication attempts can be an indicator of a potential dropzone.

CVE-2016-0777, CVE-2012-0814, and CVE-2010-4478 are vulnerabilities that allow an attacker to obtain access permission on target devices by stealing sensitive information; (*e.g.,* private key

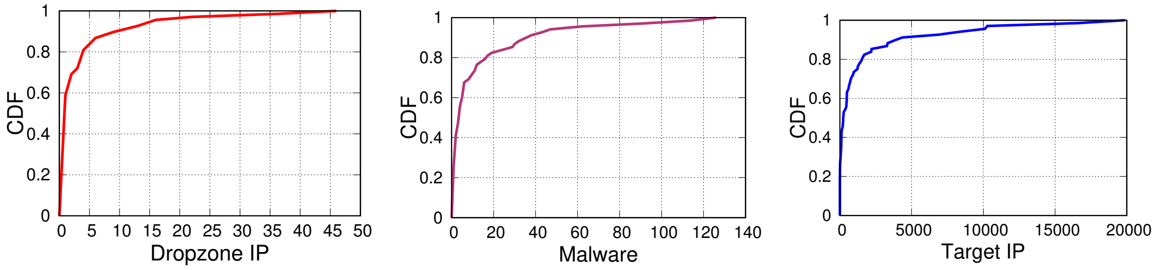
Table 4.3: Top 6 Vulnerabilities by the number of dropzone IPs. Note that the dropzones use vulnerable version of OpenSSH.

Vulnerability	IP	Malware	Description
CVE-2017-15906	203	448	OpenSSH/DDoS
CVE-2014-1692	142	320	OpenSSH/DDoS
CVE-2016-0777	142	325	OpenSSH/Private Key leakage
CVE-2012-0814	140	307	Cross-privilege boundaries/OpenSSH
CVE-2011-4327	140	307	OpenSSH/Authentication leakage
CVE-2010-4478	140	307	OpenSSH/Authentication override

or authorized key), bypassing the authentication and crossing privilege boundaries. Additionally, CVE-2011-4237 enables remote attackers to make an unauthorized modification to the kist of authenticated keys by injecting arbitrary HTTP header. We observe that 17.04% of dropzone IP addresses (*i.e.,* 144) have at least one of these four vulnerabilities which provides a large attack surface to the attackers. This is, these vulnerabilities provide a broad range of attack surface that can be abused by attackers to compromise the devices, then act as dropzones. Moreover, some of the dropzones use the default credentials making the devices an attractive target.

4.4 IP-Owning Organizations Analysis

We then analyze the organizations that own the given IP space, *e.g.,* Starhub Mobile. In Fig. 4.1 we have plots for the CDF between the IP-owning organizations and their: number of dropzone IPs, number of malware instances, and number of target IPs they point to. A breakdown of the top 10 IP-owning organizations is presented in Table. 4.4. Note that for each organization, there is clearly a relationship between the number of dropzone IPs and the number of malware instances they are associated with. However, there are abnormal cases in our dataset, such as Cogeco Peer 1 (Canada) and MAROSNET Telecommunication Company LLC (Russia) who have only 1 dropzone IP but point to 2,214 and 2,178 target IPs, respectively. In contrast, HOSTKEY (that operates in the



(a) Organizations v. the number of dropzone IP.

(b) Organization v. the number of malware.

(c) Organization v. the number of target IP.

Figure 4.1: Organization v. the number of dropzone IP, malware and target IP. The similarity in the CDFs is because there is a high probability that the organization that includes more malware will contain more dropzone and target, except in a few cases (*i.e.,* where one dropzone refers to thousands of targets, or one dropzone refers to only one target).

Netherlands and Russia [6]) has only 2 dropzone IPs that are associated with 2 target IPs and 2 instances of malware.

Interestingly enough, the *locations of these organizations coincide with the heatmap of US States* presented in Fig. 3.8(a) which illustrates the highest distribution of dropzones. For example, the organization with the greatest number of dropzone IPs according to our data is Wowrack.com, which is a cloud service provider with headquarter offices in Seattle, Washington [14]. In addition to its flagship data center in Seattle, Wowrack operates eight other data centers in multiple cities across the United States and Southeast Asia. As reported by AbuseIPDB [2], Wowrack.com IPs have received several complaints of abusive activity from multiple sources. The Canadian Internet Registration Authority (CIRA) has also urged IT, administrators, to block domains originating from Wowrack.com (*e.g.,* `ns6.wowrack.com`) because they are associated with the Mirai IoT botnet [85].

Referring to Fig. 3.8(a) again, you will notice that New York State contains a high number of dropzones--which is most likely caused by two organizations from Table. 4.4 that have data centers in the city of Buffalo, NY (Wowrack.com and ColoCrossing [5]). Also highlighted in red is the

Table 4.4: Top 10 organizations by the number of their dropzone IPs. In this table, most organizations increase linearly in the number of dropzone, malware, and target.

Rk.	Dropzone IPs	Target IPs	Malware	Organization
1	46	16,586	126	Wowrack.com
2	36	19,878	114	Aruba S.p.A
3	22	10,282	62	Choopa LLC
4	16	3,816	47	DigitalOcean LLC
5	16	3,330	29	ColoCrossing
6	13	8,373	38	NForce Entertainment B.V.
7	11	1,701	24	Hydra Communications Ltd
8	9	4,354	35	Ad Net Market Media Srl
9	9	388	17	Wholesale Data Center LLC
10	6	1,220	8	Input Output Flood LLC

state of Arizona, which is the home of Input Output Flood LLC [7] (ranked 10th for organizations with several dropzones).

We cannot say for certain why these organizations contain the most dropzone IPs in our dataset; we can only surmise that they are more tolerant when it comes to harboring customers who engage in "malicious" activities. For example, the organization with the 3rd-highest number of dropzones in our dataset is Choopa LLC. With their primary Point of Presence (POP) in the State of New Jersey [4], you can clearly see that this affects the heatmap (colored dark-red) shown in Fig. 3.8(a).

While online public reviews may not be the most-trustworthy source [79], the low ratings from the Google reviews of Choopa LLC put them in a negative light when several people post complaints such as the following:

> *Why is this company still operating?! Yes, maybe it's not them trying to hack into my email, but they let user do it!!!! Crooks!!* --M. Alther

> *Lots of SSH attacks coming from their network. Reported via blocklist.de multiple times and I still am seeing scans.* --C. Simmons

You guys need to pick your customers better...Our e-commerce site is getting a lot of fraud orders coming from your IP. I know its not your fault that customers are using your servers for illegal activities. But do understand they ruining your reputation as a company. --W. Struggle

4.5 Summary

We observe that while port 80 and 22 are the most widely used across endpoints, we see the usage of port 111 and port 123 by the target IP addresses. Considering the possibility of the two ports being an entry point for an attacker, the usage should be limited. We also observe that the low presence of vulnerable services by dropzone IP addresses. The vulnerabilities in the dropzones reflect that the attackers override their authentication status and then utilize the OpenSSH vulnerabilities to gain access to the device. Our analysis also sheds light on the tolerance of organizations towards the endpoints with malicious intent. The tug of war between the profit and the loss of trust among the users (prospective domain buyers) deserve the focus of the community.

CHAPTER 5: UNDERSTANDING THE DYNAMICS AND INTERDEPENDENCIES IN IOT BOTNETS[1]

5.1 Background

To address IoT security, malware behavior analysis methods are employed, including dynamic and static analysis. Dynamic analysis is concerned with understanding malware by inspecting run-time 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. [46]. 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 [15, 19, 20], making this approach safer and faster than dynamic analysis [49], although subjecting to static analysis circumvention techniques, such as code obfuscation [33, 61], typically addressed with de-obfuscators.

Static Analysis. In this section, 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

[1] This content was reproduced from the following article: J. Choi, A. Abusnaina, A. Anwar, A. Wang, S. Chen, D. Nyang and A. Mohaisen, “Honor Among Thieves: Towards Understanding the Dynamics and Interdependencies in IoT Botnets”. IEEE Conference on Dependable and Secure Computing, IDSC, pp. 1-8, Hangzhou, China, 2019. The copyright form for this article is included in the appendix.

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 research. The results of our analysis are further in subsection, IoT Malware Static Analysis.

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.

5.2 Motivation

- We conduct string static analysis over IoT malware binaries to extract communicated and referred IP addresses, and keywords reflecting the malware behavior.
- We conduct a spatial distribution analysis on the extracted dropzones and targets, where region dependencies within the extracted addresses are shown.
- 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.

Table 5.1: 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%

5.3 Dataset

In this research, we relied on a dataset of IoT malicious binaries obtained from CyberIOCs [32]. 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 [32]. We reverse-engineered the samples using *Radare2* [11], 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* [34] and gathered the scanning results corresponding to each sample. Then, we used AVClass [71] to match the samples with their corresponding IoT malware families. Table 5.1 shows the distribution of malware families in the dataset; as shown Gafgyt and Mirai represent the majority of our dataset, 99.31%.

5.4 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 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.

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.

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 5.1 shows a general structure of the malware-IP relationship.

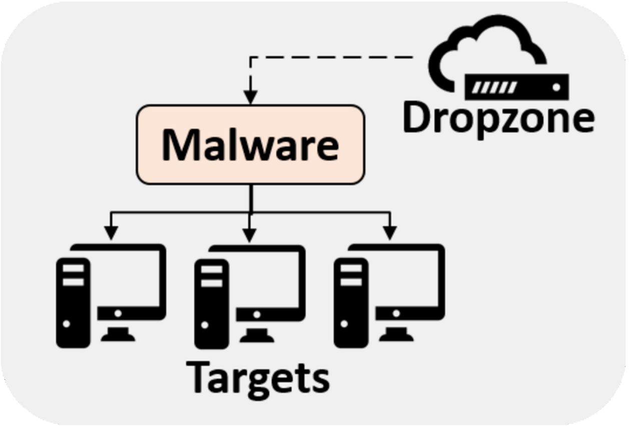


Figure 5.1: General structure of the malware-IP relationship. Malicious binaries are obtained from the remote dropzone and are accessible using *wget*, *GET*, etc.

5.5 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 run-time 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 5.2 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.

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 hard-coded 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

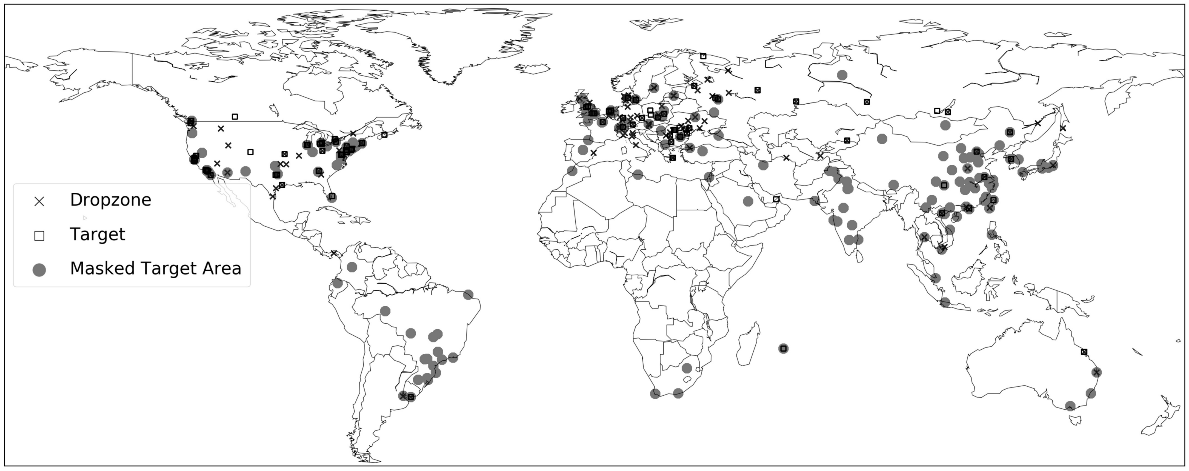


Figure 5.2: The distribution of the extracted components worldwide. Here, the target refers to the location of the unique target IP address extracted from the string analysis of the IoT malware. The masked target is represented by its center. To estimate the locations of the masked target area, we convert the masked part to zeros (*i.e.,* 183.229.%d.%d to 183.229.0.0).

Table 5.2: 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%

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 5.2 shows the distribution of the unique target IP addresses.

Table 5.3: 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

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 5.3 shows the distribution of the benign target IP addresses over organizations and information centers, highlighting a wide distribution.

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 5.4 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 5.5 shows the distribution of the malware families per dropzone. We notice that

Table 5.4: 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 5.5: 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%

one dropzone contains the malicious binaries of Gafgyt, Mirai and Tsunami families, highlighting the shared infrastructure.

Dropzones Distribution. Dropzones have spatial localities in their distribution, as shown in Figure 5.2. Moreover, Figure 5.3 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 5.6 shows the top dropzones hosting countries, highlighting—not surprisingly—a heavy-tailed distribution.

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.

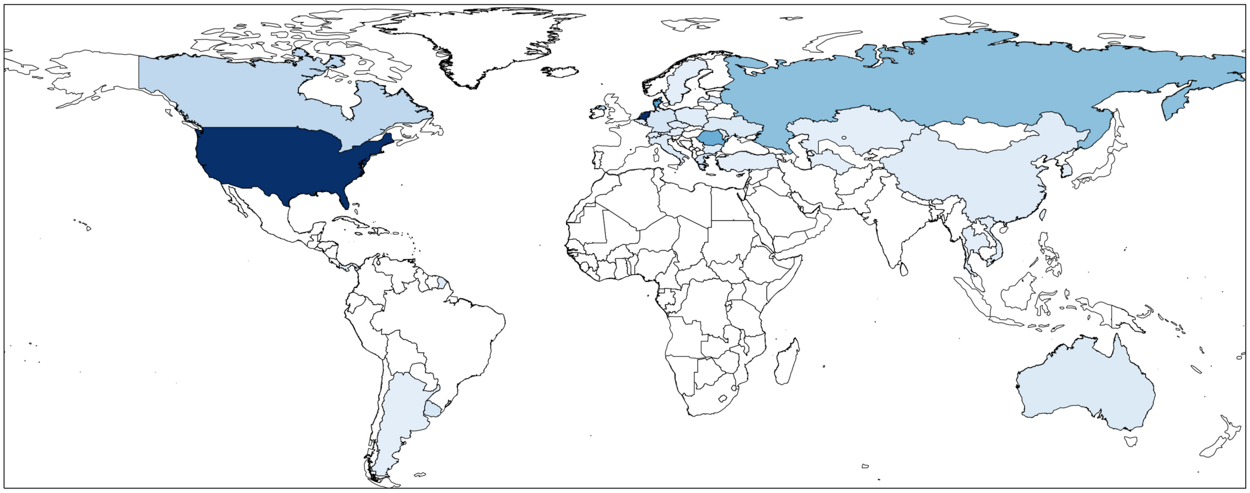


Figure 5.3: The distribution of dropzones over the countries. The color shade reflects the number of dropzones within the country, where darker shade represents more dropzones.

Table 5.6: 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%

5.6 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 run-time 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.

5.7 Dropzones Chains

In section IP Addresses Analysis, 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 5.4 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,

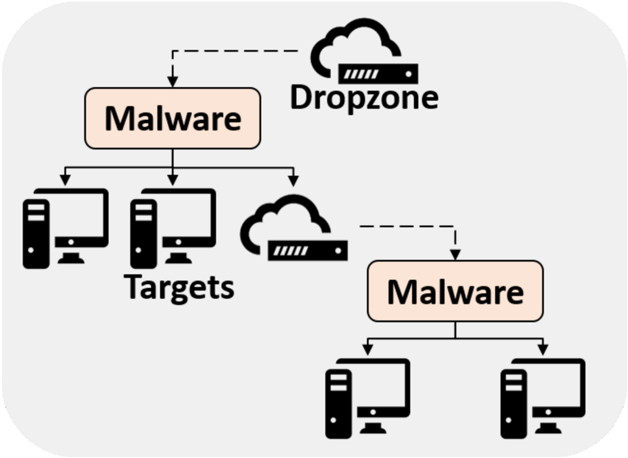


Figure 5.4: Structure of the dropzones chain of length two, where malware access dropzone, and targets another dropzone.

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 5.5 shows the dropzone to dropzone chain links visualization worldwide. Notice that some dropzones are directly linked to several dropzones.

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 5.7 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 [22, 54].

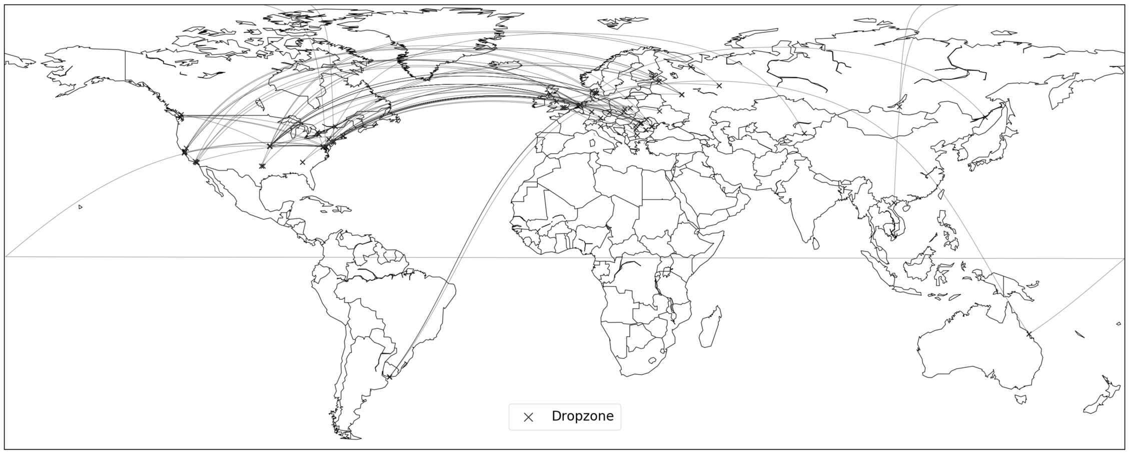


Figure 5.5: Dropzone to dropzone links visualization. Here, a remote location may be a dropzone of several dropzones. Links connect the location of the dropzone with the location of the targeted dropzone.

Table 5.7: 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

Chains Region Distribution. Dropzones within the chain have locality characteristics. Figure 5.6 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 5.8 shows the top five countries hosting dropzones within chains. Notice that the countries are the same as Table 5.6. 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, mean-

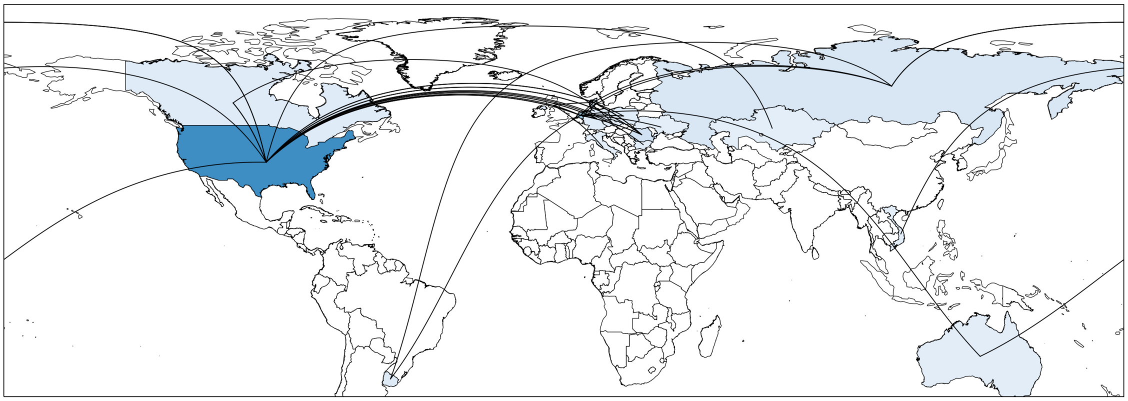


Figure 5.6: Country level distribution of the dropzones within the chains. Color shade indicates the number of dropzones within the country. Here, link connect the country of the dropzone with the country of the targeted dropzone.

Table 5.8: 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%

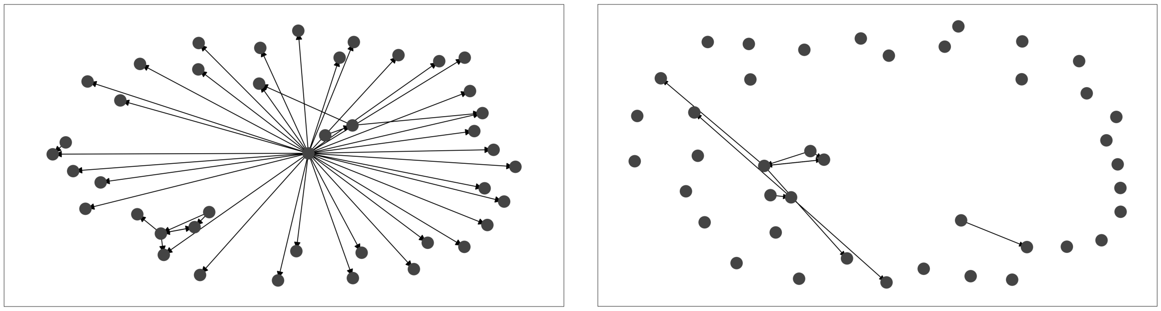
ing that the remaining dropzones (79.34%) may be part of chains not observed by the collected samples.

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. To find central dropzones, we use graph characteristics, namely the centralities such as the degree-, betweenness-, closeness-, and eigenvector-centrality. In our experiment, we found degree-centrality is the most effective way to find the essential nodes of the dropzone chain. We utilize the closeness, betweenness, and eigenvector centrality to remove the center dropzone in the chain, but these factors remove another node and reduce only two to five edges. Figure 5.7 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 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.

5.8 Summary

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



(a) Before

(b) After

Figure 5.7: The effect of removing the central dropzone over the chain. Fig. 5.7(a) is the chain before the removal of the dropzone, and Fig. 5.7(b) is the same chain after the removal of the dropzone. Notice that most dropzones became disconnected and isolated, thus removed from the chain.

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 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.

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 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 section Malware-Target Relationship.

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 [16, 30]. In addition, it has been reported that Mirai is an evolution of Bashlite malware, including Gafgyt [22, 54]. Our analysis, although not concerned with the capabilities, hint on such an evolution from an infrastructure standpoint.

CHAPTER 6: OPTIMIZATION FOR DEFENSE

We analyze the IoT malware ecosystem and the interdependencies of IoT botnets. In the IoT malware ecosystem, we aim to inter-relationship between dropzones and targets, geographical distributions, and network centric analysis. Also, we study interdependencies and dynamics in IoT botnets. Based on this analysis, we define a chain that relationships among the malware dropzones. Through these studies, we look at methods to mitigate the threat of IoT malware in countries.

We extract the malicious IP addresses from IoT malware through static analysis. Based on this, we analyze the open ports of the IP addresses targeted by IoT malware. Through this, the device can be distinguished from the port that is mainly used and the port that is not. If unused ports are opened, an IoT device that is poorly managed is likely to be an attacker. Therefore, we suggest blocking the ports that are not used for the main role in the devices. Moreover, similarly, we study which ports affect country-level through this analysis.

In this work, we make the following contributions:

- We propose a method to mitigate the threat in the IoT malware endpoints. It is based on the open ports by the countries.
- We inspect open ports in the target IP addresses, we list the open ports and their numbers, and vulnerabilities.
- We analyze countries ratios in each open ports. In this analysis, we provide insight into strategy for block open ports.

Table 6.1: Suspicious ports in the target IP addresses open ports list.

25	2323	1433	8888	502
445	3306	110	143	993
47808	995	1911	1521	5432
102	21	22	8080	53
23	587	7547	20000	443

6.1 Open Ports Distribution

In this section, we analyze the open ports distribution at the country-level. There are vulnerable ports and we show the distribution for those ports over the countries. We inspect the reason why they show different patterns. We pick the ports from the previous analysis, attack exposure. We select ports that used each device under 1%, which is considered they are not the main role for the function. In the Table 6.1, we choose 25 ports to analyze the method for the mitigation of threats. These open ports exist in the IP addresses and they are not mainly used.

In the Table 6.2, we list the open port numbers, descriptions, number of vulnerabilities, and count of open ports, which is essential when using the Internet, occupy a large portion. *e.g.,* 443 (HTTPS), 22 (SSH), 8080 (HTTP alternative), 21 (FTP), etc.

In case port 7547, it is associated with TR-069, which is an application layer protocol for remote management of end-user devices. This port used by modems, gateways, routers VoIP phones, and set-top boxes. CVE-2016-10372 is one of vulnerability related to this port, allows remote attackers to execute arbitrary commands, it is recommended to block or filter it at the NAT router/-gateway [58, 77]. In the Figure 6.1, US account for 31% of the total number of 7547 port.

For port 8888, this port is used in the alternative of HTTP port. This port has a 6 of known vulnerabilities (CVE-2014-2967, CVE-2018-6892, CVE-2018-7886, CVE-2019-7678, CVE-2019-7677, and CVE-2019-7676). For Autodesk VRED programs that have a vulnerability (CVE-2014-2967)

Table 6.2: Open ports in the countries sorted by the number of CVE vulnerabilities. TSAP stands for Transport Service Access Point protocol, DNP stands for Distributed Network Protocol for SCADA networks, TR-069 stands for application layer protocol for remote management of end-user devices, BACnet stands for Building Automation and Control Networks, and MTP stands for Starlight Networks Multimedia Transport Protocol.

Port	CVE-Vulnerabilities	Count	Describe
102	12	3,166	TSAP
22	12	10,574,166	SSH
443	12	32,029,435	HTTPS
502	8	16,982	Modbus Appication protocol
53	8	5,545,751	DNS
23	7	2,382,294	Telnet
8080	7	8,850,356	HTTP (Alternative)
25	6	3,895,884	SMTP
8888	6	2,717,164	althttpd
21	5	7,469,501	FTP
20000	5	168	DNP
110	3	2,877,341	POP3
3306	2	2,961,005	MYSQL
143	2	2,640,195	IMAP
445	2	1,549,161	Microsoft-DS
1433	2	522,370	MSSQL
5432	1	428,575	PostgreSQL
1521	1	58,945	Oracle
2323	1	267,742	Applications
7547	1	15,147,494	TR-069
587	0	2,932,974	SMTP
47808	0	10,202	BACNET
995	0	2,550,497	POP3 over TLS/SSL
993	0	2,631,357	IMAP overTLS/SSL
1911	0	16,900	MTP

that could allow unauthorized remote code execution, this port allows an attacker to remotely access it. Importing Python's library, an attacker can execute an attacker's command with the privileges the program is running on. From the Figure 6.2, we can see that this port is mainly distributed in East Asia countries, Taiwan and China, 29.36% and 26.05%, respectively.

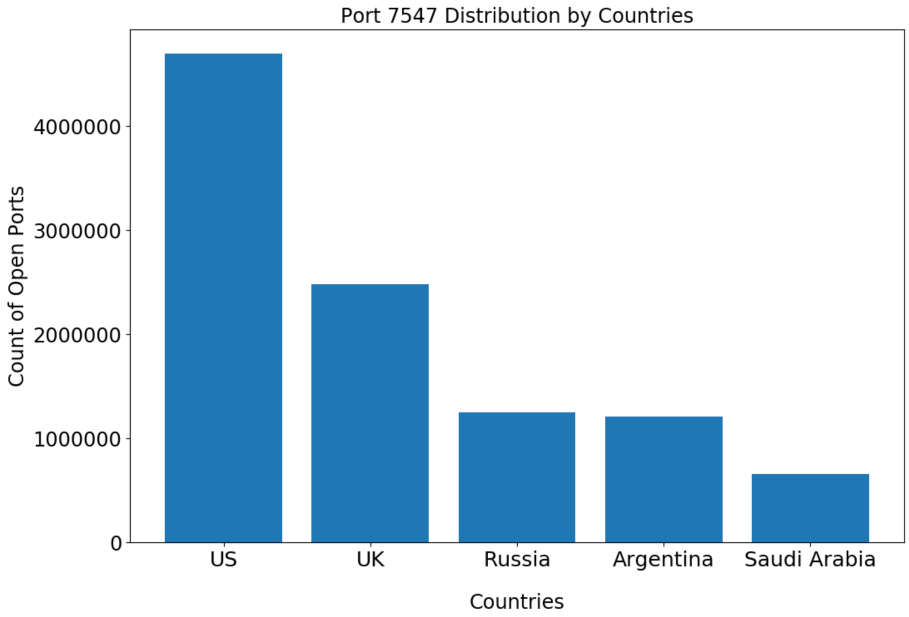


Figure 6.1: Port 7547 distribution by the countries. United States have most portion in the dataset.

Port 1433 uses for the Microsoft SQL Server. This port has two vulnerabilities, CVE-2002-1123 and CVE-2014-4684. CVE-2002-1123 is a buffer overflow in the authentication function for MSSQL server 2000 and Microsoft Desktop Engine (MSDE) 2000 allows remote attackers to execute arbitrary code through this port. CVE-2014-4684 also allows remote authenticated users to gain privileges with port 1433. This vulnerability exists in the database server in Siemens SIMATIC WinCC before 7.3 version. We can see the distribution of the port 1433 by the countries in Figure 6.3. Poland has 26.72% of the total number of port 1433, and China has 21.1% of it.

6.2 Mitigation for Defense

The defense method we propose through this analysis is as follows. It is to analyze the ports that are more important and account for each country, and block those ports. To do this, a detailed

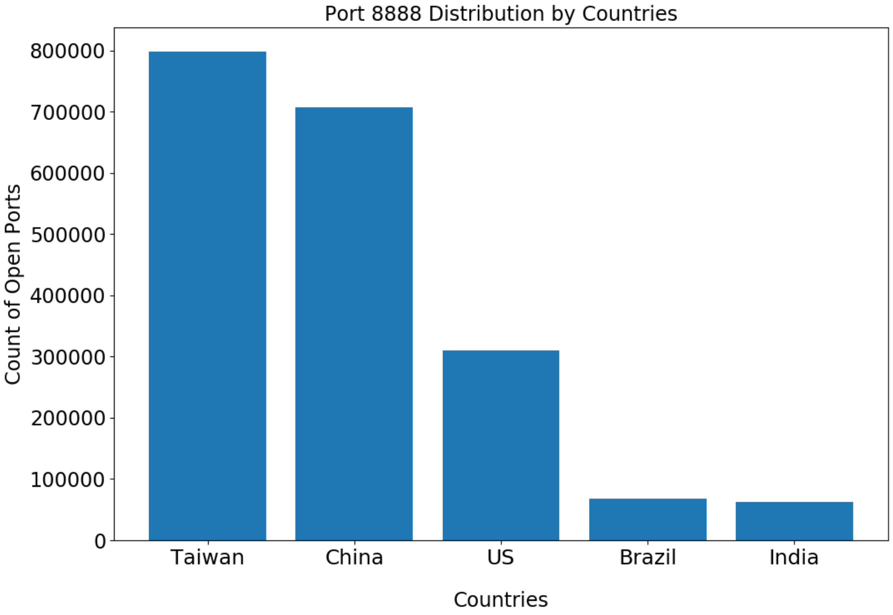


Figure 6.2: Port 8888 distribution by the countries. Taiwan and China have a lot of number of ports.

analysis of the network devices used in each country is necessary. In particular, it is necessary to analyze the ports that are open in network devices. Through this, it is possible to distinguish a port used for the main purpose of a device and a port that is not. In this research, we focused on ports that are not used mainly and analyzed how those ports are distributed by country. In addition, it should be reflected in the policy of filtering or blocking ports according to the vulnerabilities revealed by ports. For example, in the case of port 7547, it has one identified vulnerability, and it can be seen that it is mainly distributed in the UK, Russia, and the like. Even though there is only one vulnerability, the vulnerability should be managed more intensively in a country or enterprise because a remote attacker can execute arbitrary commands. Not only port 7547, but also other ports have vulnerabilities that attackers can exploit and execute remote commands or gain privilege with unauthenticated login. This is the reason why we focus on the mitigation method and block the ports.

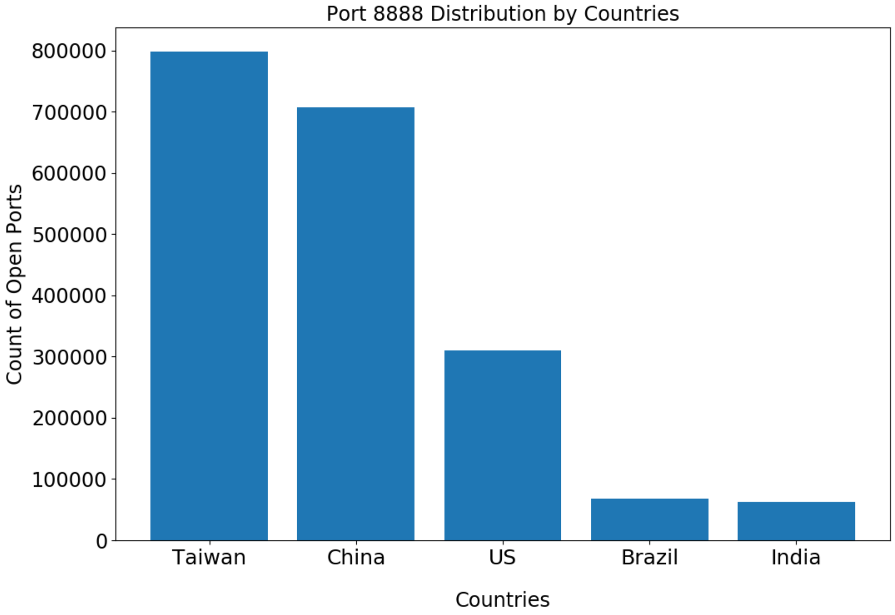


Figure 6.3: Port 1433 distribution by the countries. Poland and China have most of the ports.

6.3 Summary

Using our research results, we can select ports that are likely to utilize for the attack. Also, we analyzed the open ports distributed by country and vulnerabilities of each port. Through this analysis, we can more aggressively filter and block ports by country to prevent the spread and attack by IoT malware. If we apply the proposed method, mitigation strategies through these port blocks, such as router/gateway, can be used by companies in each country, network, and security within limited resources for defense purpose.

CHAPTER 7: DISCUSSION

7.1 Limitation

Our study is not without limitations. The key shortcomings in our work have to do with the IP address-Geolocation mapping, Obfuscated malware binary, dataset and target behavior, as seen from the static analysis artifacts. In this section, discuss to limitations of this research. We utilize IP addresses locations to discover affinities and dynamics among IoT malware endpoints. Toward this, we use Ultratools [13] and Censys services [3]. However, they provide geolocation based on the mapping service. Also, most of malware binaries are obfuscated to avoid detection and protect their code from the reverse-engineering. This makes it harder to analyze IoT malware binaries.

7.1.1 Accuracy of IP addresses-Geolocation Mapping

Ultratools derive the country information of an IP address from Regional Internet Registries (RIRs) and other locational data. RIR is an organization that manages the allocation and registration of IP addresses and autonomous system numbers within a region of the world. Some ISPs may contribute information to databases to help geolocation services. They use statistical formulas and analytical tools and analyze the data provided by online users [13, 35]. However, the location of IP addresses may be limited. Even if users utilize VPN, proxy, they can change the IP address. If we use a genuine IP address, it is difficult to analyze the location of the endpoint more precisely because it depends on the accuracy of a country, state, and city. We analyzed the dynamics and affinities between the endpoints of IoT malware based on the information released on the public Internet in this study. Using data about users' IP addresses and more detailed physical locations could be used to analyze relationships within a narrower range.

7.1.2 Obfuscated Malware Binaries

In this study, we used Radare2 to extract strings, especially IP addresses and keywords from non-obfuscated malware binaries. Manual string extraction from obfuscated malware binary requires a lot of time and effort. Therefore, we develop an automated tool for string extraction supporting IoT diverse architectures and using Radare2's API, and extract and analyze data for more range of IoT malware binaries. This, however, does not take into account the fact that some malware samples might be obfuscated, where our tool may not yield meaningful artifacts. This is indeed a limitation, and we will explore in the future dynamic analysis-based and other de-obfuscation tools that can be used to gain insight into string artifacts for obfuscated codes and binaries.

7.1.3 Dataset

Table 3.1 and Table 5.1 show the distribution of the families in the dataset utilized in this research. The dataset is biased toward Gafgyt, where 95%, 88% of the samples belong to these family, respectively. 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. We note, however, that this family is the most popular by far, and the findings in this work are of significant value given its prevalence.

7.1.4 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.

7.1.5 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.

CHAPTER 8: CONCLUSION

There is no doubt that IoT devices have found multiple applications in our daily life, and the number of those devices in use is soaring into the tens of billions. Simultaneously, adversaries target the IoT devices for the lack of security mechanisms. To mitigate such threats, we need to understand the IoT malware ecosystem and underlying dynamics.

In this research, we perform IP and network centric analysis to investigate affinities among the endpoints. We analyze the $\approx$78.2% of total responsive public IPv4 endpoints among dropzones and their targets as extracted from IoT software and spread across the globe from diverse perspectives. First, we analyze the dropzone-target inter-relationship and look for the affinity between them. We observe that the list of targets is shared between attackers, or are compiled using similar conditions on IoT search engines like Shodan or Censys. For our geographical analysis, we comprehensively analyzed the distribution of the number of dropzones and targets (country-, state-, and city-level). We visualize the target areas representing dropzone locations and their size scaled by the number of associated targets. Also, we analyze the attack exposure of endpoints and correlate to the risk they possess. These endpoints need to be carefully and individually analyzed to extract patterns for predicting the chances of them being compromised.

Moreover, we utilized IoT search engines to facilitate network penetration analysis of the dropzones and target IPs. We extract information such as the organization, the number of active ports, and vulnerabilities associated with the IP address. Knowing which network ports are open for a particular IP address can potentially allow attackers to exploit them for DDoS attacks, the usage should be limited. Seeing the number of vulnerabilities associated with dropzone IP addresses, for example, reveals the level of risk involved and which malware instances they are associated with. Our analysis also sheds light on the tolerance of organizations towards the endpoints with malicious intent. The tug of war between the profit and the loss of trust among the users (prospective domain buyers) deserve the focus of the community.

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 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.

We propose a defense method to suggest to countries, where enterprises can use this information to improve their security. In the academic space, this comprehensive analysis is lacking, and is likely to stimulate further explorations. We sum up our research results, we choose suspicious ports. Moreover, we analyze open ports distributed by country, and vulnerabilities of each port. Through this analysis, we can block ports by country. We can expect the mitigation strategies through these port blocks, can be used by companies in each country.

APPENDIX A: COPYRIGHT INFORMATION

ACM Copyright and Audio/Video Release

Title of the Work: IoT Malware Ecosystem in the Wild: Analysis and Exposures
Submission ID:sedgec19-p127
Author/Presenter(s): Jinchun Choi; Afsah Anwar; Hisham Alasmary; Jeff Spaulding; Daehun Nyang; Aziz Mohaisen
Type of material: Full Paper

Publication and/or Conference Name: SEC '19: The Fourth ACM/IEEE Symposium on Edge Computing Proceedings

I. **Copyright Transfer, Reserved Rights and Permitted Uses**

* Your Copyright Transfer is conditional upon you agreeing to the terms set out below.

Copyright to the Work and to any supplemental files integral to the Work which are submitted with it for review and publication such as an extended proof, a PowerPoint outline, or appendices that may exceed a printed page limit, (including without limitation, the right to publish the Work in whole or in part in any and all forms of media, now or hereafter known) is hereby transferred to the ACM (for Government work, to the extent transferable) effective as of the date of this agreement, on the understanding that the Work has been accepted for publication by ACM.

Reserved Rights and Permitted Uses

(a) All rights and permissions the author has not granted to ACM are reserved to the Owner, including all other proprietary rights such as patent or trademark rights.

(b) Furthermore, notwithstanding the exclusive rights the Owner has granted to ACM, Owner shall have the right to do the following:

(i) Reuse any portion of the Work, without fee, in any future works written or edited by the Author, including books, lectures and presentations in any and all media.

(ii) Create a "Major Revision" which is wholly owned by the author

(iii) Post the Accepted Version of the Work on (1) the Author's home page, (2) the Owner's institutional repository, (3) any repository legally mandated by an agency funding the research on which the Work is based, and (4) any non-commercial repository or aggregation that does not duplicate ACM tables of contents, i.e., whose patterns of links do not substantially duplicate an ACM-copyrighted volume or issue. Non-commercial repositories are here understood as repositories owned by non-profit organizations that do not charge a fee for accessing deposited articles and that do not sell advertising or otherwise profit from serving articles.

(iv) Post an "Author-Izer" link enabling free downloads of the Version of Record in the ACM Digital Library on (1) the Author's home page or (2) the Owner's institutional repository;

(v) Prior to commencement of the ACM peer review process, post the version of the Work as submitted to ACM ("Submitted Version" or any earlier versions) to non-peer reviewed servers;

(vi) Make free distributions of the final published Version of Record internally to the Owner's employees, if applicable;

(vii) Make free distributions of the published Version of Record for Classroom and Personal Use;

(viii) Bundle the Work in any of Owner's software distributions; and

(ix) Use any Auxiliary Material independent from the Work. (x) If your paper is withdrawn before it is published in the ACM Digital Library, the rights revert back to the author(s).

When preparing your paper for submission using the ACM TeX templates, the rights and permissions information and the bibliographic strip must appear on the lower left hand portion of the first page.

The new ACM Consolidated TeX template Version 1.3 and above automatically creates and positions these text blocks for you based on the code snippet which is system-generated based on your rights management choice and this particular conference.

NOTE: For authors using the ACM Microsoft Word Master Article Template and Publication Workflow, The ACM Publishing System (TAPS) will add the rights statement to your papers for you. Please check with your conference contact for information regarding submitting your source file(s) for processing.

Please copy and paste \setcopyright{acmcopyright} before \begin{document} and please copy and paste the following code snippet into your TeX file between \begin{document} and \maketitle, either after or before CCS codes.

```
\copyrightyear{2019}
\acmYear{2019}
\acmConference[SEC 2019]{The Fourth ACM/IEEE Symposium on Edge Computing}{November
7--9, 2019}{Arlington, VA, USA}
\acmBooktitle{The Fourth ACM/IEEE Symposium on Edge Computing (SEC 2019), November
7--9, 2019, Arlington, VA, USA}
\acmPrice{15.00}
\acmDOI{10.1145/3318216.3363379}
\acmISBN{978-1-4503-6733-2/19/11}
```

If you are using the ACM Microsoft Word template, or still using an older version of the ACM TeX template, or the current versions of the ACM SIGCHI, SIGGRAPH, or SIGPLAN TeX templates, you must copy and paste the following text block into your document as per the instructions provided with the templates you are using:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

SEC 2019, November 7–9, 2019, Arlington, VA, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6733-2/19/11…$15.00
https://doi.org/10.1145/3318216.3363379

NOTE: Make sure to include your article's DOI as part of the bibstrip data; DOIs will be registered and become active shortly after publication in the ACM Digital Library. Once you have your camera ready copy ready, please send your source files and PDF to your event contact for processing.

☑ A. Assent to Assignment. I hereby represent and warrant that I am the sole owner (or authorized agent of the copyright owner(s)), with the exception of third party materials detailed in section III below. I have obtained permission for any third-party material included in the Work.

□ B. Declaration for Government Work. I am an employee of the National Government of my country and my Government claims rights to this work, or it is not copyrightable (Government work is classified as Public Domain in U.S. only)

Are any of the co-authors, employees or contractors of a National Government? ○ Yes ● No

---

II. Permission For Conference Recording and Distribution

* Your Audio/Video Release is conditional upon you agreeing to the terms set out below.

I hereby grant permission for ACM to include my name, likeness, presentation and comments in any and all forms, for the Conference and/or Publication.

I further grant permission for ACM to record and/or transcribe and reproduce my presentation as part of the ACM Digital Library, and to distribute the same for sale in complete or partial form as part of an ACM product on CD-ROM, DVD, webcast, USB device, streaming video or any other media format now or hereafter known.

I understand that my presentation will not be sold separately as a stand-alone product without my direct consent. Accordingly, I give ACM the right to use my image, voice, pronouncements, likeness, and my name, and any biographical material submitted by me, in connection with the Conference and/or Publication, whether used in excerpts or in full, for distribution described above and for any associated advertising or exhibition.

Do you agree to the above Audio/Video Release? ● Yes ○ No

III. Auxiliary Material

Do you have any Auxiliary Materials? ○ Yes ● No

IV. Third Party Materials

In the event that any materials used in my presentation or Auxiliary Materials contain the work of third-party individuals or organizations (including copyrighted music or movie excerpts or anything not owned by me), I understand that it is my responsibility to secure any necessary permissions and/or licenses for print and/or digital publication, and cite or attach them below.

● We/I have not used third-party material.
○ We/I have used third-party materials and have necessary permissions.

V. Artistic Images

If your paper includes images that were created for any purpose other than this paper and to which you or your employer claim copyright, you must complete Part V and be sure to include a notice of copyright with each such image in the paper.

● We/I do not have any artistic images.
○ We/I have any artistic images.

VI. Representations, Warranties and Covenants

The undersigned hereby represents, warrants and covenants as follows:

(a) Owner is the sole owner or authorized agent of Owner(s) of the Work;

(b) The undersigned is authorized to enter into this Agreement and grant the rights included in this license to ACM;

(c) The Work is original and does not infringe the rights of any third party; all permissions for use of third-party materials consistent in scope and duration with the rights granted to ACM have been obtained, copies of such permissions have been provided to ACM, and the Work as submitted to ACM clearly and accurately indicates the credit to the proprietors of any such third-party materials (including any applicable copyright notice), or will be revised to indicate such credit;

(d) The Work has not been published except for informal postings on non-peer reviewed servers, and Owner covenants to use best efforts to place ACM DOI pointers on any such prior postings;

(e) The Auxiliary Materials, if any, contain no malicious code, virus, trojan horse or other software routines or hardware components designed to permit unauthorized access or to disable, erase or otherwise harm any computer systems or software; and

(f) The Artistic Images, if any, are clearly and accurately noted as such (including any applicable copyright notice) in the Submitted Version.

☑ I agree to the Representations, Warranties and Covenants

Funding Agents

1. National Research Foundation of Korea award number(s): 2016K1A1A2912757
2. Florida Center for Cybersecurity, University of South Florida award number(s):

DATE: **09/17/2019** sent to mohaisen@ucf.edu at **23:09:40**

IEEE COPYRIGHT AND CONSENT FORM

To ensure uniformity of treatment among all contributors, other forms may not be substituted for this form, nor may any wording of the form be changed. This form is intended for original material submitted to the IEEE and must accompany any such material in order to be published by the IEEE. Please read the form carefully and keep a copy for your files.

Honor Among Thieves: Towards Understanding the Dynamics and Interdependencies in IoT Botnets
Jinchun Choi, University of Central Florida, United States; Ahmed Abusnaina, University of Central Florida, United States; Afsah Anwar, University of Central Florida, United States; An Wang, Case Western Reserve University, United States; Songqing Chen, George Mason University, United States; Daehun Nyang, INHA University, South Korea; and Aziz Mohaisen, University of Central Florida, United States
2019 IEEE Conference on Dependable and Secure Computing (DSC)

COPYRIGHT TRANSFER

The undersigned hereby assigns to The Institute of Electrical and Electronics Engineers, Incorporated (the "IEEE") all rights under copyright that may exist in and to: (a) the Work, including any revised or expanded derivative works submitted to the IEEE by the undersigned based on the Work; and (b) any associated written or multimedia components or other enhancements accompanying the Work.

GENERAL TERMS

1. The undersigned represents that he/she has the power and authority to make and execute this form.
2. The undersigned agrees to indemnify and hold harmless the IEEE from any damage or expense that may arise in the event of a breach of any of the warranties set forth above.
3. The undersigned agrees that publication with IEEE is subject to the policies and procedures of the IEEE PSPB Operations Manual.
4. In the event the above work is not accepted and published by the IEEE or is withdrawn by the author(s) before acceptance by the IEEE, the foregoing copyright transfer shall be null and void. In this case, IEEE will retain a copy of the manuscript for internal administrative/record-keeping purposes.
5. For jointly authored Works, all joint authors should sign, or one of the authors should sign as authorized agent for the others.
6. The author hereby warrants that the Work and Presentation (collectively, the "Materials") are original and that he/she is the author of the Materials. To the extent the Materials incorporate text passages, figures, data or other material from the works of others, the author has obtained any necessary permissions. Where necessary, the author has obtained all third party permissions and consents to grant the license above and has provided copies of such permissions and consents to IEEE

You have indicated that you DO wish to have video/audio recordings made of your conference presentation under terms and conditions set forth in "Consent and Release."

CONSENT AND RELEASE

1. In the event the author makes a presentation based upon the Work at a conference hosted or sponsored in whole or in part by the IEEE, the author, in consideration for his/her participation in the conference, hereby grants the IEEE the unlimited, worldwide, irrevocable permission to use, distribute, publish, license, exhibit, record, digitize, broadcast, reproduce and archive, in any format or medium, whether now known or hereafter developed: (a) his/her presentation and comments at the conference; (b) any written materials or multimedia files used in connection with his/her presentation; and (c) any recorded interviews of him/her (collectively, the "Presentation"). The permission granted includes the transcription and reproduction of the Presentation for inclusion in products sold or distributed by IEEE and live or recorded broadcast of the Presentation during or after the conference.
2. In connection with the permission granted in Section 1, the author hereby grants IEEE the unlimited, worldwide,

irrevocable right to use his/her name, picture, likeness, voice and biographical information as part of the advertisement, distribution and sale of products incorporating the Work or Presentation, and releases IEEE from any claim based on right of privacy or publicity.

BY TYPING IN YOUR FULL NAME BELOW AND CLICKING THE SUBMIT BUTTON, YOU CERTIFY THAT SUCH ACTION CONSTITUTES YOUR ELECTRONIC SIGNATURE TO THIS FORM IN ACCORDANCE WITH UNITED STATES LAW, WHICH AUTHORIZES ELECTRONIC SIGNATURE BY AUTHENTICATED REQUEST FROM A USER OVER THE INTERNET AS A VALID SUBSTITUTE FOR A WRITTEN SIGNATURE.

JINCHUN CHOI

Signature

30-09-2019

Date (dd-mm-yyyy)

Information for Authors

AUTHOR RESPONSIBILITIES

The IEEE distributes its technical publications throughout the world and wants to ensure that the material submitted to its publications is properly available to the readership of those publications. Authors must ensure that their Work meets the requirements as stated in section 8.2.1 of the IEEE PSPB Operations Manual, including provisions covering originality, authorship, author responsibilities and author misconduct. More information on IEEE’s publishing policies may be found at http://www.ieee.org/publications_standards/publications/rights/authorrightsresponsibilities.html Authors are advised especially of IEEE PSPB Operations Manual section 8.2.1.B12: "It is the responsibility of the authors, not the IEEE, to determine whether disclosure of their material requires the prior consent of other parties and, if so, to obtain it." Authors are also advised of IEEE PSPB Operations Manual section 8.1.1B: "Statements and opinions given in work published by the IEEE are the expression of the authors."

RETAINED RIGHTS/TERMS AND CONDITIONS

- Authors/employers retain all proprietary rights in any process, procedure, or article of manufacture described in the Work.
- Authors/employers may reproduce or authorize others to reproduce the Work, material extracted verbatim from the Work, or derivative works for the author's personal use or for company use, provided that the source and the IEEE copyright notice are indicated, the copies are not used in any way that implies IEEE endorsement of a product or service of any employer, and the copies themselves are not offered for sale.
- Although authors are permitted to re-use all or portions of the Work in other works, this does not include granting third-party requests for reprinting, republishing, or other types of re-use.The IEEE Intellectual Property Rights office must handle all such third-party requests.
- Authors whose work was performed under a grant from a government funding agency are free to fulfill any deposit mandates from that funding agency.

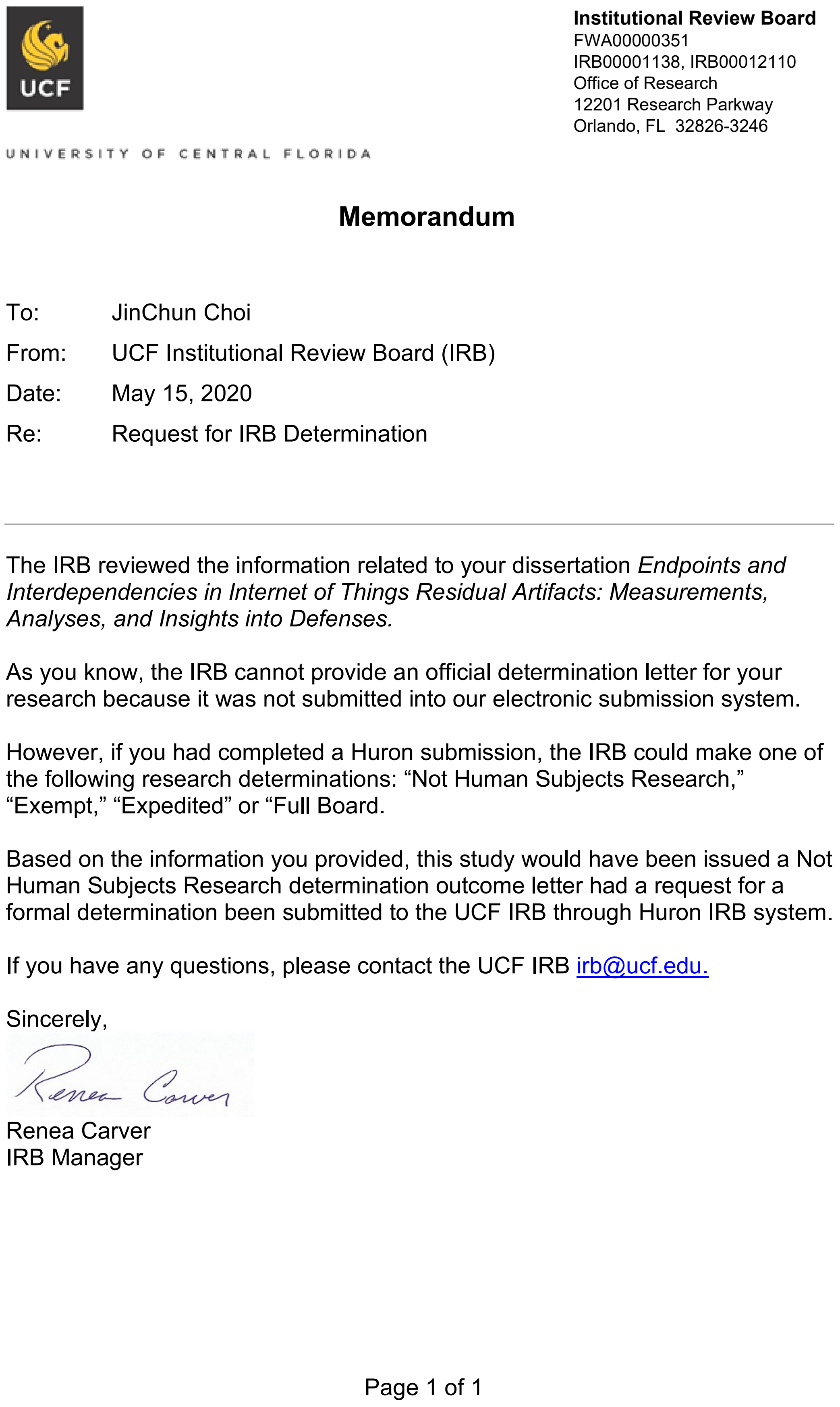
AUTHOR ONLINE USE

- **Personal Servers**. Authors and/or their employers shall have the right to post the accepted version of IEEE-copyrighted articles on their own personal servers or the servers of their institutions or employers without permission from IEEE, provided that the posted version includes a prominently displayed IEEE copyright notice and, when published, a full citation to the original IEEE publication, including a link to the article abstract in IEEE Xplore. Authors shall not post the final, published versions of their papers.
- **Classroom or Internal Training Use.** An author is expressly permitted to post any portion of the accepted version of his/her own IEEE-copyrighted articles on the author's personal web site or the servers of the author's institution or company in connection with the author's teaching, training, or work responsibilities, provided that the appropriate copyright, credit, and reuse notices appear prominently with the posted material. Examples of permitted uses are lecture materials, course packs, e-reserves, conference presentations, or in-house training courses.
- **Electronic Preprints.** Before submitting an article to an IEEE publication, authors frequently post their manuscripts to their

own web site, their employer's site, or to another server that invites constructive comment from colleagues. Upon submission of an article to IEEE, an author is required to transfer copyright in the article to IEEE, and the author must update any previously posted version of the article with a prominently displayed IEEE copyright notice. Upon publication of an article by the IEEE, the author must replace any previously posted electronic versions of the article with either (1) the full citation to the IEEE work with a Digital Object Identifier (DOI) or link to the article abstract in IEEE Xplore, or (2) the accepted version only (not the IEEE-published version), including the IEEE copyright notice and full citation, with a link to the final, published article in IEEE Xplore.

Questions about the submission of the form or manuscript must be sent to the publication's editor.
Please direct all questions about IEEE copyright policy to:
IEEE Intellectual Property Rights Office, copyrights@ieee.org, +1-732-562-3966

APPENDIX B: INSTITUTIONAL REVIEW BOARD LETTER



UNIVERSITY OF CENTRAL FLORIDA

Institutional Review Board
FWA00000351
IRB00001138, IRB00012110
Office of Research
12201 Research Parkway
Orlando, FL 32826-3246

Memorandum

To: JinChun Choi

From: UCF Institutional Review Board (IRB)

Date: May 15, 2020

Re: Request for IRB Determination

The IRB reviewed the information related to your dissertation *Endpoints and Interdependencies in Internet of Things Residual Artifacts: Measurements, Analyses, and Insights into Defenses.*

As you know, the IRB cannot provide an official determination letter for your research because it was not submitted into our electronic submission system.

However, if you had completed a Huron submission, the IRB could make one of the following research determinations: “Not Human Subjects Research,” “Exempt,” “Expedited” or “Full Board.

Based on the information you provided, this study would have been issued a Not Human Subjects Research determination outcome letter had a request for a formal determination been submitted to the UCF IRB through Huron IRB system.

If you have any questions, please contact the UCF IRB irb@ucf.edu.

Sincerely,

Renea Carver

Renea Carver
IRB Manager

LIST OF REFERENCES

[1] A new DDoS reflection attack: Portmapper; an early warning to the industry. `https://bit.ly/2veV3ZM`, Aug 2015.

[2] AbuseIPDB. `https://bit.ly/2mZ3DZ3`, 2018.

[3] Censys landing page. `https://censys.io`, 2018.

[4] Choopa NJ Datacenter. `https://bit.ly/2AqKVTF`, 2018.

[5] Colocrossing: Buffalo Colocation. `https://bit.ly/2LJAvmH`, 2018.

[6] HOSTKEY. `https://www.hostkey.com/about/about`, 2018.

[7] IOFLOOD. `https://bit.ly/2vmT1XA`, 2018.

[8] IPinfo, IP address API and data solutions. `https://ipinfo.io`, 2018.

[9] National vulnerability database. `https://nvd.nist.gov`, 2018.

[10] Port 111 details. `https://bit.ly/2NTS7cP`, 2018.

[11] Radare2. `https://rada.re/r/`, 2018.

[12] Shodan landing page. `https://www.shodan.io`, 2018.

[13] Ultratools free IP tools. `https://bit.ly/2v2cLk4`, 2018.

[14] Wowrack. `https://bit.ly/2M0THci`, 2018.

[15] A. Abusnaina, A. Khormali, H. Alasmary, J. Park, A. Anwar, and A. Mohaisen. Adversarial Learning Attacks on Graph-based IoT Malware Detection Systems. In *39th IEEE International Conference on Distributed Computing Systems, ICDCS 2019, Dallas, TX, USA, July 7*, volume 10, page 2019, 2019.

[16] 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)*, pages 877–882, 2016.

[17] 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)*, pages 271–276. IEEE, 2017.

[18] H. Alasmary, A. Abusnaina, R. Jang, M. Abuhamad, A. Anwar, D. NYANG, and D. Mohaisen. Soteria: Detecting adversarial examples in control flow graph-based malware classifiers. In *40th IEEE International Conference on Distributed Computing Systems, ICDCS*, pages 1296–1305, 2020.

[19] 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*, pages 259–272. Springer, 2018.

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

[21] H. S. Anderson, J. Woodbridge, and B. Filar. DeepDGA: Adversarially-tuned domain generation and detection. In *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2016*, pages 13–21, 2016.

[22] 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)*, pages 1093–1110, 2017.

[23] 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 *Proceedings of the 21st USENIX Security Symposium*, pages 491–506, 2012.

[24] 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*, 19(3):727, 2019.

[25] J. Choi, A. Abusnaina, A. Anwar, A. Wang, S. Chen, D. Nyang, and A. Mohaisen. Honor among thieves: Towards understanding the dynamics and interdependencies in iot botnets. In *2019 IEEE Conference on Dependable and Secure Computing (DSC)*, pages 1–8. IEEE, 2019.

[26] J. Choi, A. Anwar, H. Alasmary, J. Spaulding, D. Nyang, and A. Mohaisen. Iot malware ecosystem in the wild: A glimpse into analysis and exposures. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, pages 413–418, 2019.

[27] E. Cozzi, M. Graziano, Y. Fratantonio, and D. Balzarotti. Understanding Linux malware. In *IEEE Symposium on Security and Privacy, S&P*, pages 161–175, 2018.

[28] C. Crane. 20 Essential Internet of Things Statistics to Keep You Up to Speed. `https://bit.ly/2Lqfw6k`, Sep 2019.

[29] CVE. `https://bit.ly/1FxfOi2`, 2018.

[30] 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.*, pages 807–816, 2017.

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

[32] Developers. CyberIOCs. Available at [Online]: `https://freeiocs.cyberiocs.pro/`, 2019.

[33] Developers. The ultimate packer for executables. Available at [Online]: `https://upx.github.io/`, 2019.

[34] Developers. VirusTotal. Available at [Online]: `https://www.virustotal.com`, 2019.

[35] Developers. How Does an IP Address Give Away Your Location? Available at [Online]: `https://whatismyipaddress.com/geolocation`, 2020.

[36] S. Dharmaraj. IoT, digital transformation require Vietnam to focus on cybersecurity. `https://bit.ly/3bSIYgj`, Nov 2019.

[37] D.L. Mills. Network Time Protocol (NTP). RFC 958, Sep 1985.

[38] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman. A search engine backed by Internet-wide scanning. In *Proceedings of the 22nd ACM Conference on Computer and Communications Security, CCS*, pages 542–553, 2015.

[39] X. Feng, Q. Li, H. Wang, and L. Sun. Acquisitional rule-based engine for discovering Internet-of-Things devices. In *Proceedings of the 27th USENIX Security Symposium*, pages 327–341, 2018.

[40] D. Formby, P. Srinivasan, A. Leonard, J. Rogers, and R. A. Beyah. Who's in control of your control system? device fingerprinting for Cyber-Physical Systems. In *Network and Distributed System Security Symposium, NDSS*, 2016.

[41] B. Genge and C. Enachescu. Shovat: Shodan-based vulnerability assessment tool for Internet-facing services. *Security and Communication Networks*, 9(15):2696–2714, 2016.

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

[43] J. Graham-Cumming. How to Launch and Defend Against DDoS. https://bit.ly/2K4sOlP, Oct 2013.

[44] B. Herzberg, D. Bekerman, and I. Zeifman. Breaking down mirai: An IoT DDoS botnet analysis. https://bit.ly/2dQbvYo, Oct 2016.

[45] T. Holz, M. Engelberth, and F. Freiling. Learning more about the underground economy: A case-study of keyloggers and dropzones. In *Proceedings of the 14th European Conference on Research in Computer Security*, ESORICS'09, pages 1–18, 2009.

[46] 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.

[47] J. Graham-Cumming. Understanding and Mitigating NTP-based DDoS Attacks. https://bit.ly/2ifu8pa, Jan 2014.

[48] 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)*, pages 1–5. IEEE, 2017.

[49] K. Kendall and C. McMillan. Practical malware analysis. In *Black Hat Conference, USA*, page 10, 2007.

[50] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas. DDoS in the IoT: Mirai and other botnets. *Computer*, 50(7):80–84, 2017.

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

[52] E. Leloglu. A review of security concerns in Internet of Things. *Journal of Computer and Communications*, 5(01):121, 2016.

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

[54] 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)*, pages 813–818. IEEE, 2018.

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

[56] P. Middleton. Forecast Analysis: Internet of Things–Endpoints, Worldwide, 2016 Update. `http://gtnr.it/2oRo4aN`, 2016.

[57] V. K. Mikhail Kuzin, Yaroslav Shmelev. Number of data centers in the United States by location and type in 2018. `https://bit.ly/2WV26Vg`, Sep 2018.

[58] Mitre. Common Vulnerabilities and Exposures. `https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-10372`, May 2017.

[59] A. Mohaisen and O. Alrawi. AMAL: high-fidelity, behavior-based automated malware analysis and classification. In *Proc. of WISA*, 2014.

[60] A. Mohaisen and O. Alrawi. Av-meter: An evaluation of antivirus scans and labels. In *Proc. of DIMVA*, 2014.

[61] 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)*, pages 421–430. IEEE, 2007.

[62] J. MSV. Security is fast becoming the achilles heel of consumer Internet of Things. `https://bit.ly/2Ovco9B`, Nov 2016.

[63] 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*, page 4. ACM, 2018.

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

[65] J. Ouellette, A. Pfeffer, and A. Lakhotia. Countering malware evolution using cloud-based learning. In *Proceedings of the 8th International Conference on Malicious and Unwanted Software: "The Americas", MALWARE*, pages 85–94, 2013.

[66] Y. M. P. 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*, 24(3):522–533, 2016.

[67] J. P. Paro. Internet of Things: The Global Regulatory Ecosystem and the Most Promising Smart Environments Part II. `https://bit.ly/3cTqIod`, Aug 2019.

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

[69] R. Srinivasan. Binding Protocols for ONC RPC Version 2. RFC 1833, Aug 1995.

[70] M. Z. Rafique and J. Caballero. FIRMA: malware clustering and network signature generation with mixed network behaviors. In *Proceedings of the 16th International Symposium on Research in Attacks, Intrusions, and Defenses, RAID*, pages 144–163, 2013.

[71] M. Sebastián, R. Rivera, P. Kotzias, and J. Caballero. AVclass: A tool for massive malware labeling. In *Research in Attacks, Intrusions, and Defenses - 19th International Symposium, RAID 2016, Paris, France, September 19-21, 2016, Proceedings*, pages 230–253, 2016.

[72] F. Shen, J. D. Vecchio, A. Mohaisen, S. Y. Ko, and L. Ziarek. Android malware detection using complex-flows. In *37th IEEE International Conference on Distributed Computing Systems, ICDCS 2017, Atlanta, GA, USA, June 5-8, 2017*, pages 2430–2437, 2017.

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

[74] A. Sivanathan, D. Sherratt, H. H. Gharakheili, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman. Characterizing and classifying IoT traffic in smart cities and campuses. In *Proceedings of the 2017 IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS)*, pages 559–564, 2017.

[75] 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)*, pages 519–529. IEEE, 2017.

[76] S. Soltan, P. Mittal, and H. V. Poor. BlackIoT: IoT botnet of high wattage devices can disrupt the power grid. In *Proceedings of the 27th USENIX Security Symposium*, pages 15–32, 2018.

[77] SpeedGuide.net. Ports Database. `https://www.speedguide.net/ports.php`, May 2020.

[78] J. Su, D. V. Vargas, S. Prasad, D. Sgandurra, Y. Feng, and K. Sakurai. Lightweight classification of IoT malware based on image recognition. *arXiv preprint arXiv:1802.03714*, 2018.

[79] B. Tuttle. Why You Shouldn't Trust Positive Online Reviews—Or Negative Ones, For That Matter. `https://ti.me/1KoyHFo`, Aug 2012.

[80] UDP Flood Attacks. `https://bit.ly/2NTNNu6`, 2018.

[81] N. Vlajic and D. Zhou. IoT as a land of opportunity for DDoS hackers. *Computer*, 51(7):26–34, 2018.

[82] 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*, pages 176–186. Springer, 2017.

[83] A. G. West and A. Mohaisen. Metadata-driven threat classification of network endpoints appearing in malware. In *Proceedings of the 11th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, DIMVA*, pages 152–171, 2014.

[84] C. Willems, T. Holz, and F. C. Freiling. Toward automated dynamic malware analysis using cwsandbox. *IEEE Security & Privacy*, 5(2):32–39, 2007.

[85] R. Williamson. Weekly web security warning - Mirai worm crawls to the top. `https://bit.ly/2At2cf8`, Feb 2018.

[86] M. Ye, N. Jiang, H. Yang, and Q. Yan. Security analysis of Internet-of-Things: A case study of august smart lock. In *Proceedings of the 2017 IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS*, pages 499–504, 2017.