MalNet: A binary-centric network-level profiling of IoT Malware

Ali Davanian
University of California Riverside
adava003@ucr.edu

Michalis Faloutsos
University of California Riverside
michalis@cs.ucr.edu

Abstract

Where are the IoT C2 servers located? What vulnerabilities does IoT malware try to exploit? What DDoS attacks are launched in practice? In this work, we conduct a large scale study to answer these questions. Specifically, we collect and dynamically analyze 1447 malware binaries on the day that they become publicly known between March 2021 and March 2022 from VirusTotal and MalwareBazaar. By doing this, we are able to observe and profile their behavior at the network level including: (a) C2 communication, (b) proliferation, and (c) issued DDoS attacks. Our comprehensive study provides the following key observations. First, we quantify the elusive behavior of C2 servers: 91% of the time a server does not respond to a second probe four hours after a successful probe. In addition, we find that 15% of the live servers that we find are not known by threat intelligence feeds available on VirusTotal. Second, we find that the IoT malware relies on fairly old vulnerabilities in its proliferation. Our binaries attempt to exploit 12 different vulnerabilities with 9 of them more than 4 years old, while the most recent one was 5 months old. Third, we observe the launch of 42 DDoS attacks that span 8 types of attacks, with two types of attacks targeting gaming servers. The promising results indicate the significant value of using a dynamic analysis approach that includes active measurements and probing towards detecting and containing IoT botnets.

ACM Reference Format:
Ali Davanian and Michalis Faloutsos. 2022. MalNet: A binary-centric network-level profiling of IoT Malware. In *ACM Internet Measurement Conference (IMC '22), October 25–27, 2022, Nice, France.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3517745.3561463

IMC '22, October 25–27, 2022, Nice, France

ACM ISBN 978-1-4503-9259-4/22/10…$15.00
https://doi.org/10.1145/3517745.3561463

1 Introduction

A critical component in combating Internet of Things (IoT) malware is timeliness: Indicators of Compromise (IoC) and signatures need to be made available as fast as possible. Although this is true for malware in general, this is more urgent for IoT devices, which have significantly less built-in protection. For example, home sensors and industrial controllers are typically not protected by on-device defenses such as anti-virus software. Therefore, they rely evermore on firewalls and blacklists. In addition, if we identify the command and control (C2) servers or the exploits that the malware will use, we can improve our defense by hardening, spying and even subverting the botnets. For example, Internet Service Providers (ISPs) and law enforcement can block and take down these C2 servers to disrupt the botnet [26, 28]. In addition, several recent works argue for the need for IoT specific studies and countermeasures [11, 12, 16].

Problem: How much information can we extract from a newly found IoT malware binary? This is the question that motivates our work. Our goal is to reveal a missed opportunity: malware samples are captured and made publicly available through services, such as MalwareBazaar, but are not really used to profile IoT malware network traffic in a systematic and timely way. Specifically, the input to the problem is a malware binary, and the desired output is a comprehensive profile that includes: (a) its C2 server communication, (b) its proliferation techniques, and (c) its attacks as they are being launched.

Previous work: Our work has several key differences with prior efforts, which we can classify in the following categories. First, several efforts analyze the non-network behavior of the IoT malware [13, 14]; in contrast, we are only focused on the network behaviors. Second, several efforts focus on only one of the three categories of IoT malware network traffic that we stated above. For example, several of the related works in this category analyze the C2 communication [18, 29, 30, 36, 39]. Another group within this category, explores the proliferation behaviors of the malware [6, 23, 27]. The last group in this category study DDoS attacks [24, 25, 32]. Third, several related works analyze multiple aspects of the IoT malware network characteristics [7, 16, 19, 21, 33]. These work either are not binary-centric (they only look at traffic and network addresses), or do not

Dataset Name	Size	Methodology	Misc
D-Samples	1447	Daily Collection from VT and MalwareBazaar	MIPS samples for both C2 and P2P malware with the following YARA and AVCLass2 labels: Mirai, Gafgyt, Tsunami, Daddyl33t, VPNFilter, Mozi, Hajime (see Appendix C for a description of these malware)
D-C2s	1160	CnCHunter and VT	C2 addresses found by CnCHunter and cross verified with VT and manually
D-PC2	448	Probing using CnCHunter	Traffic of 7 C2s on a 4 hours interval recorded for 2 weeks
D-Exploits	197	Handshaker	Exploits found by completing the handshake with malware when targeting a victim
D-DDOS	42	Spying IoT C2 commands	Traffic of DDoS commands and the attacks launched by malware

Table 1. The datasets used in this measurement study

provide a holistic view on all three aspects. We elaborate on significant and subtle differences between our work and prior efforts in section 7.

Contribution: We conduct an extensive and systematic daily analysis of the newly reported IoT malware binaries by VirusTotal and MalwareBazaar. Note that, unlike traffic analysis studies, a binary-centric study can create a holistic picture of the IoT malware with full attribution. In other words, we can connect a binary and its family, with a live C2 server, a set of proliferation techniques, and even actual launched DDoS attacks including their type of attack and the target. Our study is focused on timeliness in two ways. First, we collect binaries as soon as they become available from VirusTotal and MalwareBazaar for a year (March 2021 - March 2022). Second, we dynamically analyze these binaries *on the day* that we capture them. Overall, we collect and analyze 1447 malware binaries, which seem to cover seven major malware families as we will see in Table 1. We use two approaches to analyze the malware dynamically: (a) ***observational***, where we let the malware contact its own server, and (b) ***active probing***, where we redirect the C2 communication to potential targets in search of live servers.

We highlight key results from our study. A key goal is to provide a proof of concept for the value of a measurement approach: binary-centric and focused on timeliness.

a. Capturing the ephemeral and elusive behavior of C2 servers. We study the spatiotemporal properties of live C2 servers. First, we find that the responsiveness of live C2 servers is spotty: the servers never respond to all six probes within a day. In fact, 91% of the time a server does not respond to a second probe four hours after a successful probe. Second, we find that the *observed* lifespan for close to two thirds of the servers is one day. Third, we find that 15% of the live servers that we identify are not known to threat intelligence feeds, which could be partly caused by the elusive nature of the servers.

b. Proliferation techniques use old vulnerabilities. Our binaries attempt to exploit 12 different vulnerabilities with 9 of them more than 4 years old. Even the most recent vulnerability was 5 months old. On the one hand, this is an optimistic result: finding ways to defend against well-known and old vulnerabilities could safeguard our IoT devices. On the other hand, this could be an indication that IoT devices are so vulnerable that hackers don't have to try hard to compromise them.

c. Eavesdropping on live DDoS attacks. After connecting to live C2 servers, we capture the launch of 42 DDoS attacks, as evidenced by the command received from their C2 server. We obtain the target addresses and we identify 8 types of attacks with two types of attacks targeting gaming servers. Furthermore, we find that three target IP addresses were within networks owned by Google, Amazon and Roblox.

Potential Impact and large-scale deployment. Our comprehensive profiling of freshly-caught binaries can help: (a) secure the network, through firewall rules, (b) harden the security of the device, and (c) provide intelligence of attacks as they launch. The key goal of this work is to show the significance of a binary-centric and timely dynamic analysis of malware. Our preliminary results show the type and value of the information that can be extracted. Our goal is to expand the scope of the study in the future into a large-scale continuous IoT malware monitoring infrastructure. Achieving this will require: (a) expanding the sandbox capability to activate binaries efficiently by emulating different host devices, and (b) develop techniques to profile the collected information into easy to use rules for different firewall technologies. Using our approach within a large and ongoing measurement effort can provide significant value.

Open sourcing and sharing. Our group is committed and has a track record of sharing tools and our data openly.

In fact, this was a key reason for leveraging and expanding existing open-source tools [1].

This work was supported by NSF SaTC Grant No. 2132642.

2 Methodology and Datasets

In this section, we explain our experimental set up, and the methodology for establishing the datasets that we use in our study.

2.1 Experimental setup: our sandbox

In our dynamic analysis, we activate the malware binary in a sandbox. Among the various tools, we selected CnCHunter [17], a powerful open-source tool for analyzing IoT malware binaries. Leveraging the capabilities of the tool, we conduct experiments in two different modes.

In the first mode, we find the referred C2 servers in the binary as follows. We emulate the execution of the malware binary using QEMU [9]. In this study, we focus on MIPS 32 bit CPU architecture as our focus is on IoT malware. MIPS is a Reduced Instruction Set Computer (RISC) Instruction Set Architecture (ISA) that is popular by IoT vendors. We use the short form MIPS 32B binary to refer to the executable for the MIPS 32 bit architecture. Considering additional types of malware would require extending the capabilities of our sandbox to other CPU architectures, which we intend to do in the future. After the emulation, we analyze the network communication of the C2 malware. As reported, we can detect C2-bound traffic with a 90% precision [17].

In the second mode of execution, we weaponize a binary and use it for probing a set of IP:port targets of interest. In this mode, we identify the live C2 servers in the target address space that engage and communicate with the weaponized binary. In both modes of execution, the traffic generated by the malware can be captured in a pcap format. We discuss the limitation of our sandboxing approach in section 6.

2.2 Creating the malware binary dataset

We use VirusTotal [1] and MalwareBazaar[4] malware feeds to collect our malware samples. VirusTotal provides a free online file scanning service where users can submit suspicious files and instantly get the result of analysis from 75 (as of Aug 28, 2022) AntiVirus products [40]. These files will be available for analysis to the premium users. Several prior work has already used VirusTotal datasets [13, 22, 37]. A study [37] reports that malware appear in VirusTotal feeds between few hours to one day in advance compared to other sources. That said, the shared feeds can have delays of up to 24 hours [37]. In contrast to VirusTotal, our second source, MalwareBazaar provides malware samples freely to all users. MalwareBazaar uses open source intelligence tools (OSINT), and integrates data from 18 (as of Aug 28, 2022) sources to provide malware feeds [3].

We collect malware binaries with the following process. First, we collect malware on a daily basis. Every day between March 2021 and March 2022, we collect the new IoT malware binaries released by VirusTotal and MalwareBazaar. Then, we dynamically analyze the new malware binaries on the same day. We were able to collect 1447 MIPS 32B malware binaries, which we refer to as *D-Samples*. Our dataset seems to cover a wide range of malware families as shown in Table 1.

We briefly discuss our approach to verify the nature of the collected binaries. First, we ensure that each binary is malware by getting the corroboration of at least 5 malware detection engines. Note that the threshold of 5 engines is aligned with established best practices [41]. Second, we identify the family of the malware as follows. We use crowd-sourced YARA rules (provided in VirusTotal results) in addition to AVClass2 [35] to identify the malware family labels. Note that the AVClass2 seems to be often unreliable for for MIPS binaries. For example, all the instances of the Mozi family, a peer-to-peer (P2P) malware in our analysis, are wrongly classified as Mirai.

2.3 Profiling IoT C2 addresses

We profile the IoT C2 addresses and create two datasets: (a) *D-C2s* and (b) *D-PC2*.

a. The *D-C2s* dataset. In creating *D-C2s*, our first step is to filter out the P2P samples (mostly Mozi malware family) from our *D-Samples*. Having done that, we have a set of malware samples that would have C2 communication. Next, we use our sandbox to analyze the binaries and find its referred C2 address. Then, we cross validate the results with Virus Total Intelligence feeds by checking whether the reported C2 address (IP or DNS) is malicious. In order to measure the miss rate of the threat intelligence feeds provided by VT, we query VT two times. Once on the day the binary is published, and once on May 7th 2022. If a C2 is reported as malicious by the second query, but not by the first one, we consider it a miss. Finally, we perform a manual verification of samples that have unverified (by the two VT queries) live C2 addresses. Our manual verification compares the captured traffic with Mirai, Gafgyt, Tsunami and Daddyl33t network protocols. We refer to this dataset of C2 addresses as *D-C2s*, which has 1160 addresses of C2 servers. We also want to understand the temporal behavior of IoT C2 servers. *D-C2s* dataset is not suitable for such a study as it depends on the latency of sharing the IoT malware binaries and it also does not track their online presence of C2 servers over time. For that, we need active probing process that we describe below.

[1] https://github.com/adava/CnCHunter/wiki/MalNet-Datasets

b. The *D-PC2* dataset. The key idea in creating *D-PC2* is to probe a target subnet and a set of ports, and then observe the C2 servers as they go online and go offline. To this end, we conduct an active probing study of 6 sample subnets and 12 ports with past history of malicious activity. These ports are listed in Appendix B. The next step to conduct this study is to select malware samples; we select two samples, one Gafgyt and another one Mirai. We probe the subnets and ports for two weeks on a 4 hours interval basis. In this period, we find 7 C2 servers. At the end of the measurement, we create *D-PC2* dataset that contains 64 traffic measurements per C2 address.

AS Name	ASN	Country	Hosting	Anti DDoS?
ColoCrossing	36352	US	Yes	Yes
Delis LLC	211252	US	N/A	N/A
DigitalOcean	14061	US	Yes	Yes
FranTech Solutions	53667	LU	Yes	Yes
HOSTGLOBAL	202306	RU	Yes	Yes
Serverion LLC	399471	NL	Yes	Yes
OVH SAS	16276	FR	Yes	Yes
IP SERVER LLC	44812	RU	Yes	Yes
Apeiron Global	139884	IN	Yes	No
Serverius	50673	NL	Yes	Yes

Table 2. Information about the top 10 Autonomous Systems that host the C2 IPs (more information in Appendix A).

2.4 Observing exploits and vulnerabilities

To better understand the proliferation behavior of the IoT malware, we extract the exploits from the malware using dynamic analysis and trickery. We trick the malware into sending over its exploits to fake victim targets that we control [6]. Our process is as follows. First, we identify the ports that the malware uses to scan and attack. We select the most popular ports in our experiment based on a threshold on the number of distinct IPs that are contacted for a particular destination port. We choose the value 20 for this threshold, which gives good results as we will see later. Next, in a separate thread, we create a socket on that port and redirect the future traffic to that local port for the next IP addresses. This way, the malware completes the TCP handshake with the fake target and we collect the payload sent by the malware. We call this method handshaker following the nomenclature [23, 36].

We create dataset *D-Exploits* with the exploits that we extract using the handshaker method. Overall, we successfully extract exploits from 197 samples targeting 12 vulnerabilities, which we study in section 4.

2.5 Observing the launch of real DDoS attacks

As we mentioned earlier, we observe the launch of real DDoS attack, including the command from the C2 server, and the traffic generated by the malware in our sandbox. Note that this ability to listen in to the server commands and connect them with the actual attack is possible in our binary-centric study, but not possible in a passive analysis of network traffic as in some prior studies. For instance, a passive solution that sits at the perimeter of a cloud provider is able to detect a DDOS attack and all the originating bots (assuming there is no reflection). However, such a solution can not connect the attack to the C2 server that issued the attack command.

We describe our method in more detail. First, we find malware samples with live C2 servers. For this, we analyze the malware on the day they are first submitted, and watch the C2 communication. If the communication is successful, we let the malware run for 2 hours in a restricted mode (only C2 traffic is allowed). Next, we rely on two methods to find DDoS commands as we explain below.

a. Extracting DDoS commands from known IoT C2 protocols: We build a profile of three IoT malware application layer communication protocols: Mirai, Gafgyt and Daddyl33t. For Mirai, Gafgyt , we build the profiler based on the available source code of these malware families. For Daddyl33t, we reverse engineer the communicated traffic and create the profile. While Mirai employs a binary based protocol, Gafgyt and Daddyl33t use a text based protocol. Using these profiles, we search the communicated C2 traffic for DDoS commands.

b. Extracting DDoS commands based on behavioral heuristics: In order to cover other malware families and new variants, we employ a heuristic detection method to find DDoS commands. We count the number of packets sent to non-C2 IP addresses, and measure rate of packets per second. If this rate is higher than a set threshold, we consider the last issued C2 command as a C2 command and record it. By default, we set the threshold to 100 packets per second in this study based on empirical observation.

After collecting traffic that passes one of the above two filters, we further verify the correctness of the results manually. For the first method, we verify the command by evaluating whether the bot started to send traffic to that given DDoS target continuously. For the second method, we extract the target address and search for string and/or binary representation of the target IP in the last issued C2 command.

We refer to the dataset containing the DDoS commands, and the traffic as *D-DDOS*. This dataset contains the 42 commands issued to 20 different malware samples.

2.6 Ethical Considerations

We are confident that our experiments have not caused any damage and took appropriate precautions in the four types

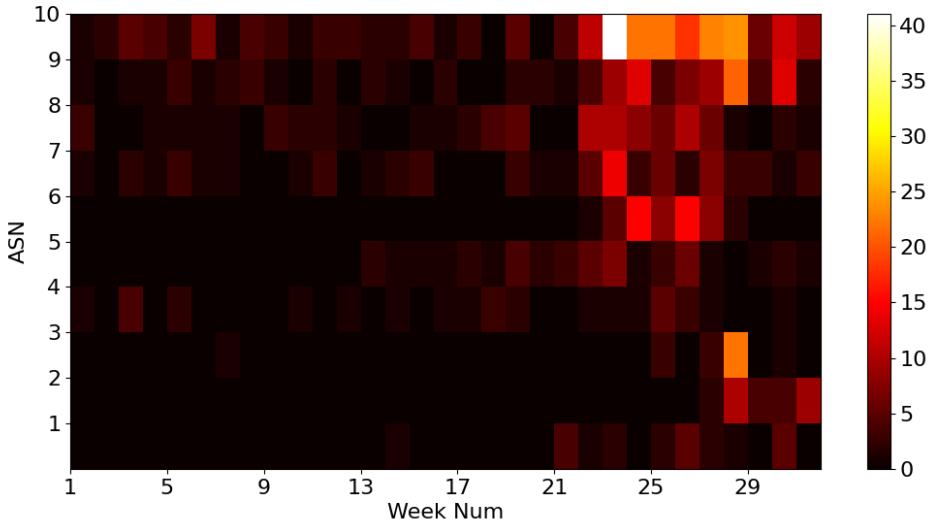

Figure 1. A spatiotemporal view of C2 activity: the heatmap showing the weekly (x-axis) activity of malware-specified C2 servers from our malware feed across the ten most active ASes (y axis) between Mar 2021 and Mar 2022. We don't have data for some weeks either due to the disruption of the service, not observing MIPS 32b samples, or not detecting any C2 server. A complete mapping of the weeks is provided in the Appendix E. The top four ASes are consistently more active with more dark red activity.

of experiments we have. We use SNORT IDS to detect and prevent malicious traffic from leaving our network. Furthermore, we had techniques in place to contain malicious traffic in each of our experiments:

a. Detection of C2s: The tools that we use to detect C2s do not need Internet connection when the malware is analyzed. Thus, this analysis does not interact with the Internet, as we "fake" it to the sandbox. If a sophisticated binary detects that the Internet is not available, we deploy InetSim [2] to simulate services like DNS and http.

b. Detection of exploits: This experiment does not need Internet connection as we fake the victims for the IoT malware. We find the addresses that the malware tries to exploit and complete the handshake with the malware pretending to be those targets. We collect the following data packets that might contain the exploit code.

c. Detection of DDoS targets: Based on the results of (a), we filter out any other communication of the malware to the outside world except to the C2 target. We record the C2 traffic, reverse engineer it and find the DDoS commands and their targets.

c. Probing IP Subnets: We only allow C2 communications like "Call-Home" messages to interact with potential C2 servers. We have manually analyzed sample traffic traces and we have not found any cases of non-C2 communications. Our target subnets were small /24 subnets with a history of malicious activity. We do not send probes if the host does not listen on a port. On live ports, we filter out hosts that present a well-known banner (such as Apache or Nginx).

3 Profiling C2 servers of IoT botnets

In this section, we answer the following questions:

Q1: What is the distribution of the C2 servers across Autonomous Systems (ASes) and how does this evolve over time? (See subsection 3.1)

Q2: Are there common features among the Autonomous Systems that are "popular" with C2 servers? (See subsection 3.1)

Q3: What is the observed lifespan of the IoT C2s? (See subsection 3.2)

Q4: How effective are the VT threat intelligence feeds in terms of comprehensiveness and timeliness? (See subsection 3.3)

3.1 Hosting Environments of IoT C2s

Some ASes are persistently more popular in hosting IoT C2s. We identify and study the Autonomous Systems (ASes) which host C2 servers. First, we want to study their spatiotemporal distribution of C2 servers. In our one year observation period, we find that 10 ASes host 69.7% (more information in Appendix A) of the total number of all C2s servers in our *D-C2s* dataset. These C2s are listed on Table 2. Furthermore, 60% of these ASes consistently appear as top

hosting ASes for IoT C2s on a weekly basis during the one year of observation. The distribution of IoT C2s across the 10 most popular ASes based on the week number of the study is depicted on Figure 1. The distribution shows more C2s since January 2022. This is partially because we get more samples since that date, and partially because the tool we use (CnCHunter) achieves a better activation rate. We see two interesting observations by analyzing Figure 1. First, IP SERVER LLC (AS-44812) and Aperion Global (AS-139884) become more active in the last 4 weeks of the study. On week 28, the highest number of C2s are from AS-44812, which is a Russian ISP. Interestingly, this is the week that Russia invaded Ukraine. A closer investigation will be needed to establish a connection between these two events. Second, we observe a peak of IoT malware samples on week 28, which leads to a peak of observed C2 addresses across all ASes.

There are commonalities among the popular ASes for IoT C2 servers. Although we can not certainly claim why these Autonomous Systems (AS) are more popular for the IoT C2 servers, we observe some common features. We use the information we find on these ASes website with the note that AS211252 does not provide any information on their website. The first common pattern is that they are all hosting providers and offer Dedicated or Virtual Private Servers (VPS) for customers. Second, all of them, except one, provide anti-DDOS services to their customers which is ironic, as they host C2 servers which enable such DDoS attacks. Third, 70% of these service providers are in the USA, Russia and the Netherlands. Fourth, 30% of these providers (AS53667, AS202306 and AS44812) accept cryptocurrency payments that can hide the identity of the payers.

Note that we did not observe a correlation between the size of the ASes or their service ranking and the number of hosted IoT C2 servers. None of these ASes are among the top-100 ASes based on the number of IPv4s they host [8]. In addition, they are not among the top VPS and Dedicated providers [5] either.

The downloader and C2 servers are often on the same server: We analyzed 47 distinct downloader addresses that are referred by the exploits in the *D-Exploits* dataset and only 12 downloader addresses are not identified as C2. All downloader servers host on http port 80.

3.2 The observed lifespan of C2 servers

This section reports on the temporal behavior of the C2 servers.

IoT C2 servers seem to be short-lived and elusive. We provide an analysis of the **observed lifespan** of the C2 servers, which we define as: the interval between the last and the first time we observe a C2 server referred by a sample. We measure the lifespan of C2 servers based on the

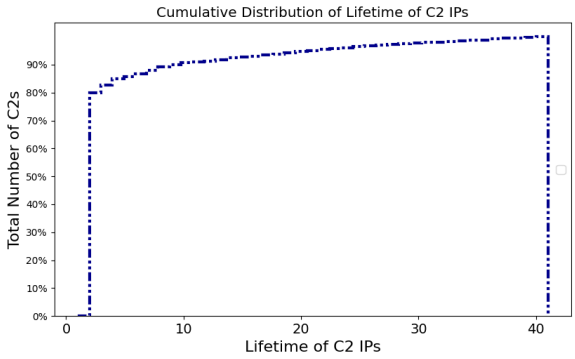

Figure 2. CDF of lifetime of C2 IPs

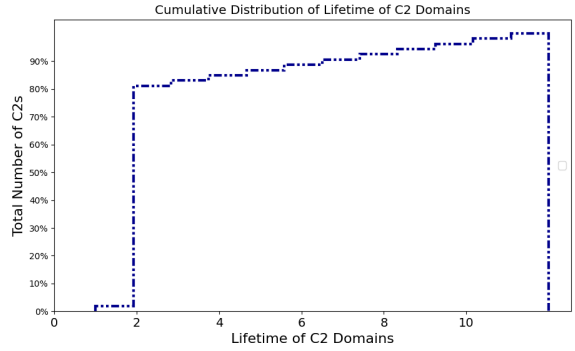

Figure 3. CDF of lifetime of C2 Domains

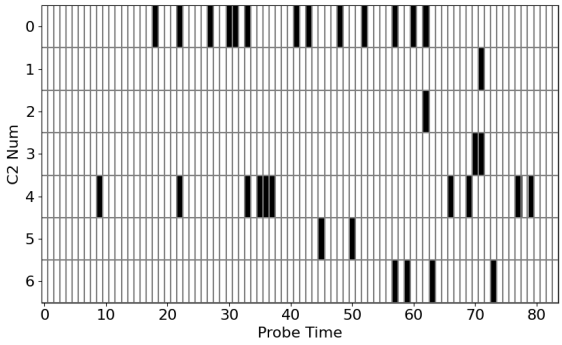

Figure 4. C2 servers are elusive: the responses of C2 servers are spotty to our 6 daily probes in the span of two weeks.

D-C2s and *D-PC2* datasets. We analyze and report our two datasets separately below. The overarching observation is that C2 servers appear short-lived and elusive: servers are not always responding to our active probes.

C2 servers are short-lived. We support this assertion with two observations. First, we measure what percentage of the binaries in *D-C2s* have a live C2 server on the day

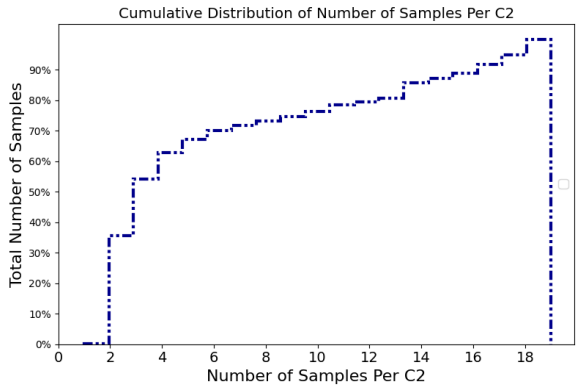

Figure 5. CDF of the number of distinct binaries that use a C2 IP address.

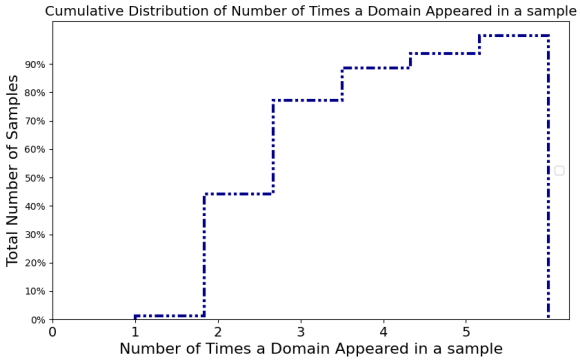

Figure 6. CDF of the number of distinct binaries that use a C2 domain

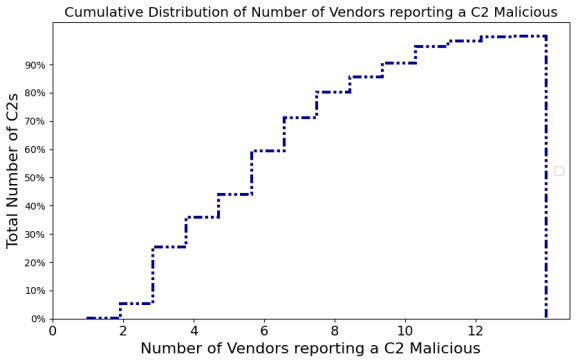

Figure 7. CDF of the number of vendors that report a known C2 server as malicious.

they were reported to the malware repositories. We find that 60% of the samples have a dead C2 server on that day. This could be attributed to the latency in reporting malware binaries. Second, we plot the cumulative distribution of C2 IPs observed lifespan in Figure 2. We see that 80% of the binaries have an observed lifespan of one day while the distribution has a mean lifespan of 4 days. The results are qualitatively similar for DNS-based C2 addresses, which we show in Figure 3.

IoT C2 servers are elusive in terms of responsiveness in our *D-PC2* dataset. We observe an interesting behavior: C2 server responsiveness to our active probing is spotty. In Figure 4, we show the responsiveness of our seven servers in our *D-PC2* dataset. Recall that we probe these servers daily with six probes and we mark as black a probe that receives a response from the C2 server. We see that C2 servers never responded to all six probes in one day. Furthermore, we see that 91% of the time a server does not respond to a second probe four hours after a successful probe. This is a strong indication that IoT C2 servers are not consistently responsive.

In practice, this observation suggests that an active probing study should be "persistent" and probe frequently to ensure accurate detection of live C2 servers.

Type	Same Day	May 7th 2022
All	15.3%	3.3%
IP-based	13.3%	1.5%
DNS-based	57.6%	35.0%

Table 3. The unreported C2 servers: The percentage of C2 servers that security vendors are not aware the day we discover them. Two months after the end of the experiment (May 7th 2022), these C2s are reported as malicious, which provides an indirect validation of our detection approach.

3.3 Threat intelligence effectiveness

In this section, we measure the effectiveness of the threat intelligence (TI) feeds provided by 89 vendors and shared by VT. We use the term effectiveness to refer to comprehensive and timeliness of information. Threat intelligence feeds could play an essential role in mitigating cyber-threats, if they are used in a blacklisting strategy. This is particularly true for IoT devices, which depend on the network perimeter safeguards, as many IoT devices do not have the computation resources to deploy sophisticated security solutions.

First, we find that **60% of C2 servers are contacted by more than one distinct binaries.** We show the cumulative distribution of number of times a C2 is contacted by different samples in *D-C2s* in Figure 5. We see that roughly 40% of C2 IPs are contacted by only one binary, while nearly 20% are contacted by more than 10 distinct binaries. The result for DNS names is similar in Figure 6. This observation shows

that if we detect and "block" a C2 server based on one binary, it could help contain the effect of other binaries that use that server.

Second, we quantify the effectiveness of threat intelligence feeds using our *D-C2s* dataset. We want to measure how many of the C2 addresses we find are known to the intelligence feeds. Specifically, we count a C2 address as a miss, if it is reported not malicious by VirusTotal on the day that we discover it. We ensure that the validity of the C2 address if: (a) it is deemed malicious the second time we query VirusTotal or (b) its behavior matches that of known C2 communication patterns as we outlined in section 2. We point out that this way of measuring effectiveness is motivated by the ephemeral nature of the IoT C2s and the need for timely intelligence for containing bots in practice.

Threat intelligence feeds fail to detect 15% of the C2 servers on the day of discovery of the binary. The result of our measurement is reported in Table 3. Threat intelligence feeds are worse for DNS defined C2 servers compared to servers with IP addresses. In addition, the results suggest that the reason for the miss is the lack of timeliness. Our measurement on May 7th shows that most of the missed C2 addresses will become reported malicious by threat intelligence feeds with a delay as we mentioned earlier. This is significant given the 1 day lifespan of the C2 addresses.

Most threat intelligence feeds miss detecting even the known C2s. The CDF of number of different vendor feeds that report a C2 address as malicious is illustrated in Figure 7. Out of 44 threat intelligence feeds for IoT C2 servers, 25% of the known C2 servers are reported by one or two feeds. This means that either intelligence sharing is absent, or it happens with a lag. Regardless of the reason, the result shows that for lower false negatives, an effective blacklist needs to aggregate data from multiple sources. However, this aggregation needs to be done carefully to avoid increasing the false positives as discussed in recent studies [10, 41].

Feedback from the threat intelligence community. We conducted a survey among threat intelligence vendors. Despite asking a dozen vendors, we only got responses from three of them. First, the vendors did not provide an estimate of their miss rate for their blacklist. Second, the vendors said that a miss rate of more than 10% would be considered unacceptable. Finally, two of the vendors said that an obstacle in detecting C2 servers is **the lack of infrastructure** to execute IoT malware binaries.

4 Profiling IoT Malware Proliferation

In this section, we answer the following questions:

Q5: How recent are the vulnerabilities used by our malware?

Q6: What threat sources should be used in testing IoT devices?

Q7: Are there recent exploitations of disclosed vulnerabilities?

Q8: What are the most popular vulnerabilities based on the number of samples?

IoT malware authors rely on the exploitation of known and old vulnerabilities. We find the exploitation of 14 vulnerabilities that are all known for a while. In more detail, these vulnerabilities and descriptions about them are listed on Table 4. These vulnerabilities are 3 years old on average. Five of these vulnerabilities do not have an assigned CVE number. although they have publicly available exploits. On the other hand, two of the vulnerabilities have CVEs assigned but do not have publicly available exploits.

The more intelligence threat sources the better. None of the popular vulnerability and exploit databases, such as NVD, EDB and OPENVAS, cover all the exploited vulnerabilities. Therefore, a practitioner would need to consider all three sources to ensure a more complete coverage of the vulnerabilities.

IoT malware authors keep adding new exploits but not necessarily for new vulnerabilities. One interesting observation is the additions of two new exploits for CVE-2016-5680 and CVE-2021-45382 compared to a study in 2020 [6]. While CVE-2021-45382 was disclosed after that study, CVE-2016-5680 has been known for 6 years and just recently has become exploited. This confirms a similar finding reported in a recent study [6].

The most popular vulnerabilities are not the newest ones. We rank vulnerabilities based on the number of binaries that use them in our datasets. The top four popular vulnerabilities (CVE-2015-2051, CVE-2018-10561, CVE-2018-10562 and MVPower DVR Shell RCE) are at least 4 years old. We point out that our dataset contains only the recently reported samples, and these are vulnerabilities that were first used by IoT malware in the year they were disclosed [6]. This suggests that despite their age, these vulnerabilities are preferred by hackers. The popularity of old vulnerabilities used by newly captured IoT malware binaries seems to run contrary to the emphasis that we place on the zero-day vulnerabilities

The variance of popularity among vulnerabilities is high. Not surprisingly, we observe that some vulnerabilities are more popular than others. Here we opted for a visual and temporal view of their pupolarity. We show the number of binaries in *D-Exploits* per day that exploit a vulnerability in Figure 8. We see four vulnerabilities that are consistently and heavily used by binaries, while the rest of the vulnerabilities have shorter and less intense usage.

Vendors seem to rarely offer a patch for popular IoT vulnerabilities. We analyzed the availability of patches for 10 of vulnerabilities listed on Table 4 with assigned CVE

ID	Vulnerability	Exploit ID	Publication Date	Target Device	# Samples
1	CVE-2018-10561	EDB-44576	May 3, 2018	GPON Routers	139
1	CVE-2018-10562	EDB-44576	May 3, 2018	GPON Routers	129
2	CVE-2015-2051	EDB-ID-37171	February 23, 2015	D-Link Devices	132
3	CVE-2017-18368	N/A	May 2, 2019	ZyXEL	38
4	Vacron NVR RCE	OPENVAS:1361412562310107187	October 11, 2017	Vacron NVR	46
5	CVE-2017-17215	EDB-43414	March 20, 2018	Huawei Router HG532	1
6	MVPower DVR Shell unauthenticated RCE	EDB-ID-41471	February 27, 2017	MVPower DVR TV-7104HE	74
7	CVE-2021-45382	N/A	December 19, 2021	D-Link DIR-820L command injection	3
8	Linksys unauthenticated RCE	EDB-ID-31683	February 16, 2014	Linksys E-series devices	2
9	WAN Side RCI	EDB-ID-40740	November 8, 2016	Eir D1000 Wireless Router	9
10	CVE-2018-20062	EDB-45978	December 11, 2018	Devices that use ThinkPHP	2
11	CVE-2016-5680	EDB-ID-40200	August 31, 2016	NUUO NVRmini2 / NVRsolo / Crystal Devices / NETGEAR ReadyNAS	1
12	Netlink GPON Router RCE	EDB-48225	March 18, 2020	Netlink GPON Routers	2

Table 4. A description of the vulnerabilities that were exploited by the malware in our *D-Exploits* dataset.

numbers using the vulnerability database vuldb [34]. There are patches available only for 3 of the vulnerabilities (from a single vendor). Five other vulnerabilities can only be mitigated via firewalling. Finally, two of the vulnerabilities can only be mitigated by replacing the device.

Additionally, we analyze the exploits of the vulnerabilities and we identify two patterns. First, most of the exploits are similar, and seem to use the same template with variations on the downloader server address and the loader name. Recall that the loader is the file that downloads the malware and executes on the victim. Second, the peculiarity of the loader names and their frequency suggests that authors use the same loader repeatedly in different exploits. To illustrate this, we plot the frequency of loader names in the *D-Exploits* dataset in Figure 9.

5 Profiling IoT malware attacks

In this section, we answer the following questions:

Q9: What types of DDoS attacks are launched by the IoT malware? (See subsection 5.1)

Q11: What protocols are the targets of IoT malware DDoS attacks? (See subsection 5.2)

Q10: Who are the targets of the IoT malware DDoS attacks? (See subsection 5.3)

We use our *D-DDOS* dataset and track the issuance of DDoS commands from 6 malware variants across three malware families: Mirai (two variants), Gafgyt (two variants) and Daddy33t (two variants). In total, we observe 42 attacks issued by 17 distinct C2 servers to 20 of our malware binaries. The C2 servers are located in 6 different countries. We find that servers in the USA, the Netherlands and the Czech Republic were responsible for 80% of the attacks. We point out that these are not VPN servers (we see no tunnelling setup or traffic encapsulation). Two of the C2 servers (107.174.24.16 and 192.236.248.222) were not listed as malicious by VirusTotal on the day that the attacks were launched, which is aligned with our observations in Table 3. Furthermore, this suggests that if our real-time eavesdropping had translated into actions, one could have actionable just-in-time information to potentially react to these attacks.

Attack-launching C2 servers have longer observed lifespan. Interestingly, the C2 servers that issued DDoS attacks have a longer observed lifespan compared to the rest of C2s in our dataset. The attack-launching servers have an average lifetime of roughly 10 days which is longer than the overall lifespan average of 4 days (see subsection 3.2).

5.1 Types of observed attacks

We observe 8 types of DDoS attacks based on their issued commands. These attacks vary in the way they are mounted, and their target network protocol. We review each attack based on their observed network behavior below. A distribution of these attacks based on the malware family that would use them is depicted in Figure 11.

UDP DDoS Attack: This is the most common type of DDoS attack and appears in all three malware families with different names. In this type of attack, the target is flooded

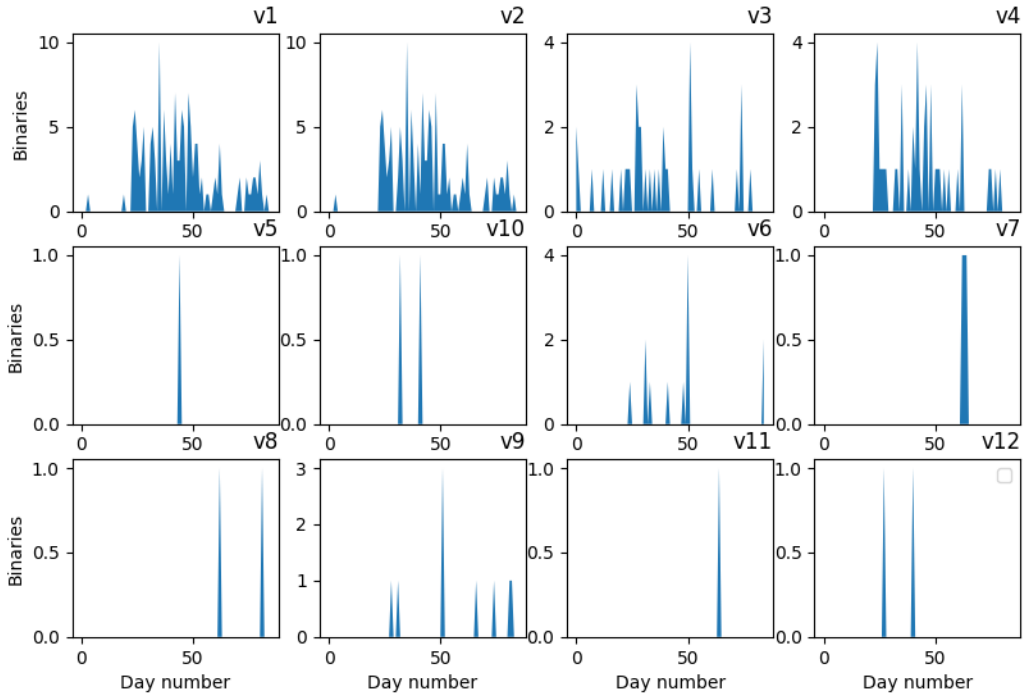

Figure 8. The number of binaries per day that target each one of the 12 vulnerabilities

.

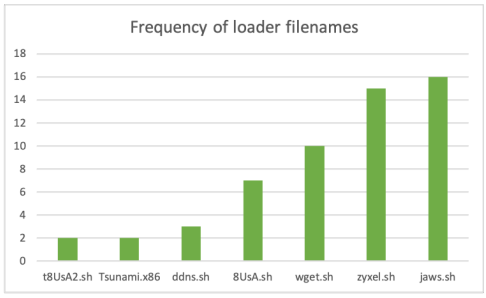

Figure 9. The number of binaries that use each loader file names in our *D-Exploits* dataset.

with continuous packets at one or multiple UDP port(s). Although the implementation of this attack is similar in all three malware families, there are subtle differences that we describe next.

Mirai uses value "0" in the DDOS command to refer to this attack (UDP Flood in Figure 11). The original implementation of this attack published with the source code of Mirai receives a target address, source port, destination address and the time length of the attack and floods the target for the given time length. In our measurement, we saw implementations of Mirai that receive the target IP and port but show different behaviors regarding the choice of the source port. Some variants use the same initial port during the attack, while other variants use multiple source ports. The payload of the attack is the null byte (00h).

By contrast, Gafgyt uses the string *UDP* and daddyl33t uses *UDPRAW* to launch this attack (see Figure 11 for their distribution). Similar to one variant of Mirai, they receive a target address and a target port, and select a source port that remains the same throughout the attack. The payload of the attack is the same as Mirai.

SYN Flood Attack: In this type of attack, a target is flooded at one or multiple TCP port(s) repeatedly with the first packet of the TCP handshake (SYN flag set). We saw instances of this attack launched by daddyl33t and Mirai botnets (*HYDRASYN* and SYN Flood in Figure 11) that we describe below.

In the case of daddyl33t, the C2 server sends a *HYDRASYN* command that includes the target IP and port. The bots attack the target with multiple source ports. In case of Mirai, we saw two different implementations of this attack: (a) multiple source ports targeting the same destination port, and (b) multiple source ports targeting multiple destination ports.

TLS attack: In this type of attack, a service that uses TLS is targeted. The computation on the server side is more

resource intensive compared to the client, and hence it is possible to overwhelm the server. We see two implementation of this attack by daddyl33t and Mirai that we compare below.

Binaries of daddyl33t family seem to target a UDP port possibly running DTLS, and send an encoded message repeatedly. Mirai completes the TCP handshake with the target, sends a large message in different chunks and then sends a RST flag and starts over.

BLACKNURSE attack: This type of DDoS attack targets the ICMP protocol. The bots sends unsolicited IMCP type 3 (Destination Unreachable) packets to the target to overwhelm it. We only see daddyl33t employing this type of attack.

STOMP attack: This attack targets the application layer protocol STOMP that uses TCP transport. The attacker completes the TCP handshake and then floods the target with fake STOMP requests that contain junk data.

VSE attack: This DDoS attack targets the Valve Source Engine of the Steam game platform [38]. It is a UDP amplification attack where the bot sends TSource Engine Query requests to a gaming server. This attack first appeared with the release of Mirai source code, but we see one instance of this attack launched by the Gafgyt malware.

STD attack: In this attack, the target is flooded with random strings. We saw one instance of this attack launched by Gafgyt towards a UDP port. The random string is generated once, and then used repeatedly throughout the attack towards a target.

NFO attack: This attack specifically targets NFO servers hosting vendor that manufactures its own servers [31]. Our claim is based on two observations. First, the target of the attack is an IP address that belongs to this vendor AS. Second, the attack mentions the NFOV6. We are not sure what vulnerability the attack tries to exploit but we see a custom payoload for the attack that targets port 238 UDP on the target IP. This type of attack by IoT malware has been reported before [42]. In our dataset, we see a launch of this attack by daddyl33t.

5.2 DDOS attack traffic

The DDoS attacks we observe target 4 protocols: UDP (excluding DNS), TCP, DNS and ICMP. The distribution of the attacks is illustrated in Figure 10. The vast majority (74%) of the attacks target a service (excluding DNS) on top of the UDP protocol. That said, because of the nature of UDP flood DDoS attacks, we can not certainly say whether a service has been targeted or the target IP. As we mentioned in the previous section, except one case, all attacks receive the target port as part of the attack command. As we don't have access to the C2 code, we speculate that the C2 splits all the

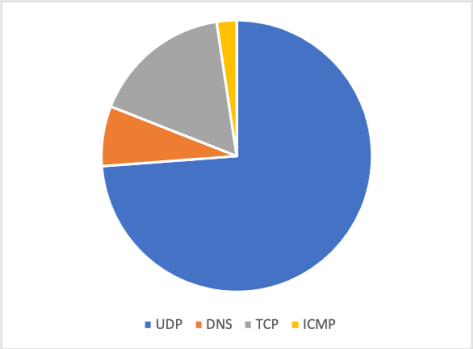

Figure 10. Distribution of DDoS attacks by target protocol: UDP-based attacks are dominant.

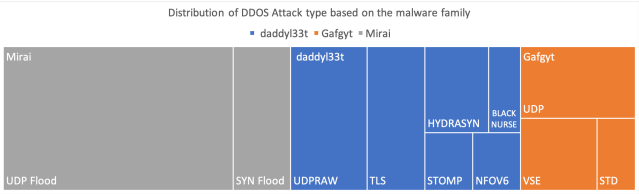

Figure 11. Distribution of DDoS attack type based on the malware family. Mirai (grey) has more attacks, Daddyl33t is second and is more diverse in the types of attacks, and Gafgyt (orange) has fewer attacks.

target UDP ports and then distributes them to the bots. That said, 21% of the attacks target port 80 (mostly on UDP), and 7% target port 443 that are the default ports for the HTTP and HTTPS respectively.

One target hit by multiple attacks. We analyzed the binaries and types of attacks. One interesting observation is that, 25% of the targeted IP addresses are attacked using two different attack types in a single session. For instance, 142.x.x.109 on port 4567 UDP, was once attacked by the TLS, and then shortly after by HYDRASYN. Another example, is target 172.x.x.77, that was first attacked by TLS on port 443 UDP, and then BLACKNURSE targeting the ICMP protocol.

5.3 Targets of the attacks

We observe a few patterns in the targets of the DDoS attacks. We analyze the Autonomous Systems of the target victims to detect patterns of similarity. The first pattern is about the type of AS that the victim is located at. Targets are located in 23 Autonomous Systems that span 11 countries. 45% of these ASes are Internet Service Providers (ISP), and 36% are Hosting providers. The rest of the attacks target businesses, with the main ones being: Google, Amazon and Roblox. These results are aligned with findings in a previous work[32]. An interesting observation is the game industry orientation of the businesses and the hosting providers: 18% of the ASes

are specialized in the computer gaming industry. Figure 12 shows the location of the targets and the type of AS hosts.

6 Discussion and Limitations

We discuss limitations, extensions, and practical issues.

a. Potential Impact: our approach within the cybersecurity ecosystem. We see MalNet as a continuous monitoring effort, which becomes a building block in the ecosystem of tools and services against IoT malware and botnets. The role of our approach within this ecosystem could be as follows: (a) **IoT Honeypots and malware feeds** collect the binaries which they provide to our service; (b) **Public and private sources** exchange information with our service regarding the reputation of IP addresses; (c) **Firewalls and Network Intrusion Detection Systems** incorporate rules and malware signatures provided by our service, (d) **ISPs and web hosters** can use our improved list of malicious actors (bots and C2 servers) to clean up their networks. All of those entities of course could be providing useful information to validate, expand and improve our capabilities.

b. Are our malware binaries relevant and representative? This is the typical question for any empirical study, that can mostly be answered indirectly. First, we collect the malware binaries *as soon as* they are made available by feeds that aggregate a large number of sources. Second, the fact that some of the C2 servers that we find are not already in threat intelligence feeds is an indication that we find reasonably fresh binaries and unknown C2 servers. Third, our malware covers several major malware families, such as Gafgyt, Tsunami, and Mirai, as we saw in Table 1 and discussed in a few places earlier. In a future large scale study, we would like to experiment and compare not only the newly reported binaries, but conduct a longitudinal study of malware of different years.

c. How statistically reliable and generalizable are our observations? This is a question that all measurement studies need to grapple with. First, all our observations are measurement driven and can only describe the observed behavior within the datasets that we collected. With that in mind, we have clearly explained how we collected and created our datasets. Given our interest to analyze newly reported malware, the number of binaries that a study with a historical perspective may have had available is limited. We find that examining the information that newly reported malware has to offer is an important study, which in fact, seems to not have been done in the past in the way we conduct it here.

d. Could this approach be deployed in practice and at a large-scale? The current work provides a proof of concept that shows the promise of our binary-centric approach. Our ambition is to deploy it at large scale, which will have some challenges. Such a deployment would require us to: (a) expand the supported architectures, (b) adapt and continuously update state of the art anti-evasion techniques in our sandbox, and (c) collaborate with ISPs and cloud providers for massive probing. Note that these challenges include some that are active research problems in their own right, and some that are mostly engineering tasks.

e. Can our approach extend to non-IoT malware? The overarching approach applies to any malware that can be activated in a sandbox. In addition, some of our current methods for identifying the C2-bound traffic may need to be adapted in the case of different malware. However, with some customization, the overarching framework could be made to expand to additional types of malware and platforms. Note that our focus on IoT malware is motivated by the fact that IoT malware is newer, less studied but with an increasing presence and potential for harm. This motivation is backed up by our vendors survey that revealed lack of IoT malware execution infrastructure as a main barrier.

f. What would be the potential downsides of using sandboxing as opposed to malware execution on devices? An emulation environment might not accurately model the bare metal device, and this might sometimes impact the correctness of the execution. Additionally, the side effects of an emulation might signal the malware about being analyzed. In response to which, the malware might abort the execution. In order to compare with on device execution, we measure activation as how successful our emulation is compared to on-device execution. Our activation rate is at 90% that is in par with the previous studies [16].

7 Related Work

There are several categories of related work to our research. Below, we discuss each category and explain how this research is different. Overall, none of the related efforts has focused on a binary centric approach that provides a holistic view on the network behaviors of the IoT malware

a. IoT malware system-level behavior analysis. Studying the behavior of IoT malware has become a hot topic both for academia and industry [13, 14]. These work focus more on the system level behavior of the malware, namely: the types of techniques that malware employs to evade detection, the type of device that the host is, and how malware makes itself persistent. Understanding the system level behavior of the malware is complementary, but significantly different from the network behavior which is our focus here.

b. IoT malware network behavior analysis. Some efforts focus on characterizing the behavior of a single malware family like Mirai [7, 19] or Hajime [21] while a few others characterize the behaviors of several malware families at the same time [16, 33] or the waiting phase behavior [20].

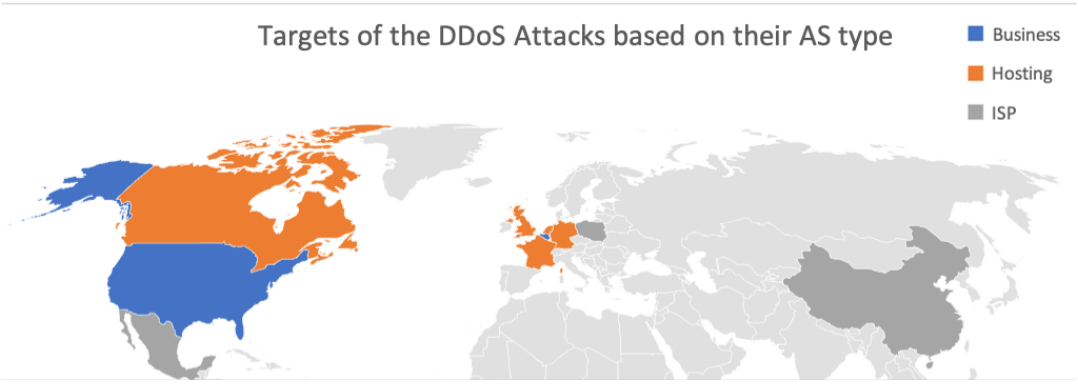

Figure 12. Targets of DDoS Attacks Based on the location and the Autonomous System type. A country is colored according to nature of the majority of its targets.

Although the previous efforts explore the network behavior of the IoT malware to some extent, neither they are binary-centric nor they provide a holistic view on all three types of IoT malware traffic as we do here. Other studies [19] [33] [16][20] do not study the proliferation or the attack phase of a botnet. Another study [7] provides an analysis of the DDoS attacks, but they rely on ISP traffic, which is an interesting and challenging problem in its own right: given network traffic, one has to identify and characterize the attack traffic. By contrast, we follow a binary-centric approach, which allows attribution of the attacks to the C2 servers and the malware binaries that initiate and carry out the attack.

c. IoT malware proliferation behavior analysis: A few related work explore the proliferation behaviors of the IoT malware [6, 23, 27]. While we share many similarities with these work, we focus on the recent malware, and hence recent trends in terms of exploits and vulnerabilities. In addition, we provide a holistic view including the C2 behaviors and the DDoS attacks.

d. C2 Communication Analysis: Several studies analyze C2 server communication from a networking point of view [17, 18, 29, 30, 36, 39]. Although interesting and informative, these studies focus on understanding the infrastructure that supports the botnet operation. By contrast, we provide a binary-centric measurement study with complementary profiling of proliferation and attack activity. In addition, we are the first to illustrate IoT C2 servers are elusive using active probing techniques.

e. Studying DDoS Attacks: Several studies focus on different aspects of DDoS attacks [24, 25, 32]. An earlier work [32] studies the victims of the DDoS attacks, and another work [25] studies the servers who issue the attack commands. Both studies use honeypots. On the other hand, a recent work [24] studies the effectiveness of booters takedown operations by analyzing network data from ISPs. Our fundamental difference is in our binary-centric approach. By only using malware binaries and spying into IoT malware botnets, we provide a similar analysis on the victims and origins of the DDoS attacks and get similar results as the previous work.

8 Conclusion

Our study can be seen as a proof of concept of different type of malware analysis that focuses on: (a) dynamic analysis, (b) timely collected malware, and (c) comprehensive profiling of all major bot activities. We collect daily and analyze on the same day the newly reported IoT malware from VirusTotal and MalwareBazaar. A binary-centric study can create a holistic picture of the IoT by connecting a binary and its family, with live C2 servers, a set of proliferation techniques, and even actual launched DDoS attacks.

First, we quantify the elusive behavior of C2 servers: 91% of the time a C2 server does not respond to a second probe sent four hours after a successful probe. We also find that 15% of the live servers that we find are not known by threat intelligence feeds available on VirusTotal. Second, we find that the IoT malware relies on fairly old vulnerabilities in its proliferation. Our binaries attempt to exploit 12 different vulnerabilities with 9 of them more than 4 years old, while the most recent one was 5 months old. Third, we observe the launch of 42 DDoS attacks that span 8 types of attacks while we observe that a target is often hit by two different attacks.

The overarching goal is to show the potential of a binary-centric dynamic analysis of malware with a focus on newly discovered binaries. Our preliminary results show the information and insights that can be obtained. Our future goal is to expand the scope of the study into a large-scale continuous IoT malware monitoring infrastructure.

References

[1] [n. d.]. VirusTotal. https://www.virustotal.com.

[2] Hungenberg,Thomas and Eckert, Matthias. 2022. Internet Services Simulation Suite. https://www.inetsim.org.

[3] abuse.ch. [n. d.]. About MalwareBazaar. https://bazaar.abuse.ch/about/.

[4] Abuse.ch. [n. d.]. MalwareBazaar. https://bazaar.abuse.ch/.

[5] Forbes Advisor. [n. d.]. Best Dedicated Hosting Services Of 2022. https://www.forbes.com/advisor/business/software/best-dedicated-server-hosting.

[6] Arwa Abdulkarim Al Alsadi, Kaichi Sameshima, Jakob Bleier, Katsunari Yoshioka, Martina Lindorfer, Michel van Eeten, and Carlos H Gañán. 2022. No Spring Chicken: Quantifying the Lifespan of Exploits in IoT Malware Using Static and Dynamic Analysis. (2022).

[7] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J Alex Halderman, Luca Invernizzi, Michalis Kallitsis, et al. 2017. Understanding the Mirai Botnet. In *Proceedings of the USENIX Security Symposium*.

[8] Big Data Cloud API. [n. d.]. Autonomous Systems (AS) advertised IPv4 space rank. https://www.bigdatacloud.com/insights/as-rank.

[9] Fabrice Bellard. 2005. QEMU, A Fast and Portable Dynamic Translator. In *Proceedings of the USENIX Annual Technical Conference (ATC, FREENIX Track)*.

[10] Xander Bouwman, Harm Griffioen, Jelle Egbers, Christian Doerr, Bram Klievink, and Michel van Eeten. 2020. A different cup of TI? The added value of commercial threat intelligence. In *Proceedings of the USENIX Security Symposium*. https://www.usenix.org/conference/usenixsecurity20/presentation/bouwman

[11] Daming D Chen, Maverick Woo, David Brumley, and Manuel Egele. 2016. Towards Automated Dynamic Analysis for Linux-based Embedded Firmware.. In *NDSS*.

[12] Andrei Costin, Jonas Zaddach, Aurélien Francillon, Davide Balzarotti, and Sophia Antipolis. 2014. A Large-Scale Analysis of the Security of Embedded Firmwares.. In *USENIX Security Symposium*. 95–110.

[13] Emanuele Cozzi, Mariano Graziano, Yanick Fratantonio, and Davide Balzarotti. 2018. Understanding Linux Malware. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*.

[14] Emanuele Cozzi, Pierre-Antoine Vervier, Matteo Dell'Amico, Yun Shen, Leyla Bilge, and Davide Balzarotti. 2020. The Tangled Genealogy of IoT Malware. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*.

[15] Crowdstrike. [n. d.]. Linux-Targeted Malware Increases by 35% in 2021: XorDDoS, Mirai and Mozi Most Prevalent. https://www.crowdstrike.com/blog/linux-targeted-malware-increased-by-35-percent-in-2021/.

[16] Ahmad Darki and Michalis Faloutsos. 2020. RIoTMAN: a systematic analysis of IoT malware behavior. In *Proceedings of International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*.

[17] Ali Davanian, Ahmad Darki, and Michalis Faloutsos. 2021. CnCHunter: An MITM-Approach to Identify Live CnC Servers. *Black Hat USA* (2021).

[18] Jonathan Fuller, Ranjita Pai Kasturi, Amit Sikder, Haichuan Xu, Berat Arik, Vivek Verma, Ehsan Asdar, and Brendan Saltaformaggio. 2021. C3PO: Large-Scale Study Of Covert Monitoring of C&C Servers via Over-Permissioned Protocol Infiltration. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*.

[19] Harm Griffioen and Christian Doerr. 2020. Examining Mirai's Battle over the Internet of Things. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*.

[20] Huy Hang, X. Wei, M. Faloutsos, and Tina Eliassi-Rad. 2013. Entelecheia: Detecting P2P Botnets in their Waiting Stage.. In *IFIP Networking*.

[21] Stephen Herwig, Katura Harvey, George Hughey, Richard Roberts, and Dave Levin. 2019. Measurement and analysis of Hajime, a peer-to-peer IoT botnet. In *Proceedings of the Network and Distributed Systems Security Symposium (NDSS)*.

[22] Heqing Huang, Cong Zheng, Junyuan Zeng, Wu Zhou, Sencun Zhu, Peng Liu, Suresh Chari, and Ce Zhang. 2016. Android malware development on public malware scanning platforms: A large-scale data-driven study. In *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 1090–1099.

[23] Seiya Kato, Rui Tanabe, Katsunari Yoshioka, and Tsutomu Matsumoto. 2021. Adaptive Observation of Emerging Cyber Attacks targeting Various IoT Devices. In *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 143–151.

[24] Daniel Kopp, Matthias Wichtlhuber, Ingmar Poese, Jair Santanna, Oliver Hohlfeld, and Christoph Dietzel. 2019. DDoS hide & seek: on the effectiveness of a booter services takedown. In *Proceedings of the Internet Measurement Conference*. 65–72.

[25] Johannes Krupp, Mohammad Karami, Christian Rossow, Damon McCoy, and Michael Backes. 2017. Linking amplification DDoS attacks to booter services. In *International Symposium on Research in Attacks, Intrusions, and Defenses*. Springer, 427–449.

[26] Victor Le Pochat, Sourena Maroofi, Tom Van Goethem, Davy Preuveneers, Andrzej Duda, Wouter Joosen, Maciej Korczyński, et al. 2020. A practical approach for taking down avalanche botnets under real-world constraints. In *Proceedings of the 27th Annual Network and Distributed System Security Symposium*. Internet Society.

[27] Tongbo Luo, Zhaoyan Xu, Xing Jin, Yanhui Jia, and Xin Ouyang. 2017. Iotcandyjar: Towards an intelligent-interaction honeypot for iot devices. *Black Hat USA* (2017).

[28] Yacin Nadji, Manos Antonakakis, Roberto Perdisci, David Dagon, and Wenke Lee. 2013. Beheading hydras: performing effective botnet takedowns. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. 121–132.

[29] Yacin Nadji, Manos Antonakakis, Roberto Perdisci, and Wenke Lee. 2011. Understanding the Prevalence and Use of Alternative Plans in Malware with Network Games. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*.

[30] Matthias Neugschwandtner, Paolo Milani Comparetti, and Christian Platzer. 2011. Detecting Malware's Failover C&C Strategies with Squeeze. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*.

[31] NFOservers. [n. d.]. NFOservers. https://www.nfoservers.com.

[32] Arman Noroozian, Maciej Korczyński, Carlos Hernandez Gañan, Daisuke Makita, Katsunari Yoshioka, and Michel van Eeten. 2016. Who gets the boot? analyzing victimization by ddos-as-a-service. In *International Symposium on Research in Attacks, Intrusions, and Defenses*. Springer, 368–389.

[33] Kevin Valakuzhy Ryan Court Kevin Snow Fabian Monrose Manos Antonakakis Omar Alrawi, Charles Lever. 2021. The Circle Of Life: A Large-Scale Study of The IoT Malware Lifecycle. In *Procedings of the USENIX Security Symposium*.

[34] pyxyp inc. [n. d.]. VulDB vulnerability database. https://vuldb.com/?kb.about.

[35] Silvia Sebastián and Juan Caballero. 2020. Avclass2: Massive malware tag extraction from av labels. In *Procedings of the Annual Computer Security Applications Conference (ACSAC)*.

[36] Rui Tanabe, Tatsuya Tamai, Akira Fujita, Ryoichi Isawa, Katsunari Yoshioka, Tsutomu Matsumoto, Carlos Gañán, and Michel Van Eeten.

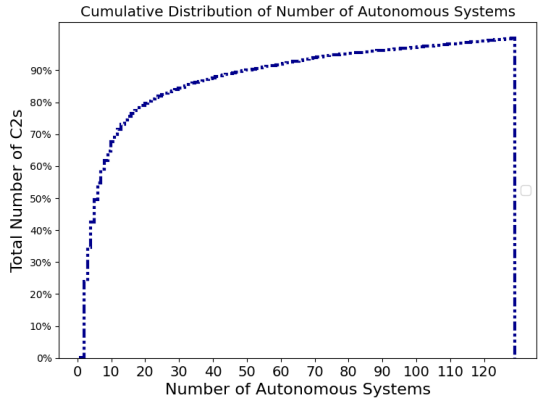

Figure 13. CDF of the number of AS that host a known C2 server.

2020. Disposable botnets: examining the anatomy of iot botnet infrastructure. In *Proceedings of the International Conference on Availability, Reliability and Security (ARES)*.

[37] Xabier Ugarte-Pedrero, Mariano Graziano, and Davide Balzarotti. 2019. A close look at a daily dataset of malware samples. *ACM Transactions on Privacy and Security (TOPS)* 22, 1 (2019), 1–30.

[38] Valve Developer Community. 2022. Forum. https://developer.valvesoftware.com/wiki/Main_Page.

[39] Pierre-Antoine Vervier and Yun Shen. 2018. Before toasters rise up: A view into the emerging iot threat landscape. In *Proceedings of the International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*.

[40] VirusTotal. [n. d.]. VirusTotal Contributors. https://support.virustotal.com/hc/en-us/articles/115002146809-Contributors.

[41] Shuofei Zhu, Jianjun Shi, Limin Yang, Boqin Qin, Ziyi Zhang, Linhai Song, and Gang Wang. 2020. Measuring and Modeling the Label Dynamics of Online Anti-Malware Engines. In *Proceedings of the USENIX Security Symposium*.

[42] Albert Zsigovits. 2021. Mirai/Gafgyt Fork with New DDoS Modules Discovered. https://cujo.com/mirai-gafgyt-with-new-ddos-modules-discovered/.

A Autonomous systems with C2 activity

In this section, we provide more details about the autonomous systems of C2 servers that we observed in our dataset. In total, 128 autonomous systems appeared in our dataset. Among these, ASN 15169 (Google LLC), ASN 16509 (Amazon.com Inc) and ASN 37963 (Hangzhou Alibaba Advertising Co.Ltd) are among the top 100 largest autonomous systems [8] (at the time of writing this paper). A CDF of number of autonomous systems and the distribution of C2s is depicted in Figure 13.

B Ports selected for probing

We list the ports selected for probing to compile *D-PC2* dataset in Table 5.

Ports	1312, 666, 1791, 9506, 606, 6738, 5555, 1014, 3074, 6969, 42516, 81

Table 5. The port configuration parameter of the probing to compile the *D-PC2* dataset.

Malware family	Description
Mirai	It exploits IoT devices and turns them into bots. It first appeared in 2016, and has been associated with many notorious DDOS attacks including the one targeting Dyn DNS service provider and OVH hosting service provider. Mirai communication protocol is binary based.
Gafgyt	It is a malware which infects Linux systems (especially those running BusyBox) in order to launch DDOS attacks. It first appeated in 2014, and since then many other variants have emerged. The main difference (for this study) that Gafgyt has with Mirai is its communication protocol that is text based.
Tsunami	It is a Linux backdoor that allows access to the infected machine/device. It also has capabilities to download and execute files from the Internet. For this study, its main distinction is its communication over the IRC protocol.
Daddyl33t	It is a modified version of another malware named QBot. QBot is a banking trojan that has keylogger functionalities. Daddyl33t, on the other hand, targets IoT devices. For the purpose of this study, we are interested in its distinct DDOS attacks that targets ICMP protocol and gaming servers.
Hajime	It is another IoT malware that originally was based on Peer 2 Peer (P2P) communications. According to [21], Hajime secures the infected device but at the same time tries to extend its reach by infecting more devices.
Mozi	Mozi is an evolution of Mirai and Gafgyt in many aspects, and shares similarities with Hajime in the P2P communication. According to CrowdStrike, it is one of the most prelavant linux malware, and it has already grown 10 times in the number of samples in 2021 [15].
VPNFilter	It is an Advanced Persistent Threat (APT) that targets router and network devices. VPNFilter has very sophisticated features compared to other IoT malware. For instance, it can persists itself on the IoT device and survive even after a reboot.

Table 6. A description of malware families analyzed in this study.

C Malware Families

In this section, we provide details of the malware families we found in our dataset. Table 6 contains this information.

vendor	Num of c2s	vendor	Num of c2s
0xSI_f33d	799	Kaspersky	798
PhishLabs	798	Netcraft	746
SafeToOpen	799	Forcepoint ThreatSeeker	745
AutoShun	799	CRDF	728
Lumu	799	Comodo Valkyrie Verdict	697
StopBadware	798	Fortinet	681
Cyan	799	Webroot	683
NotMining	798	Avira	568
CMC Threat Intelligence	578	Avira	568
CyRadar	387	G-Data	324

Table 7. This table reports the number of C2 addresses that different vendors could report malicious on a set of 1000 C2 IP addresses.

D Threat Intelligence Vendors

We use the feed provided by VirusTotal [1] to cross validate our detected C2s. The complete list of all the vendors that provide reputation information for IP and DNS names is available on the VirusTotal website [40]. At the time of conducting this study, 89 vendor feeds were available by VirusTotal (VT) API.

We provide a high level overview of these feeds here, and for a complete list you can refer to our dataset page[2]. From all the vendors that contribute to the VT threat intelligence, only 44 vendors could flag the C2 IPs in our dataset malicious at least for 1 C2 address (45 never flag any C2 address). Table 7 lists top 20 of vendors that could detect at least 20% of the C2 addresses.

E Weeks of Study

A mapping between the 31 weeks of our study in Figure 1 and their actual dates is presented below. Weeks 1 to 20 of the study happened in 2021 while weeks 21 to 31 were in 2022. Week 1 maps to the week 14 of the year, weeks 2 to 11 map to weeks 24 to 33, weeks 12 to 20 map to weeks 44 to 53 of year 2021. Weeks 21 to 31 map to weeks 2 to 12 of year 2022.

[2]https://github.com/adava/CnCHunter/wiki/MalNet-Datasets