MoZombie: A Case Study of the Self-Sustaining Mozi Botnet Architecture

Murtuza Mohammed, Georgios Smaragdakis, and Harm Griffioen

Delft University of Technology, The Netherlands
{m.a.mohammed, g.smaragdakis, h.j.griffioen}@tudelft.nl

Abstract. Mozi is one of the most prominent examples of an IoT peer-to-peer (P2P) botnet, having infected hundreds of thousands of devices. In late 2023, an update was pushed that aimed to dismantle infected devices, leading to what was widely described as a complete takedown. Yet, more than two years later, Mozi remains unexpectedly active: devices continue to communicate over Mozi's P2P protocol, propagate malware, and participate in scanning activity despite the absence of its maintainers. This persistence poses fundamental questions about the long-term behavior of decentralized botnets, the resilience of abandoned malware ecosystems, and the security posture of the global IoT landscape. In this paper, we present the first longitudinal analysis of Mozi's post-takedown ecosystem. We quantify the current size and distribution of active Mozi nodes, compare the visibility each method provides, and identify shared characteristics across our data sources. We further analyze samples still being spread by active nodes to determine whether the malware continues to evolve. Finally, we investigate the current structure of Mozi's P2P topology, geographic distribution, and the vulnerabilities still exploited for propagation.

1 Introduction

Peer-to-peer (P2P) botnets have long been regarded as among the most resilient forms of malicious infrastructure [10], as their decentralized design eliminates the single points of failure typical of conventional Command-and-Control (C&C) architectures [47]. One prominent example was Mozi, an Internet-of-Things (IoT) focused P2P botnet that rapidly grew after its appearance in 2019 by exploiting weak credentials and long-standing vulnerabilities in internet-connected devices such as routers, DVRs, and embedded systems [4]. At its peak, Mozi accounted for a substantial fraction of global IoT malware activity [46].

In late 2023, Mozi's operational infrastructure was forcibly dismantled by a "kill switch binary" [15], removing the original infection and installing itself. While it is unknown who was behind the takedown, the takedown should be slowly removing the infections across the Internet. Surprisingly, however, Mozi did not disappear. Nearly two years after the takedown, devices still exchange Mozi protocol messages, propagate the malware, and participate in scanning activity, despite the absence of its original maintainers [13].

This persistence raises fundamental questions about the long-term behavior of P2P botnets and the conditions under which decentralized malware ecosystems fail, or refuse to fail. Understanding why Mozi remains active requires examining the residual population of infected devices, the vulnerabilities that still enable its propagation, and the extent to which abandoned or "headless" P2P networks can maintain functional without active management. Moreover, Mozi's survival highlights structural challenges in IoT security: the prevalence of unpatched devices, the existence of vendor-abandoned firmware, and the capacity of P2P architectures to operate indefinitely in the absence of centralized coordination.

In this paper, we conduct the first longitudinal analysis of Mozi's post-takedown ecosystem. Using a large measurement infrastructure, active probing, and multi-month observations, we identify the geographic and network distribution of surviving Mozi nodes, and characterize the exploit vectors still used for propagation. By analyzing how Mozi's P2P topology behaves under decay, we seek to understand why the botnet has not died out, what external factors continue to sustain it, and what this implies for the long-term resilience of P2P malware. Our contributions are as follows:

- We quantify the current state of the persistent Mozi infection population, identifying active nodes and their geographic and network distribution. We show that a significant fraction of the botnet remains active nearly two years after the takedown.
- We analyze the exploit characteristics used by active Mozi nodes to propagate the malware, identifying which vulnerabilities are still being targeted. We find that Mozi continues to exploit a consistent set of old vulnerabilities, and is still able to maintain its population.
- We identify 'supernodes' within the Mozi P2P network that are highly resillient and serve as a 'backbone' for the botnet's connectivity, facilitating communication among infected devices.

2 Background

2.1 Botnets

Since the advent of the internet, groups have aimed to take control of internet exposed devices which are then used for a variety of illicit activities such as facilitating proxies to conceal the origin of other criminal activites, mining cryptocurrencies, participating in click fraud, email or ad spam campaings, and also participating in Distributed Denial of Service (DDoS) attacks. Earlier versions of botnets spread through spam emails [41], and moved on to credential stuffing attacks on unsecured SSH and Telnet devices [26, 27, 50]

The explosion in the field of IoT devices brought with it a large amount of internet connected devices, which are often designed to be low-cost and power efficient, leading to sacrifices made in securing these devices. This materializes in the form of devices with poor authentication methods such as default passwords,

or through poorly written code for internet facing services that can then be exploited to gain access to the device. Many of the large botnets have propagated through these infection vectors. Common examples include the Mirai botnet [11] which at its peak commanded around 200-300k devices and carried out a DDoS attack with a peak volume of 1.1 Tbps [3].

These botnets have been extensively studied and their disruption remains a critical problem to be solved. Solutions aimed at targeting these botnets, usually involve looking at `choke-points`, or highly critical portions of the botnets supporting infrastructure as viable targets for takedowns or blocklisting. In an aim to combat this, botnet creators come up with a variety of solutions such as Domain Generation Algorithms (DGA), fast-flux networks [28], bullet-proof hosters [25], and repeated reinfections [42]. These mitigate the disruption methods but do not eliminate the underlying weakness of a centralized botnet signalling and propagation infrastructure. Another method employed is to completely decentralise these components in order to make disruption extremely resource intensive and thus impractical.

2.2 P2P Botnets

Botnets often rely on a decentralized method to transfer Command-and-Control (C&C) instructions and also the infection and recruitment methods. There are several examples of succesful botnets that employ a decentralized infrastructure. An earlier example of these types of botnets is the Storm worm [39], which propagated mainly through social engineering with malicious websites and spam emails, starting around 2006. They also maintain C&C over a decentralised platform using the OVERNET protocol, which is a Distributed Hash Table scheme based on Kademlia. Based on the work enumerating these devices by Holz et al. [23], the Storm Worm had anywhere from 6,000 to 80,000 concurrent bots. Another example is that of the P2P version of the Zeus botnet [10] which originated in 2011 and achieved a peak of around 200,000 bots. The Zeus botnet is also based on a P2P network to exchange binaries and C&C commands. In order to account for older nodes that may leave the network and new nodes that join the network, the botnet nodes also hold their own routing tables of which they share a portion with each other. Interestingly these bots have a DGA as a backup in case a node is unable to bootstrap a routing list of other nodes, and also use proxy bots to help maintain the network.

2.3 Mozi

The Mozi botnet first emerged in September 2019, when a sample containing portions of Gafgyt [2], another prevalent botnet was analysed by 360NETLAB [4]. It spread using a list of 15 Remote Code Exploits (RCEs) and also through SSH and Telnet weak password bruteforcing. Since its creation, a study of the Mozi botnet had observed up to 1.5 million infected devices cumulatively [7]. The factor that made Mozi unique compared to other botnets was a combination of its usage of IoT devices and the fact that each infected device hosted the malware

binary itself as well, eliminating the need for centralised hosters. Moreover, the communication between nodes is facilitated by a decentralized network built on the Bittorrent protocol. Due to its decentralised nature and rapid scanning, the Mozi botnet gained a lot of attention as it produced a large amount of logs accounting for up to 90% of all traffic originating from IoT devices [6]. This eventually led to the takedown of the Mozi botnet, either by the authors or the Chinese authorities [15]. The Mozi samples were replaced by another sample that killed the running malicious sample and ceased all network activity, aimed ostensibly at killing the botnet.

3 Related Work

There are several works that focus on IoT botnets as a whole, Antonakakis et al. [11] did a longitudinal study of the Mirai botnet, combining data from various sources to perform a thorough analysis of the botnets spread and lifetimes. Griffioen et al. [19] analysed the landscape of Mirai variants and the competition between them over a limited set of devices. Vervier et al. [45] used low and high interaction honeypots to document the landscape of IoT devices over a 6 month period. Affinito et al. [8] looked at Mirai variants and their evolution over a period of 6 years.

P2P botnets have been subject to academic study since a long time, Grizzard et al. [20] performed one of the first analyses on a P2P botnet, the Peacomm bot or Storm Worm, although older bots such as Phatbot and Nugache were also studied. Holz et al. [23] performed a similar analysis of the botnet. They analysed the methods it used to spread, the communication protocols and provided methods to disrupt the botnet, which also proved to be effective against future strains of decentralised botnets. Andriesse et al. [10] performed an analysis of Gameover Zeus, a large P2P botnet. They provided an in-depth analysis of the underlying P2P network and comment on the highly persistent and disruption resistant nature of this botnet, owing to its sophistication. There are also works on how to enumerate P2P botnets [9] and enumeration and attack resilient protocols for P2P networks [9, 24, 48, 49]

Rossow et al. [38] provided a theoretical basis for disruption tactics and outline methods of which they then implemented a subset to evaluated their effectiveness on different botnets. This study targeted botnets that were active more than 15 years ago and although this work provided a solid foundation for these disruption methds, this was a work that needs to be refreshed and re-evaluated to account for developments in infrastructure and the rise of IoT botnets.

Bock et al. [13] compared two recent popular P2P IoT botnets: Hajime and Mozi (which is also the focus of this paper), using well established methods to provide a clear and complete picture of the state of these botnets over the course of their 8 month experiment. Similarly, Herwig et al. [21] performed a detailed analysis of the Hajime botnet by using a scraper and an embedded sensor as mentioned in the two previous works. They also downloaded files from these bots. They provided a detailed commentary on the spread and behavior of these

botnets but did not comment on possible disruption methods. Tu et al. [44] performed a similar study detailing in depth charateristics of the Mozi botnet but as an older work which needs to be refreshed, especially to understand the post takedown behavior and persistence of this botnet.

Hiesgen et al. [22] and Pauley et al. [36] created a general purpose medium interaction honeypot by responding to all incoming SYN requests with an ACK, thereby completing the first two steps of a handshake. For most botnets this was enough proof that a vulnerable service exists and they proceeded with the exploit attempts, allowing the honeypot to capture all infection attempts that rely on RCE or Telnet password stuffing attacks. This provides a much more easily scalable platform to collect these exploit attempts, their payloads and the malicious binaries themselves. We also use similar infrastructure to capture infection attempts from a myriad infected devices also including those which are infected by Mozi.

Earlier works compared the coverage of methods such as scraping the decentralized nodes or joining the nodes as a listener to enumerate the network, but our work is the first to also compare how well passive and active monitoring of the bots' exploitation activities compares to these measures. Moreover, to the best of our knowledge, we are the only work to investigate the persistence of the Mozi botnet post takedown.

4 Data Collection and Methodology

In order to gain a better understanding of the Mozi infection, we collect four complementary datasets that allow us to observe different aspects of the botnet's operation. Table 1 summarizes the datasets we use in this study, including their purpose, coverage, collection period, and data volume. Below, we describe the methodology used to generate each dataset in more detail.

Dataset	**Purpose**	**Coverage**	**Period**	**Volume**
Reactive Telescope	- Infection attempts - Source IPs	- Net #1: 256 - Net #2: ≈2k	10 months	2.17M packets
Passive Telescope	Passive monitoring	65k addresses	10 months	115M packets
Mozi Scraper	Node Routing Table	1 host	≈4 months	81.42M logs
Malware Downloader	- Malware samples - Download activity - Server uptime	- Net #1: 256 - Net #2: 2,048	10 months	2,926 samples

Table 1: Description of datasets used in this study. k=thousand, M=million.

4.1 Reactive Telescope

Mozi bots infect vulnerable devices by scanning the internet for open services and exploiting known vulnerabilities, ranging from bruteforcing credentials to

remote code execution (RCE) attempts. To capture this initial scanning activity and subsequent infection attempts, we deploy a reactive telescope setup similar to Spoki [22] and DScope [36]. In this setup, a set of IP addresses are monitored for incoming connection attempts and are instructed to complete the TCP handshake, allowing us to observe the full connection and any subsequent payloads sent by the scanning bots.

We emulate, like Spoki [22], DScope [36], and Ferrero et al. [18], an unresponsive protocol by only responding at the transport layer. For this study, we are interested in the application-layer information sent by the Mozi nodes. As shown in previous works, many exploitation scripts including Mozi do not verify that a service is running before sending an exploit, but instead send an exploit as soon as an open port is identified [37]. For services running on top of a plain-text protocol such as HTTP, we are then able to interpret the commands sent by the adversaries and identify their techniques.

As Mozi bots send an exploit payload immediately after the TCP connection is established, we are able to identify the devices infected and trying to spread the Mozi malware. We are able to attribute these infection attempts to Mozi by (1) the filename, (2) P2P download methodology, and (3) the specific malware binary that is downloaded. We deploy the system on a /25 network located in an enterprise environment and on a /21 network. The networks are located in the same geographical region, but span two distinct network locations and Autonomous Systems. Our deployment was inactive on two occasions during the collection period due to operational policies of the network provider.

An example of a Mozi infection attempt captured by our reactive telescope is shown in Figure 1. The figure illustrates the TCP handshake followed by the exploit payload sent by the Mozi bot, which includes the necessary commands to download and execute the Mozi malware on the target device.

4.2 P2P Network Monitoring

In order to sustain a P2P network, all the bots that participate in the network need to have a list of a subset of the network of other devices that they can share information and interact with. Mozi nodes contain an internal routing list that is populated with 128 other Mozi nodes that they have discovered through exchanges or that have contacted them. To study the dynamics of this peer list and understand the structure of the network, we employ scraping.

Scraping. Mozi nodes implement a custom version of the `find_nodes` and `get_peers` commands to exchange node lists [44]. These commands are similar to DHT implementations and require no authentication. We query the nodes to understand the state of their routing table at a given time. To bootstrap our scanning procedure, we perform a recursive search similar to the Mozi infection process: we query a root BitTorrent node for an infohash beginning with the string 888888, which matches the Mozi node ID prefix. This utilizes the Kademlia XOR distance metric to return a list of nodes sharing the Mozi prefix.

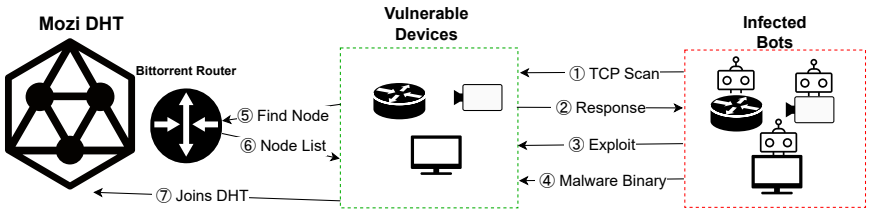

Fig. 1: Mozi infection and enrollment into DHT.

Returned nodes are stored in a Clickhouse table containing the NodeID, NodeIP, NodePort, FirstSeen, LastUpdated, and ConfigReceived fields. If an ID is observed for the first time, we add it as a fresh node; otherwise, we update its LastUpdated timestamp. We run this sequence every 5 minutes, checking nodes updated in the previous 24 hours to avoid unnecessary pings to disinfected or churned IPs. We use the ConfigReceived field as a secondary marker to ensure nodes are indeed Mozi nodes. In order to ensure that we do not poison the routing table of the nodes by attempting to enumerate them ourselves, we limit our scraper implementation to run from only one IP address taking a performance penalty in enumerating the ecosytem before it updates to ensure that we do not influence it ourselves.

4.3 Malware Samples

Every Mozi node is capable of hosting and serving the Mozi malware binary to other nodes that request it. We download the malware binary from infected nodes identified through our reactive telescope and peer-list datasets to identify potential variations and study its characteristics. We collect 413 Mozi samples out of a total of 2,926 samples over a period of 7 months.[1] In our reactive telescope, we extract all information relating to the exploit attempt, including the download path of the malware binary. Most collected Mozi samples are packed using UPX. These binaries are often difficult to unpack because malware authors alter or zero out portions of the binary used by the UPX unpacker. To obtain the unpacked sample, we utilize QEMU and GDB to run the sample in an isolated environment and track syscalls indicative of the UPX unpacking process. We determine from manually inspecting samples that the UPX unpacker calls the munmap function several times during the unpacking process, at the third call all sections of memory that are of interest to us are unpacked and loaded. We write a GDB script to count the number of munmap syscalls. Once the syscall signifying complete unpacking is captured, we pause execution and dump the process memory and dump all the sections that are marked as executable together to retrieve the total unpacked binary. This can now also be run and

[1] Hashes of these samples can be found at `https://osf.io/ezmsh/overview?view_only=d8fbc72909e24729a530cc2247f7efc7`

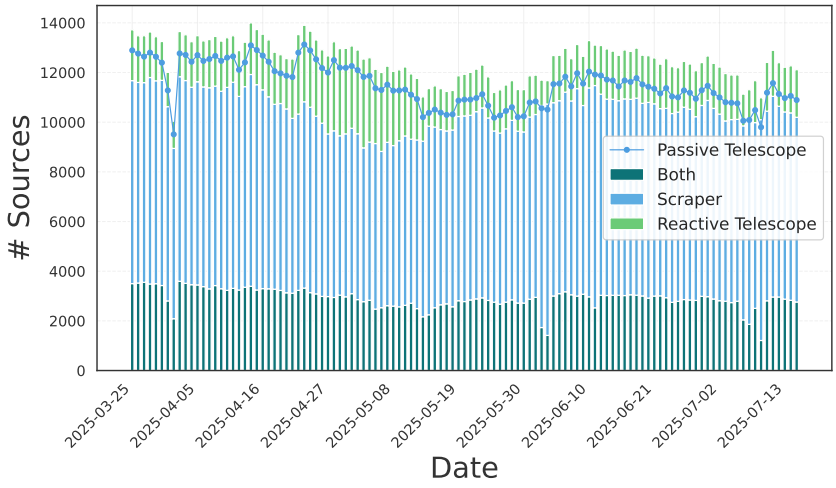

Fig. 2: Daily counts of Mozi nodes that we enumerate using our Reactive Telescope and DHT scraper, and observed addresses in our passive telescope.

behaves as the packed binary which we verified by running a subset of samples in a sandboxed environment. Through this analysis, we find five distinct variants of the Mozi malware in circulation.

5 Is Mozi Dead?

Mozi was taken down in late 2023 by a kill-switch binary that removed the original infection and installed itself [15]. However, nearly two years after the takedown, we observe that active Mozi infections continue to persist worldwide. In this section, we present our findings on the current state of the Mozi botnet, analyzing the active hosts, their behavior, and the characteristics of the malware samples being propagated. The goal of this section is to understand why Mozi remains active, and whether it can be considered truly dead.

5.1 Active Hosts

During our 10-month measurement period, we continuously monitor the Mozi botnet using our reactive telescope and DHT scraper. Figure 2 shows the number of devices identified per day, split over our datasets. We identify, on average, 12,501 hosts that are infected with Mozi daily. These values are in line with numbers provided by Bock et al. [13] and Tu et al. [44], which show averages around 17k and 10-20k in the years 2022 and 2019, respectively. This indicates that Mozi continues to maintain a significant presence in the wild, despite the takedown of its original infrastructure. We can therefore conclude that Mozi is not 'dead', and the P2P infrastructure enables it to persist. In the remainder of this paper, we will analyze the characteristics of these active hosts to understand

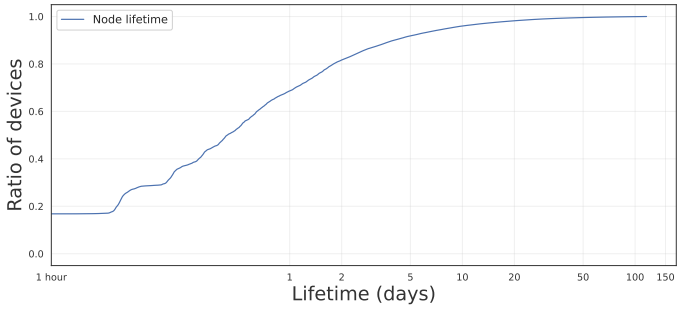

Fig. 3: CDF of lifetime of mozi nodes as seen in the reactive telescope (x-axis in log scale).

why they remain infected and how the botnet continues to propagate, and identify whether the botnet is still in use by its operators to shed light on whether we could consider it a real 'zombie' network and effectively 'dead'.

5.2 All hosts in the network are actively scanning

Figure 2 shows the overlap between the hosts we observe in our reactive telescope and those we capture by scraping the DHT. Although there is a significant overlap, the total addresses we see from just the reactive telescope is much less than those we see from just the DHT scraper. We hypothesize that this discrepancy arises from a lack of scanning activity from some infected devices, possibly due to network configurations such as firewalls or carrier-grade NATs (CGNATs) that prevent outward scanning. Additionally, some devices may be part of the P2P network but are not actively attempting to infect new hosts during our observation period. To identify potential scanning activity from the nodes we scrape, we cross-reference the IP addresses from both datasets with a larger passive telescope dataset consisting of approximately 80,000 IP addresses spread out over three enterprise networks. We find that out of the 172,706 hosts that do not appear in our reactive telescope, only 6,065 sources are not observed sending any packets to our passive telescope, indicating that a significant portion of the scraped nodes do exhibit scanning behavior, albeit not captured in our reactive telescope due to the limited size. The remainder of the hosts that are not identified in the passive telescope might be scanning at a rate lower than our observation window. This is strengthened by the fact that we also see a small portion of hosts in the reactive telescope that do not appear in the passive telescope. The fact that we see this level of active scanning from all but 3.51% of hosts distributed mainly in China and India also strengthens our belief that these are indeed infections and not research nodes for the vast majority of traffic that we see.

5.3 Node Lifetime

We analyze the lifetime of Mozi nodes observed in our reactive telescope to understand the persistence of infections. To do this, we do not look at the IP

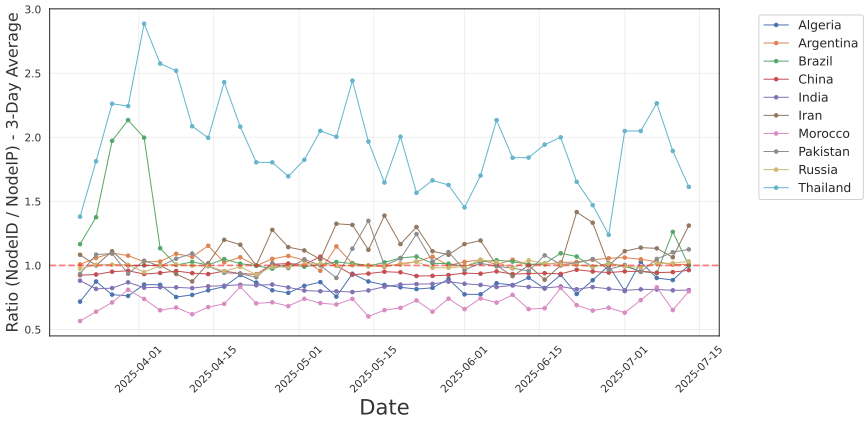

Fig. 4: Ratio of NodeIDs per NodeIP for top 10 countries.

address of the hosts, as this might churn, but instead use the unique *NodeID* specified at the start of an infection [13]. Figure 3 shows the cumulative distribution function (CDF) of node lifetimes based on their NodeIDs. We find that the mean and median lifetimes of Mozi nodes are 48.82 hours and 10.44 hours respectively. This indicates that while many nodes have short lifetimes, there is a long tail of nodes that remain active for extended periods, with some devices persisting for the entire duration of our study. The hosts that are alive for longer periods could be more resilient devices that are less likely to be disinfected or taken offline, contributing to the sustained presence of Mozi in the wild. To understand the nature of these persistent infections, we look up the top 1% (i.e., ≈3k) of devices based on lifetime in Censys [14] to identify their device types and characteristics. We find that the majority of these devices are running KRPC [1] as seen in Table 2. This is in line with the DHT activity associated with the Mozi nodes.

Protocol	Count	Protocol	Count	Protocol	Count
KRPC	141	MIKROTIK_BW	9	HTTPS	2
HTTP	71	NETBIOS	7	MQTT	2
UNKNOWN	57	SSH	5	TFTP	1
FTP	30	HIKVISION	4	SNMP	1
L2TP	13	CWMP	3	NTP	1
SIP	11	DNS	3	COAP	1
IKE	10	TELNET	2	SMTP	1

Table 2: Protocol counts for top devices in mozi.

5.4 SuperNodes

One of the reasons that a network like Mozi can remain persistent is due to the presence of long lived nodes that maintain connectivity in the distributed network, even if several nodes go offline or are disinfected. We indeed observe this behavior in our measurements, where several nodes remain active for extended

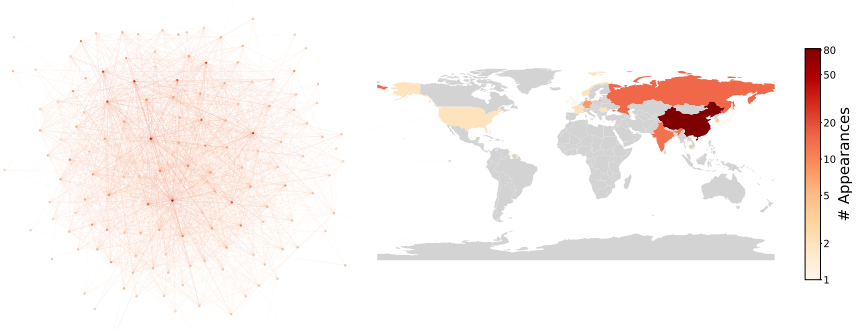

Fig. 5: Connection between supernodes that make the backbone of the Mozi DHT colored on a gradient by in-degree on the left and a Heatmap of the world showing the geographical concentration of supernodes on the right.

periods of time, even throughout the entire measurement period. Due to the way Mozi implements its routing table maintenance, nodes that respond to pings remain in the routing table of other nodes, which can lead to the creation of a backbone of long lived nodes (*supernodes*) that facilitate connectivity in the network as they are never evicted from the routing tables of other nodes. As we retrieve the routing tables of nodes during our scraping procedure, we can analyze the references between nodes to identify these supernodes.

When considering nodes that are referenced by other nodes more than 5,000 times throughout our measurement period, we identify a total of 710 nodes. 83 of these nodes remain active throughout the entire measurement period, making them highly persistent. The mean lifetime of these nodes, based on references in routing tables, is 13 days, which is significantly higher than the median lifetime of 10 hours observed for the average Mozi node. Furthermore, we find that a total of 339,604 (96.87%) nodes have referenced these 710 supernodes at least once, indicating their central role in maintaining the connectivity of the Mozi P2P network. For the 83 nodes that are active throughout the entire measurement, this effect is even more dramatic, as these nodes (accounting for 0.02% of the complete network) are referenced by 71.04% of other nodes throughout our measurements. Figure 5 visualizes the interconnectedness of these supernodes that were active for at least 100 days out of the total duration of our observation period amongst each other, colored on a gradient signifying the in-degree of these nodes from all other nodes in our dataset. As would be expected, these nodes are highly interconnected, as they are propagated through the network often (they are in many routing tables), and do not get evicted due to their long lifetimes. While we are not able to identify the type of devices that these supernodes are running on, they are important for the persistence of the Mozi network, providing stability to the infrastructure. Removal of these nodes, either through takedown or disinfection, could significantly impact the connectivity of the Mozi P2P network.

Vulnerability/Device	Count
CVE-2022-30023 [35]	217,791
CVE-2018-10561 [32]	107,702
Netgear DGN devices DGN1000 [43]	106,803
EDB-ID-40740 [16]	106,778
CVE-2014-8361 [29]	105,806
CVE-2016-20016 [34]	64,039
DVR generic vuln [17]	64,017
CVE-2019-8312/3/4/5/6/7/8/9/CVE-2019-7297 [33]	63,971
Vacro [40]	63,590
CVE-2016-6277 [30]	63,533
CVE-2017-17215 [31]	61,794

Table 3: Vulnerability Distribution Data

6 Mozi Node Characteristics

To understand the nature of the active Mozi infections, we analyze the characteristics of the exploit attempts observed in our reactive telescope. We classify the exploits based on the vulnerabilities they target and the types of devices they aim to infect. Table 3 summarizes the distribution of vulnerabilities exploited by Mozi during our measurement period.

First, we identify the newest vulnerability being exploited by Mozi, which is CVE-2022-30023 [35]. This indicates that the malware indeed has not been updated for a while, as no newer vulnerabilities are being targeted. Mozi relies on a set of old vulnerabilities, with the majority of exploit attempts target well-known vulnerabilities in IoT devices, such as CVE-2018-10561 [32] and CVE-2016-20016 [34], which have been widely exploited in the past.

While we would expect the distribution of infected devices to drop over time as new devices are immune and older ones are patched or go offline, we observe a relatively stable distribution throughout our measurement period as shown in Figure 2. This suggests that the pool of vulnerable devices remains consistent, either due to a lack of patching or the continuous addition of new vulnerable devices to the network.

6.1 Geographical and Network Distribution

Overall, the count of unique sources on a daily basis remains relatively stable. To further understand the distribution of the infected devices and whether the devices itself remain stable, we classify them based on their country of origin. Figure 6 shows the counts of sources that send us an exploit for the top 10 countries. While the overall counts remain stable for most countries, we do observe some fluctuations. First, we see a drop in devices from Brazil on the 19th of March, which could be due to network-level interventions, or due to CGNAT configurations that lead us to undercount the actual number of infected devices. We will identify the possibility of CGNATs in Section 6.2. The most notable change however is an increase in the number of infected devices from Pakistan starting around the 3rd of September, which could be due to the addition of new vulnerable devices in that region. As Mozi is only targeting older vulnerabilities, this could be due to the addition of older or end-of-life devices to the

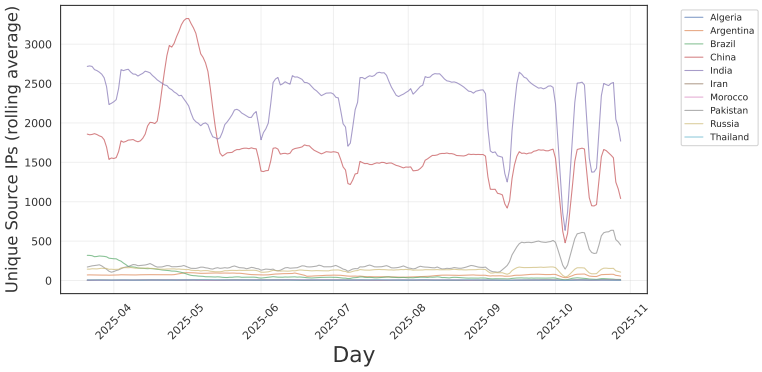

Fig. 6: Counts of sources that send us an exploit per Country

network that are subsequently infected. We analyze this further in Section 6.3. The only other country where we identify a notable increase in Mozi activity is the Phillipines (not shown in the graph), where the average number of active hosts increases from 10 to 70 (600%) throughout the measurement period.

6.2 Accuracy of Reported Node Counts

As shown in related work [12], network artifacts such as churn or CGNATs can lead to an overestimation or underestimation of the actual number of infected devices in a botnet. In order to understand the nature of the devices that remain infected with Mozi, we investigate the relationship between NodeIDs and NodeIPs observed in our scraping dataset. We use this to exclude the possibility of device churn or CGNATs leading to conclusions that are not representative of the actual number of infected devices.

We aim to identify the amount of devices that are present behind a CGNAT, which lead to a deflation in our counts. In order to do so, we look at IPs that have several NodeIds connected to them at a single point in time. This could be due to the device recovering and getting reinfected, thereby gaining a new NodeId, or due to the presence of several infected devices behind a NAT. We mainly use this to identify whether the trends that we observe in the previous sections would occur due to the presence of CGNATs (a drop in IP counts might be due to an operator installing CGNATs), more specifically the drop of infections in Brazil. When we look at the average number of NodeIds per NodeIP over time as seen in Figure 4, we see that this value remains relatively stable over time for all countries, except Brazil where we see a spike in the average NodeID/NodeIP value at the start of our measurement.

We also consider the possibility of device churn leading to an overestimation of the number of infected devices. To analyze this, we look at the number of unique NodeIPs that are associated with a single NodeID over time. A high number of NodeIPs per NodeID could indicate that devices are frequently changing their IP addresses, possibly due to DHCP leases or network reconfigurations.

After peak		Before peak	
ASN Name	**Value**	**ASN Name**	**Value**
National Telecommunication Corporation HQ	1951	National Telecommunication Corporation HQ	2050
Cyber Internet Services Pvt Ltd.	1014	Pakistan Telecommunication Company Limited	505
Pakistan Telecommunication Company Limited	346	House # 39 Street 38 F10 4	95
National WiMAXIMS environment	209	Cyber Internet Services Pvt Ltd.	4
PLAY BROADBAND PRIVATE LIMITED	9	IN CABLE INTERNET PRIVATE LIMITED	2
WellNetworks Private Limited	9	INSTACOM Pvt. LTD	2

Table 4: Comparison of unique sources per ASN before and after the peak.

However, as seen in Figure 4, we find that the majority of NodeIDs are associated with a single NodeIP, with only a small fraction showing multiple IPs. Specifically we look at the increase in the number of hosts in Pakistan as shown in Section 6.3, we see that the average number of NodeIPs per NodeID remains stable around 1-1.3 throughout the measurement period, indicating that device churn is not a significant factor in the observed increase.

6.3 Increase in Mozi Infections in Pakistan

While the population of infected devices remains relatively stable over time, we observe a significant increase in the number of Mozi infections originating from Pakistan starting around the September 3, 2025. As the Mozi malware does not appear to be actively evolving, this increase is unlikely to be due to new vulnerabilities being exploited. Instead, we hypothesize that it could be due to the addition of older or end-of-life devices to the network that are subsequently infected. To test this hypothesis, we (1) identify the specific Autonomous Systems (ASNs) contributing to this increase to understand whether it is indeed local in a few networks, and (2) analyze the vulnerabilities being exploited by these devices before and after the increase to identify which vulnerability might be driving the growth. We already ruled out device churn as a significant factor in this increase in Section 6.2. The overall steady rate of Mozi IDs per IP address lying between 1.0 to 1.4 gives credence to our initial hypothesis as this increase in number of infections cannot be due to IP address reprovisioning or removal of a set of devices from a CGNAT, as this would have reflected in the ratios.

Table 4 shows the distribution of unique sources per ASN before and after the increase in infections. We see that the top ASN contributing to the increase is the `Cyber Internet Services Pvt Ltd.`, a network service provider, which sees a jump from 4 unique sources before the increase to 1,014 after the increase. Other ASNs remain relatively stable in their counts. In both these networks with significant increase, we find that the exploit for Netgear DGN devices DGN1000 dominates after the increase of the malware. Additionally, we find a sharp increase in malware variant '1' being spread from the devices in these networks.

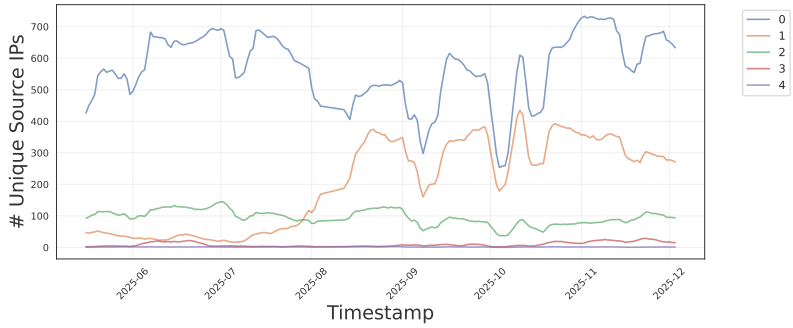

Fig. 7: Count of source IP addresses spreading the five unique malware samples over time.

6.4 Malware Loaded to the Devices

To understand whether the Mozi malware is still evolving, we analyze the samples being propagated by the active nodes in the network. We unpack and perform static analysis on the samples to identify unique strains and their characteristics. After unpacking the samples as described in Section 4.3, we extract strings from the unpacked samples using the linux 'strings' utility. We cluster samples based on this set of strings and identify 5 unique sets from a total of 347 samples that we succesfully unpacked. Among these sets, we see differences in login strings for bruteforce, some random characters that might be artifacts of code and also default linux directories, User-Agent strings, and format strings for handling internet addresses, possibly for self-hosting malware.

Figure 7 shows the distribution of the five unique samples over time. We observe that all five samples are still actively being propagated by infected nodes, remaining relatively stable throughout our measurement period. The only deviation we see is that Sample '1' sees an increase in propagation starting around August 2025. We are unable to correlate this increase to any specific event or change in the network, suggesting that it may be due to random fluctuations in the propagation behavior of infected nodes. Over the measurement period we do not observe any new samples being introduced to the network, indicating that the Mozi malware is not actively evolving anymore.

Mozi configurations. We also analyze the configurations being propagated by the active Mozi nodes. In the Mozi botnet, commands sent by the botmaster are propagated through the network using configuration files. These configurations contain instructions for the bots, such as targets for scanning, payloads to download, and other operational parameters. We identify these configurations being passed around the network to identify whether someone is still actively maintaining the botnet. We identify five configurations being propagated by the active nodes, with no new configurations being introduced over the measurement period.

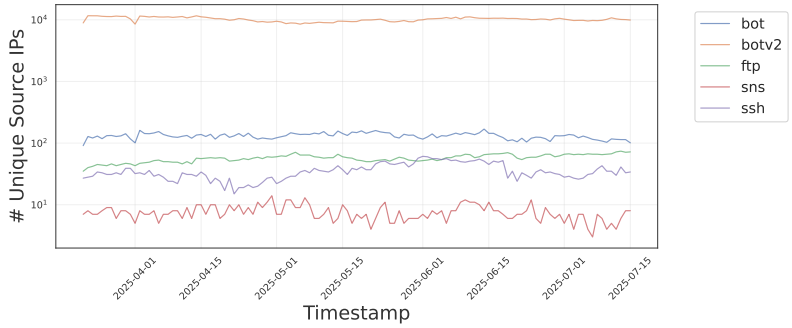

Fig. 8: Number of sources spreading a Mozi configuration daily over the measurement period.

Figure 8 shows the number of appearances of each configuration over time, the entire config is shown in Appendix A Table 5. We observe that all five configurations remain actively propagated by infected nodes, with no significant changes in their distribution. The two most popular configs are still instructing the bots to 'phone home' to *hxxp://ia[.]51[.]la*, which is still registered but appears to be offline. This is used by the botmasters to identify the infected devices and the status of the botnet [5]. The continued propagation of this configuration suggests that the takedown attempt has not fully neutralized the botnet, as infected devices are still following the original instructions set by the botmasters, meaning it could be re-activated if the authors choose to do so.

6.5 Comparison to an Active Botnet and Total Market Share

When looking at how much Mozi contributes to the overall traffic we see from RCE exploits towards our reactive telescope, we see that its distributed nature leads to it having a large share (80.8%) of sources that we receive packets from, despite having a lower share of the packet volume with only 2.96% of the total. We also compare Mozi with the scanning activities of an active botnet and observe that the active botnet has a much lower number of 7 malware download servers, and a total of 2866 unique source IPs that sent us an exploit attempt, compared to 165,989 malware download servers for Mozi. The botnet also scans and infects us less aggressively than Mozi, with a total volume of 166,432 packets compared to 694,536. The lifetime of these loaders is longer than what we see in Mozi with a mean and median of 29.71 and 31 days respectively compared to 8.68 and 11 (when removing 0 values for single observations) days. We include a graph comparing the daily counts of infectors Mozi and the active botnet and our methodology for collecting information on the active botnet in the Appendix A.2

7 Limitations

Our study has several limitations that should be considered when interpreting the results. First, our measurements are based on data collected from a reactive

telescope and DHT scraping, which may not capture the full extent of the Mozi botnet. Some infected devices may not exhibit scanning behavior or may not be reachable through the DHT, leading to potential underestimation of the botnet size. Additionally, our analysis relies on the accuracy of geolocation and ASN mapping services, which may introduce errors in the geographical and network distribution analysis. However, we have taken steps to validate our findings and ensure the robustness of our conclusions. Finally, our study focuses on a specific time period. However, the botnet interactions did not change over time, and we do not believe that this will change unless the botnet operators come back and 'revive' the botnet.

8 Discussion and Future Work

The Mozi botnet continues to persist in the wild, with a stable population of infected devices actively propagating the malware. Through our analysis, we identify some key characteristics of the botnet and the environment in which it operates. In this section, we discuss the implications of our findings and potential avenues for future research.

Exploit characteristics and device vulnerability. We show in Section 6 that Mozi continues to exploit a set of old vulnerabilities, with no new vulnerabilities being targeted. As the malware binaries are staged on the infected devices, instead of being downloaded from a Command-and-Control server, the botnet continues to function even after takedown attempts. We would expect however the population to decrease over time as devices are patched or go offline. However, we observe a stable population of infected devices, suggesting that the pool of vulnerable devices remains consistent. This could be due to a lack of patching in certain regions, or the continuous addition of new vulnerable devices to the network. The increase in infections in Pakistan, as hypothesised in Section 6.3, highlights how the addition of older or end-of-life devices to the network can lead to a resurgence in infections. It is thus not only a matter of patching existing devices, but also ensuring that new devices added to the network are secure and not vulnerable to known exploits.

Botnet resilience and supernodes. Our analysis of the Mozi botnet's network structure reveals the presence of supernodes that play a crucial role in maintaining the botnet's connectivity. These supernodes have a significantly higher degree of connectivity compared to regular nodes, allowing them to facilitate communication and coordination within the botnet. The removal of these supernodes could potentially disrupt the botnet's operations, making them prime targets for mitigation efforts. Future research could focus on developing strategies to identify and neutralize these supernodes, thereby weakening the overall botnet structure.

Mozi could be revived. The continued propagation of existing configurations and malware samples suggests that the Mozi botnet could be revived if the botmasters choose to do so. The infected devices are still following the original instructions set by the botmasters, indicating that they remain under the control of the botnet. As the number of infected devices is still significant, future research

could explore methods to effectively disinfect these devices or to disrupt the botnet's control mechanisms by poisoning the peer-list of the devices. If the peer-list can be poisoned effectively, the devices would be unable to communicate with each other, thereby neutralizing the botnet's operations. The persistence of Mozi, without any impulses from any botmasters, makes this a perfect testing ground to study the behavior of P2P botnet takedowns as there is limited outside interference.

9 Conclusion

In this study, we present a comprehensive analysis of the Mozi botnet, focusing on its persistence and characteristics. Our main objective is to identify what part of the botnet is still active, and whether the takedown of the botnet was successful. Through continuous monitoring using a reactive telescope and DHT scraping, we identify a stable population of infected devices actively propagating the Mozi malware. Our analysis reveals that the botnet continues to exploit a set of old vulnerabilities, with no new vulnerabilities being targeted. We observe a stable population of infected devices, suggesting that the pool of vulnerable devices remains consistent, possibly due to a lack of patching or the addition of new vulnerable devices to the network. The presence of supernodes within the botnet's network structure highlights potential targets for mitigation efforts. The network still poses a significant threat, as the infected devices remain under the control of the botnet and could be revived if the botmasters choose to do so. Our findings underscore the importance of continued monitoring and analysis of botnets, as well as efforts to disinfect infected devices and prevent reinfection.

10 Ethical Considerations

As our experiment usually involves interacting with infected consumer hardware, over the publically distributed Bittorrent P2P, there are a number of considerations we have to make to ensure that we do not cause or propagate any unintended harms. In this section we will detail the measures we take to ensure that we limit the side effects of our measurement study.

Firstly when bootstrapping and adding new devices, we query the root node a limited number of times for the identified characteristic hexadecimal prefix "88888888" that is used by Mozi to identify other nodes. Due to this verification, we automatically discard a significant subset of the Bittorrent space. The chances of a host randomly being assigned the Mozi prefix is $\frac{1}{16^8}$ which are astronomically low. Furthermore, when bootstrapping or identifying hosts that might not have the Mozi prefix we ping the addresses received from confirmed Mozi nodes without the prefix at large intervals (around 8 hours in practise) to gauge their response. The large interval makes the traffic we send to hosts negligible with a size of 146 bytes per packet, we have an estimated traffic sent of 48.66 Bps just over the short duration that we contact them. When we do not receive the expected response from these hosts after a set number of tries, we stop pinging the hosts permanently. Finally, we only ping hosts that have replied to us in the last day, ensuring that we do not direct traffic to now recovered or churned IPs.

Acknowledgments

This work was supported by the Dutch Research Council (NWO) under the ADAPTIve project and by the European Union under the Horizon Europe Programme as part of the project SafeHorizon (Grant Agreement #101168562).

References

1. DHT Protocol. https://bittorrent.org/beps/bep_0005.html (2008)
2. Linux Bashdoor GafGyt and Small ELF Backdoor at shellshock. https://blog.malwaremustdie.org/2014/09/linux-elf-bash-0day-fun-has-only-just.html (2014)
3. Heightened DDoS Threat Posed by Mirai and Other Botnets. https://www.cisa.gov/news-events/alerts/2016/10/14/heightened-ddos-threat-posed-mirai-and-other-botnets (2017)
4. Mozi, Another Botnet Using DHT. https://blog.netlab.360.com/mozi-another-botnet-using-dht/ (2019)
5. A new botnet attack just mozied into town. https://www.ibm.com/think/x-force/botnet-attack-mozi-mozied-into-town (2020)
6. Mozi Botnet Accounts for Majority of IoT Traffic. https://threatpost.com/mozi-botnet-majority-iot-traffic/159337/ (2020)
7. Deep tracking of Mozi botnet: 360 security brain accurate traceability, find out behind the scenes. https://mp.weixin.qq.com/s/Su0-uU5JaUrAh8ptTzTCsA (2022)
8. Affinito, A., Zinno, S., Stanco, G., Botta, A., Ventre, G.: The evolution of Mirai botnet scans over a six-year period. Journal of Information Security and Applications **79** (2023). https://doi.org/10.1016/j.jisa.2023.103629
9. Andriesse, D., Rossow, C., Bos, H.: Reliable recon in adversarial peer-to-peer botnets. In: Proceedings of the 2015 Internet Measurement Conference. p. 129–140. ACM, Tokyo Japan (Oct 2015). https://doi.org/10.1145/2815675.2815682, https://dl.acm.org/doi/10.1145/2815675.2815682
10. Andriesse, D., Rossow, C., Stone-Gross, B., Plohmann, D., Bos, H.: Highly resilient peer-to-peer botnets are here: An analysis of Gameover Zeus. In: 2013 8th International Conference on Malicious and Unwanted Software: "The Americas" (MALWARE). p. 116–123. IEEE, Fajardo, PR, USA (Oct 2013). https://doi.org/10.1109/MALWARE.2013.6703693, https://ieeexplore.ieee.org/document/6703693/
11. Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J.A., Invernizzi, L., Kallitsis, M., Kumar, D., Lever, C., Ma, Z., Mason, J., Menscher, D., Seaman, C., Sullivan, N., Thomas, K., Zhou, Y.: Understanding the Mirai Botnet. In: 26th USENIX Security Symposium (USENIX Security 17). pp. 1093–1110. USENIX Association, Vancouver, BC (Aug 2017), https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis
12. Böck, L., Levin, D., Padmanabhan, R., Doerr, C., Mühlhäuser, M.: How to Count Bots in Longitudinal Datasets of IP Addresses. In: Proceedings 2023 Network and Distributed System Security Symposium. Internet Society, San Diego, CA, USA (2023). https://doi.org/10.14722/ndss.2023.24002, https://www.ndss-symposium.org/wp-content/uploads/2023/02/ndss2023_f2_paper.pdf

13. Böck, L., Sundermann, V., Fusari, I., Karuppayah, S., Mühlhäuser, M., Levin, D.: The end of the canonical iot botnet: A measurement study of mirai's descendants (arXiv:2309.01130) (2023). https://doi.org/10.48550/arXiv.2309.01130, http://arxiv.org/abs/2309.01130, arXiv:2309.01130 [cs]
14. Durumeric, Z., Adrian, D., Mirian, A., Bailey, M., Halderman, J.A.: A Search Engine Backed by Internet-Wide Scanning. In: 22nd ACM Conference on Computer and Communications Security (Oct 2015)
15. ESET Research: Infamous IoT botnet Mozi taken down via a kill switch. https://www.eset.com/sg/about/newsroom/press-releases1/awards/eset-research-infamous-iot-botnet-mozi-taken-down-via-a-kill-switch/ (2023)
16. ExploitDB: Eir D1000 RCE. https://www.exploit-db.com/exploits/40740 (2016)
17. ExplotDB: Multiple CCTV-DVR Vendors - Remote Code Execution. https://www.exploit-db.com/exploits/39596 (2016)
18. Ferrero, D., Bassetti, E., Griffioen, H., Smaragdakis, G.: Have you SYN What I See? Analyzing TCP SYN Payloads in the Wild. In: Proceedings of the 2025 ACM Internet Measurement Conference. p. 928–936. IMC '25, Association for Computing Machinery, New York, NY, USA (2025). https://doi.org/10.1145/3730567.3764498, https://doi.org/10.1145/3730567.3764498
19. Griffioen, H., Doerr, C.: Examining Mirai's Battle over the Internet of Things. ACM, Virtual Event USA (2020). https://doi.org/10.1145/3372297.3417277, https://dl.acm.org/doi/10.1145/3372297.3417277
20. Grizzard, J.B., Sharma, V., Nunnery, C., Kang, B.B., Dagon, D.: Peer-to-Peer Botnets: Overview and case study. In: Proceedings of the First Conference on First Workshop on Hot Topics in Understanding Botnets. p. 1. HotBots'07, USENIX Association, USA (2007)
21. Herwig, S., Harvey, K., Hughey, G., Roberts, R., Levin, D.: Measurement and Analysis of Hajime, a Peer-to-peer IoT Botnet. In: Proceedings 2019 Network and Distributed System Security Symposium. Internet Society, San Diego, CA (2019). https://doi.org/10.14722/ndss.2019.23488
22. Hiesgen, R., Nawrocki, M., King, A., Dainotti, A., Schmidt, T.C., Wählisch, M.: Spoki: Unveiling a New Wave of Scanners through a Reactive Network Telescope. In: 31st USENIX Security Symposium (USENIX Security 22). pp. 431–448. USENIX Association, Boston, MA (Aug 2022), https://www.usenix.org/conference/usenixsecurity22/presentation/hiesgen
23. Holz, T., Steiner, M., Dahl, F., Biersack, E., Freiling, F.: Measurements and mitigation of peer-to-peer-based botnets: A case study on storm worm
24. Karuppayah, S., Böck, L., Grube, T., Manickam, S., Mühlhäuser, M., Fischer, M.: SensorBuster: On Identifying Sensor Nodes in P2P Botnets. In: Proceedings of the 12th International Conference on Availability, Reliability and Security. p. 1–6. ACM, Reggio Calabria Italy (Aug 2017). https://doi.org/10.1145/3098954.3098991, https://dl.acm.org/doi/10.1145/3098954.3098991
25. Konte, M., Perdisci, R., Feamster, N.: Aswatch: An as reputation system to expose bulletproof hosting ases. In: Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication. p. 625–638. SIGCOMM '15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2785956.2787494, https://doi.org/10.1145/2785956.2787494
26. Munteanu, C., Feldmann, A., Smaragdakis, G., Fiebig, T.: Catch-22: Uncovering Compromised Hosts using SSH Public Keys

27. Munteanu, C., Saidi, S.J., Gasser, O., Smaragdakis, G., Feldmann, A.: Fifteen Months in the Life of a Honeyfarm. In: Proceedings of the 2023 ACM on Internet Measurement Conference. p. 282–296. ACM, Montreal QC Canada (Oct 2023). https://doi.org/10.1145/3618257.3624826, https://dl.acm.org/doi/10.1145/3618257.3624826
28. Nazario, J., Holz, T.: As the Net Churns: Fast-Flux Botnet Observations. In: 2008 3rd International Conference on Malicious and Unwanted Software (MALWARE). pp. 24–31 (2008). https://doi.org/10.1109/MALWARE.2008.4690854
29. NVD: CVE-2014-8361. https://nvd.nist.gov/vuln/detail/cve-2014-8361 (2015)
30. NVD: CVE-2016-6277. https://nvd.nist.gov/vuln/detail/CVE-2016-6277 (2016)
31. NVD: CVE-2017-17215. https://nvd.nist.gov/vuln/detail/cve-2017-17215 (2018)
32. NVD: CVE-2018-10561. https://nvd.nist.gov/vuln/detail/cve-2018-10561 (2018)
33. NVD: CVE-2019-8318. https://nvd.nist.gov/vuln/detail/cve-2019-8318 (2019)
34. NVD: CVE-2016-20016. https://nvd.nist.gov/vuln/detail/cve-2016-20016 (2022)
35. NVD: CVE-2022-30023. https://nvd.nist.gov/vuln/detail/CVE-2022-30023 (2022)
36. Pauley, E., Barford, P., McDaniel, P.: DScope: A Cloud-Native Internet Telescope. In: 32nd USENIX Security Symposium (USENIX Security 23). pp. 5989–6006. USENIX Association, Anaheim, CA (Aug 2023), https://www.usenix.org/conference/usenixsecurity23/presentation/pauley
37. Pauley, E., Barford, P., McDaniel, P.: The CVE Wayback Machine: Measuring Coordinated Disclosure from Exploits against Two Years of Zero-Days. In: Proceedings of the 2023 ACM on Internet Measurement Conference. pp. 236–252 (2023)
38. Rossow, C., Andriesse, D., Werner, T., Stone-Gross, B., Plohmann, D., Dietrich, C.J., Bos, H.: SoK: P2PWNED - Modeling and Evaluating the Resilience of Peer-to-Peer Botnets. In: 2013 IEEE Symposium on Security and Privacy. p. 97–111. IEEE, Berkeley, CA, USA (May 2013). https://doi.org/10.1109/SP.2013.17, https://ieeexplore.ieee.org/document/6547104/
39. Smith, B.: A Storm (Worm) Is Brewing. Computer **41**(2), 20–22 (2008). https://doi.org/10.1109/MC.2008.38
40. SonicWall: Vacron Network Video Recorder Remote Command Execution. https://blog.sonicwall.com/en-us/2022/06/vacron-network-video-recorder-remote-command-execution/ (2022)
41. Stock, B., Göbel, J., Engelberth, M., Freiling, F.C., Holz, T.: Walowdac - Analysis of a Peer-to-Peer Botnet. In: 2009 European Conference on Computer Network Defense. pp. 13–20 (2009). https://doi.org/10.1109/EC2ND.2009.10
42. Tanabe, R., Tamai, T., Fujita, A., Fujita, R., Isawa, R., Yoshioka, K., Matsumoto, T., Ganan, C., Van.Eten, M.: Disposable Botnets: Examining the Anatomy of IoT botnet Infrastructure. Virtual Event Ireland (2020). https://doi.org/10.1145/3407023.3409177, https://dl.acm.org/doi/10.1145/3407023.3409177
43. Tenable: NETGEAR DGN Remote Unauthenticated Command Execution. https://www.tenable.com/plugins/nessus/104128 (2017)
44. Tu, T.F., Qin, J.W., Zhang, H., Chen, M., Xu, T., Huang, Y.: A comprehensive study of Mozi botnet. International Journal of Intelligent Systems **37**(10), 6877–

6908 (2022). https://doi.org/https://doi.org/10.1002/int.22866, https://onlinelibrary.wiley.com/doi/abs/10.1002/int.22866
45. Vervier, P.A., Shen, Y.: Before toasters rise up: A view into the emerging IoT threat landscape. In: Research in Attacks, Intrusions, and Defenses: 21st International Symposium, RAID 2018, Heraklion, Crete, Greece, September 10-12, 2018, Proceedings 21. pp. 556–576. Springer (2018)
46. Wang, B., Sang, Y., Zhang, Y., Li, S., Xu, X.: A Longitudinal Measurement and Analysis Study of Mozi, an Evolving P2P IoT Botnet. In: 2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). pp. 117–122 (2022). https://doi.org/10.1109/TrustCom56396.2022.00027
47. Wang, P., Wu, L., Aslam, B., Zou, C.C.: A Systematic Study on Peer-to-Peer Botnets. In: 2009 Proceedings of 18th International Conference on Computer Communications and Networks. pp. 1–8 (2009). https://doi.org/10.1109/ICCCN.2009.5235360
48. Yan, G., Chen, S., Eidenbenz, S.: RatBot: Anti-enumeration Peer-to-Peer Botnets, Lecture Notes in Computer Science, vol. 7001, p. 135–151. Springer Berlin Heidelberg, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24861-0_10, http://link.springer.com/10.1007/978-3-642-24861-0_10
49. Yan, G., Ha, D.T., Eidenbenz, S.: AntBot: Anti-pollution Peer-to-Peer Botnets. Computer Networks **55**(8), 1941–1956 (2011). https://doi.org/10.1016/j.comnet.2011.02.006
50. Zhu, Y., Chen, Z., Yan, Q., Wang, S., Giaretta, A., Li, E., Peng, L., Zhao, C., Conti, M.: Devils in the Clouds: An Evolutionary Study of Telnet Bot Loaders. In: ICC 2023 - IEEE International Conference on Communications. p. 2338–2344. IEEE, Rome, Italy (May 2023). https://doi.org/10.1109/ICC45041.2023.10278636, https://ieeexplore.ieee.org/document/10278636/

A Additional Information.

A.1 Full configuration strings as observed from Mozi nodes

Mapping of config names in Fig. 8 to entire config string can be seen in Table. 5.

Name	Configuration String
bot	[ss]bot[/ss][hp]88888888[/hp][count]http://ia.51.la/go1?id=19894027&pu=http%3a%2f%2fbaidu.com/[idp][/count]
botv2	[ss]botv2[/ss][dip]192.168.2.100:80[/dip][hp]88888888[/hp][count]http://ia.51.la/go1?id=17675125&pu=http%3a%2f%2fv.baidu.com/[idp][/count]
ftp	[ss]ftp[/ss][cpu].w[/cpu][hp]88888888[/hp]
sns	[ss]sns[/ss][cpu].m[/cpu][hp]888[/hp]
ssh	[ss]ssh[/ss][cpu].x[/cpu][hp]88888888[/hp]

Table 5: Mozi config types to complete contents.

A.2 Comparison with Active botnet

To compare Mozi with an active botnet, we identify a botnet by its unique characteristics, namely the exploit method which utilizes only two exploits, CVE-2021-41773 and CVE-2024-4577 and has the exploit activities as shown in Listing. 1. We ensure that we have the complete data from this botnet by checking all other activity from the infectors, loaders and including these file names and associated activity in our reactive telescope to ensure that we do not miss out on a portion of the activity.

```
POST /cgi-bin/../../../../../../../../../../bin/sh HTTP/1.1
Host: x.x.x.x:80
Accept: */*
Upgrade-Insecure-Requests: 1
User-Agent: Custom-AsyncHttpClient
Connection: keep-alive
Content-Type: text/plain
Content-Length: 107
X=$(curl http://x.x.x.x/sh || wget http://x.x.x.x/sh -O-);
    echo "$X" | sh -s apache.selfrep
```

Listing 1: Exploit attempts from our selected active botnet for comparison.

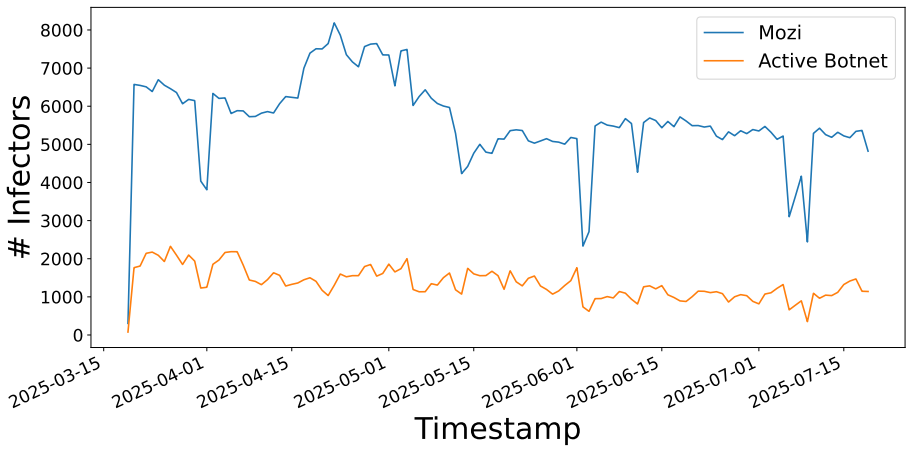

Fig. 9: Figure comparing daily counts of source IPs that attempt to infect our monitoring infrastructure over the duration of our study.