Attacks Come to Those Who Wait: Long-Term Observations in an SSH Honeynet

Cristian Munteanu
Max Planck Institute for Informatics
Saarbrucken, Germany
cmuntean@mpi-inf.mpg.de

Yogesh Bhargav Suriyanarayanan
Max Planck Institute for Informatics
Saarbrucken, Germany
ysuriyan@mpi-inf.mpg.de

Georgios Smaragdakis
Delft University of Technology
Max Planck Institute for Informatics
Delft, Netherlands
g.smaragdakis@tudelft.nl

Anja Feldmann
Max Planck Institute for Informatics
Saarbrucken, Germany
anja@mpi-inf.mpg.de

Tobias Fiebig
Max Planck Institute for Informatics
Saarbrucken, Germany
tfiebig@mpi-inf.mpg.de

ABSTRACT

Numerous studies have explored SSH attacks, often focusing on specific botnet activities or providing short-term analyses of particular honeynets. In this paper, we present an analysis of data collected from a large-scale honeynet over a three-year period, shedding light on gradual shifts in attacker behavior. Our findings suggest a trend toward more exploratory attacks, with indications that attackers are increasingly moving beyond the blind execution of scripts.

We observe changes in techniques as new bots appear with unique methods and established botnets modify their approaches over time. Furthermore, attackers have adopted a more scouting approach in recent months, showing increased adaptability in their tactics. Additionally, there is a clear preference for utilizing recently registered ASes as storage locations for malicious files. Our findings also suggest that attackers are increasingly aware of honeypot presence. Some attackers actively search for these traps, while others exploit honeypots for their own purposes, underscoring the need for a new generation of more advanced honeypots.

Lastly, we conduct a detailed investigation into one of the most prevalent attacks, challenging existing assumptions about the attacker's identity.

CCS CONCEPTS

• **Security and privacy → Network security**.

KEYWORDS

Honeypot, Intrusion Detection, Network Scanning, Internet Security, Threat Intelligence.

ACM Reference Format:
Cristian Munteanu, Yogesh Bhargav Suriyanarayanan, Georgios Smaragdakis, Anja Feldmann, and Tobias Fiebig. 2025. Attacks Come to Those Who Wait: Long-Term Observations in an SSH Honeynet. https://doi.org/10.1145/3730567.3764475

1 INTRODUCTION

The spread of malware and the rise in attacks over SSH have become significant concerns in recent years. SSH, a widely used protocol for secure communication and remote system management, has also become a frequent target for malicious actors [12, 40]. Recent trends indicate not only a growth in the volume of attacks but also an evolution in their methods. This is particularly evident in the context of global events, such as the Russia-Ukraine war, which has triggered a marked increase in hostile activity across the Internet [43].

Attackers employ a range of techniques, from worms and viruses to sophisticated botnets. While prior research has provided valuable insights into attacker behavior, these studies often focus on specific botnets [80] or rely on data collected during short timeframes from honeypots [17]. Such approaches, though informative, are limited in their ability to capture the evolving nature of malicious activity.

The goal of this study is to move beyond static snapshots of attacker behavior and investigate how it changes over extended periods. This is critical because defending against attacks is an ongoing struggle between those seeking to exploit systems and those working to protect them. As attackers refine their tactics and strategies, defenders must stay informed and adapt to mitigate evolving threats.

Monitoring attacker behavior over longer periods enables us to uncover broader patterns, detect recurring threats, identify waves of activity, and correlate these trends with external events. This knowledge is essential for designing effective and adaptive defense mechanisms. In this study, we leverage data collected through a collaboration with a large-scale honeynet operator to conduct a longitudinal analysis of SSH-based attacks. This approach provides valuable insights into how attacker behavior evolves over time, highlighting the necessity of continuous observation and adaptation in the development of defensive strategies.

Our contributions can be summarized as follows:

- We investigate 163 million intrusive sessions that execute commands over 3 years and classify these commands based on their purpose. We identify a clear shift in attacker behavior starting in 2023.
- We analyze commands attempting to execute a file and observe that attackers' preferences for loading files onto compromised targets have evolved over time.
- We examine more than 16 thousand malicious files (hashes) and apply a clustering algorithm based on executed commands to

classify various bots, including *Mirai*, *Gafgyt*, *XXorDDoS*, and others, to tracking their activity over time.

- We analyze the IPs used as malware storage locations and find that more than 70% are hosted in recently registered Autonomous Systems (ASes) no older than five years. Additionally, we observe that some attackers tend to reuse the same storage IPs over extended periods.
- We discover that attackers are actively scanning for honeypots and, in one case, abuse them as proxies for attacks.
- Finally, we investigate an aggressive attack campaign visible throughout the entire observation period and provide insights into its behavior and characteristics.

2 BACKGROUND AND RELATED WORK

SSH attacks: The Secure Shell Protocol (SSH) is a popular and extensively deployed protocol, making it a frequent target for attacks. Despite its improvements over the years, brute-force [47, 77, 95] and dictionary attacks [35, 96] remain some of the most common threats against SSH.

Although disabling password authentication can effectively mitigate these attacks, brute-force methods are still widely observed. This *persistence* can be attributed to several factors, ranging from inexperienced users to the proliferation of IoT devices. IoT devices, which commonly employ SSH for remote access, are especially vulnerable to brute-force attacks, as many are deployed with weak default passwords [54, 67]. Botnets like *Mirai* have exploited this vulnerability, using brute-force techniques to compromise large numbers of IoT devices and launch attacks [12, 20, 40, 83, 100].

Another type of SSH attack targets weaknesses within the SSH protocol itself. Recent work by Bäumer et al. [18] demonstrates that certain SSH server configurations can expose vulnerabilities, enabling a “prepending” attack. Their study reveals that, under specific conditions, attackers can exploit these vulnerabilities to successfully conduct Man-in-the-Middle (MitM) attacks, compromising the integrity and security of SSH communications.

Brute-force attacks: The problem of brute-force attacks against SSH services has been extensively studied in the literature, with multiple works analyzing attacker behavior through honeypot deployments. Two notable contributions in this domain are the works of Abdou et al. [9] and Singh et al. [77].

Abdou et al. [9] presented an in-depth analysis of automated SSH brute-force attacks. Their study revealed common patterns such as credential reuse, dictionary-based attack strategies, and the reliance on compromised hosts as intermediaries for further exploitation. A key strength of their work lies in its systematic examination of attacker infrastructure and the propagation of credentials across different targets. However, their dataset was constrained to a relatively short observation window, limiting the ability to capture long-term variations in attacker behavior.

Singh et al. [77] approached the problem from a complementary perspective, focusing on the detection and mitigation of SSH brute-force attacks in operational environments. Their research proposed practical defense mechanisms and highlighted the integration of honeypot data into intrusion detection systems. While their contribution provides actionable insights for improving security posture, the emphasis remained on immediate countermeasures rather than developing a longitudinal understanding of attacker dynamics.

In contrast, our work extends the state of the art by conducting a three-year longitudinal analysis of SSH honeypot data. This long-term perspective enables the identification of temporal trends that were not observable in prior short-term studies, such as shifts in attack intensity, changes in the distribution of adversaries, and the evolution of credential usage strategies. By bridging short-term empirical observations with long-term behavioral trends, our study provides a more comprehensive understanding of attacker behavior. This allows not only for validation and contextualization of prior findings but also for uncovering novel insights that were previously overlooked due to temporal limitations in existing research.

Honeypots: Honeypots are purpose-built to attract attackers by presenting themselves as vulnerable systems, making them powerful tools for studying attack methods [24, 65]. They are typically categorized by their interaction level, resulting in three main types: low, medium, and high-interaction honeypots [60]. Low-interaction honeypots [16, 72] simulate a limited range of system behaviors, while medium and high-interaction honeypots [29, 34, 49, 84] mimic a broader range of interactions, capturing more detailed attacker actions.

Honeypots can be deployed on virtual machines or physical hardware [38, 89, 94] and are available in various forms, including web honeypots [45, 99], mobile honeypots [97], and those focused on worm detection [31]. IoT honeypots have become particularly valuable in recent years, identifying malware families targeting IoT devices [54, 67] and providing insights for countering botnets [46, 51, 70]. In 2022, Hiesgen et al. [42] introduced “Spoki”, a hybrid between a traditional network telescope and a honeypot, to examine malicious activity, identifying attack variants such as *Mozi* and *Mirai*.

Wu et al. [102] deployed low-interaction honeypots on a /16 IPv4 prefix to study attacker behavior over three years, but their analysis was limited to connection-level attributes due to the lack of executed command capture. Similarly, Ghiëtte et al. [39] investigated attacks on 4,500 low-interaction honeypots over a one-month period. Other researchers have applied clustering [76] and natural language processing techniques [23] to group attackers’ IPs and analyze behaviors.

Regarding SSH honeypots, Kippo [49] and Cowrie [29] are among the most widely used today [65], and are the basis of most recent studies of SSH malicious activity [19, 50]. A recent study by Munteanu et al. [62] leveraged data from a large honeyfarm, i.e., a set of honeypots distributed around the globe with centralized data collection, to explore malware hash variations over fifteen months. Our work goes beyond hash variability and geographic profiling to provide an in-depth analysis of attacker behavior and characterization over time. Additionally, Izhikevich et al. [44] recently demonstrated that attackers are increasingly sophisticated in targeting cloud services. Our research complements this by examining attacker techniques against residential targets, as observed through a large Honeynet.

Nawrocki et al. [64] offer a different perspective on honeypots, also conducting an in-depth analysis of their limitations. Their findings highlight that one of the primary constraints lies in the placement and vantage points of honeypots. In our study, we extend this discussion by identifying additional limitations, such as the inability of a honeypot to capture files downloaded by attackers due to the methods used to transfer them. This gap in monitoring suggests that modern attacks have become sophisticated enough to evade certain honeypot configurations, highlighting the need for ongoing advancements in honeypot technology to keep up with evolving threat tactics.

Malicious Actor Characterization: As noted by Barron et al. [17], attribution and attacker profiling are among the most difficult aspects of defending against organized or state-sponsored threats. Studies also suggest that attackers' strategies, persistence, and the likelihood of specific attack types are influenced by target configurations, vulnerabilities, and perceived value [75, 88]. These studies emphasize the importance of differentiating between human-driven attacks and automated bot activity.

Honeypots play a crucial role in this characterization, as illustrated by Sadique et al. [74], who analyzed botnet behavior to understand attackers' actions post-compromise. While their focus was on predicting attacker behavior, our analysis investigates the possible motivations behind botnet tactics and strategy changes. Stone-Gross et al. [80] conduct an extensive and in-depth analysis of the Torpig botnet, providing valuable insights into its behavior and attack characteristics.

In addition, other studies have pursued predictive methods, such as statistical models to forecast attacks [104] or machine learning applied to honeypot data to detect severe SSH attacks [73]. Research has also shown that malware developers increasingly use sophisticated techniques like polymorphism to evade detection [53, 69, 78].

In this work, we correlate data collected from a large honeynet with multiple external sources to better understand and characterize the modus operandi of the most prominent SSH attacks observed in recent years.

3 DATASETS

In this section, we present the profile of the honeynet we collaborate with as well as other data sources we use.

3.1 Honeynet Profile

For our study we collaborate with a large non-profit honeynet operator that provide access to researchers. The rationale for deploying the honeynet is to operate honeypots geographically distributed in different countries as well as in different networks, with a focus on residential networks. The honeynet consists of 221 identically configured honeypots in 55 countries and 65 ASes. Each honeypot is realized using a customized version of the Cowrie Honeypot suite [29], a medium interaction SSH and Telnet honeypot designed to log brute force attacks as well as shell interactions executed by intruders. The selection of the Cowrie honeypot software is driven by its ease of use and because it covers multiple attack vectors over SSH and Telnet.

3.2 Collection of Honeynet Data

Each successful TCP connection handshake by a client on either the SSH port 22 or the Telnet port 23 creates a new session recorded by each honeypot. Once the sessions is closed, the recorded session is forwarded to a collector and is added to the honeynet database. A session is ended either by a TCP connection tear down from the client or a timeout by the honeypot, which is configured to be three minutes.

For each session, the honeypot records basic session information, which includes the start time, the end time (including timeout), the IP and port of the honeypot as well as the client. In addition, if SSH is used, it records the client SSH version. Moreover, the honeypot records the interactions of the client with the honeypot, namely, used credentials for login and executed commands. For each login attempt, it records the credentials used as strings and whether the used credentials are accepted. The honeypots are configured to allow password-based SSH authentication using the username root and by supplying any password except “root”. Public key based SSH authentication is not supported. For Telnet, the same authentication rules as for SSH are in place.

After a successful login, the client has access to a Unix-like shell that emulates common Unix commands. As such, the honeypot records each command executed by the client in a list of “known” or “unknown” commands. Known commands are emulated by the honeypot; unknown ones are simply recorded. If a command includes a URI (this includes anything retrieved from a remote target, including retrievals via (S)FTP, HTTP(S), etc.), the URI is recorded as well. If a command results in a creation or modification of a file, a hash of the file content is recorded. The data is processed and analyzed in situ, using the provided interface.

3.3 Honeynet Dataset Statistics

Our study spans over 33 months, i.e., December 2021 to August 2024, all 221 honeypots were active with one exception between 8 and 9 of October 2023. During this time the honeynet was on maintenance for 48 hours and did not record any sessions. Besides this incident our error reporting did not notice any outages. Overall, the honeynet recorded more than 635M sessions. In this study we focus only on the recorded SSH attacks, which accounts for more than 546M sessions comming from more than 850K unique client IPs.

We classify the attacks into the following categories:

Scanning: No credentials are used. The session records the TCP handshake.

Scouting: After the TCP handshake, the client tries to log in, but the login is unsuccessful.

Intrusion: After the TCP handshake, the connected client manages to successfully login, but executes no commands.

Command Execution: After the TCP handshake, the connected client manages to successfully login and also executes at least one command.

Of all sessions, 45M are related to scanning, while scouting sessions account for the largest share, totaling 258M. Intrusion sessions rank third with 80M, and command execution sessions are second, totaling 163M.

3.4 Abuse Datasets

The honeypots gather detailed information about attacks, including malicious files. However, only the hashes of these files are collected, and the files themselves are not stored.

To better understand the intent behind the attacks, it is necessary to correlate the file hashes with known attack types. A valuable source of information for such correlations are abuse databases, which compile and categorize malicious file hashes.

In our analysis, we utilize publicly available abuse databases to cross-reference and validate our findings. Among these, we rely on several prominent services that aggregate and maintain extensive records of malicious activities in the wild:

abuse.ch [10] is an open access platform that tracks and detects threats with a strong emphasis on malware software and botnets. Internet Service Providers (ISPs), network and cloud operators, security and hardware vendors, government agencies, and law enforcement agencies rely and integrate Abuse's feed in commercial or open source products to detect and fight attacks in their organizations. The platform also enables the sharing of threat intelligence data with the research and operational security community. We utilize the threat intelligence feed related to IP reputation and malware software samples.

Team Cymru [82] is a threat-intelligence company that provides security solutions, threat analytics, and risk assessment for enterprises and hosts on the Internet. By collecting and analyzing various network data it constructs block lists and reputation scores.

VirusTotal [93] is a threat-intelligence company that specializes on the collection and reverse engineering of malware binaries. It is one of the most popular references for malware detection and characterization. It utilizes reports by many other security companies and volunteers to tag suspicious files (searchable via their `sha256` hash or binary). We got access to VirusTotal for all the hashes in our study.

ArmstrongTechs Project [15] ArmstrongTechs is a Internet security firm specializing in various services to enhance organizational security. Additionally, ArmstrongTechs maintains a GitHub repository titled "Indicators-of-compromise-IOCs", which includes information on various threats and indicators of compromise.

In the following, we refer to all of these services together as abuse datasets.

3.5 Autonomous System Data

To better understand which infrastructure the attackers are using we categorize the investigated IPs according to the AS that announces it. For this purpose we use a service for looking up historic announcement information for IPv4 [81]. This service returns for each IP and timestamp a historic perspective which include the announcement time period, AS number, as well as AS organization details. To further categorize the returned ASes we use bgp.tools [36] as well as PeeringDB [7].

Both bgp.tools and PeeringDB employ a variety of tags and labels to describe and classify ASes comprehensively. Collectively, these platforms categorize ASes into 19 distinct tags. For our analysis, we focus specifically on differentiating between the following types:

CDNs - Content Delivery Networks.
Hosting - Hosting providers (including web-hosting, VPN).
ISPs/NSPs - Internet Service providers.
Others - Other types of networks (including governmental, academic, corporation, personal networks, or unlabeled).

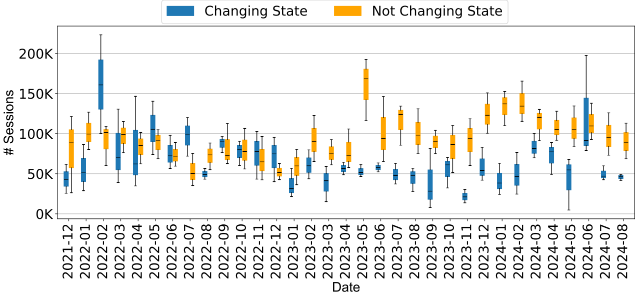


Figure 1: Sessions that contain commands split in two categories. In blue, we have the sessions that change the state of the honeypot and in orange otherwise.

We didn't consider labels such as "IPv6-only" or "Tranco 10k", as these are infrequent in our dataset and do not significantly contribute to understanding overarching patterns in attacker behavior.

4 ETHICS

For this study, honeypot logs are processed and analyzed at the collector server infrastructure of the honeynet.

Sensitive data regarding the honeypots hosting information (as well as our collaborator server infrastructure) is not mentioned in our study to reduce the risk of uncovering the identity and IP address of the honeynet infrastructure.

For the same reason, we do not discuss any sensitive information related to honeypot configuration to protect the operation of the honeypot.

In our paper, we also anonymize the IP addresses of infected devices and malware infrastructure. We do not "name and blame" cloud providers, content delivery networks, and network providers that may host malware as they may not be aware of this activity by their tenants. However, we collaborate with our honeynet provider to disclose this information to infrastructure providers.

5 COMMANDS

To initiate our analysis, we examine the types of commands observed in the sessions collected by the honeypots. Our focus is on sessions in which attackers successfully logged in and executed at least one command, totaling approximately 163 million sessions. We categorize these sessions into two distinct types based on the nature of the commands executed. The first category includes sessions where commands do not alter the state of the honeypot. These commands are generally non-intrusive, primarily aimed at gathering information about the system, such as identifying existing files, inspecting running processes, or assessing the machine's capabilities. We observe that nearly 94 million sessions fall within this type. The second category consists of sessions in which at least one command modifies the honeypot's state. Such commands may edit or delete files or execute actions that actively alter the system's original state. The remaining of 69 million sessions belong to this category.

Figure 1 presents the distribution of both types of sessions over time. Each boxplot in the figure represents the daily distribution of sessions of each type for the corresponding month. During the initial phase of our data collection, spanning from 2021 to early 2023, both session types occur at similar rates, with a notable spike observed in early 2022. This spike correlates with well-documented attacks linked to the onset of the Russia-Ukraine conflict [58] and – as we show in Figure 3(a) – is dominated by one botnet.

In early 2023, however, a shift becomes apparent, marked by an increase in sessions involving commands that do not alter the honeypot's state. This trend suggests a growing preference among attackers to explore the system rather than immediately execute malicious commands after gaining access. One possible explanation for this shift could be an evolution in attack strategies: attackers might prioritize reconnaissance to better understand a target system before executing commands, potentially as a tactic to evade detection by antivirus software, honeypots, or other security mechanisms.

Classification: To better understand the commands observed in the honeypot, we conducted an in-depth analysis and manually developed a set of regular expression (regex) rules [106] to classify attacker activity. This process resulted in 59 distinct categories of commands, designed to capture recurring patterns of malicious behavior. The regexes were iteratively constructed by first identifying common command structures and execution patterns (e.g., repeated use of `wget`, `curl`, or package installation commands), and then generalizing these observations into rules capable of grouping similar activity. Each regex therefore corresponds to a behavioral signature, rather than a single literal command, allowing us to classify even slightly varied instances of the same activity.

Out of the 59 categories, 58 were generated through explicit regex matching, while the 59th serves as a fallback for commands that could not be classified, denoted as the `unknown` category. Furthermore, 44 categories correspond to commands associated with a specific bot or attack wave, capturing the characteristic toolsets and behaviors of automated malware. The remaining 14 categories reflect more generic intrusion behaviors, such as methods of introducing malicious files (*wget*, *ftp*, *curl*, or *echo*), which are commonly reused across different bots.

Applying this classification framework to the dataset of 162 million sessions, over 161 million sessions (>99%) were successfully categorized into one of the 58 regex-based groups. The remaining 1 million sessions, which did not match any of the predefined patterns, were placed in the `unknown` category. This approach ensures that bots are not treated as individual sessions, but rather as recurring behavioral patterns spanning multiple sessions. Additional implementation details and the complete set of regex rules are provided in Appendix B.

State modification: First, we take a closer look at the sessions that do not alter the state of the honeypot. Figure 2 presents a detailed breakdown of the bot categories observed in these sessions. We observe that the majority of the sessions (over 95%) can be attributed to the top three bots, with the leading one, `echo_OK`, alone accounting for more than 80%. Another notable observation is the presence of both continuous/consistent activity patterns (e.g., `echo_OK`, `uname_svnrm`) and wave-like or campaign-based behavior (e.g., `bb_scout_cat`, `uname_a`).

We next examine sessions that modify the honeypot's state, focusing on whether they involve specific file executions or downloads. Our analysis reveals that over 54 million sessions involve commands that add, modify, or delete files without executing them. The remaining 15 million sessions contain commands attempting to execute a file on the system.

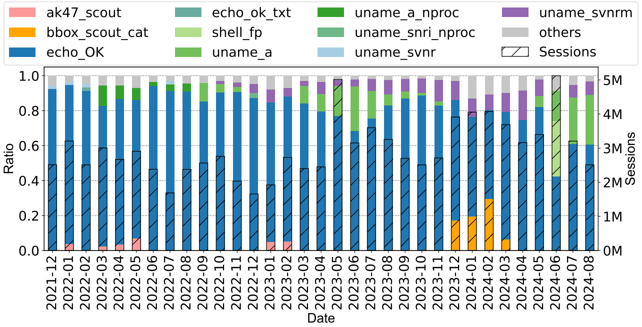


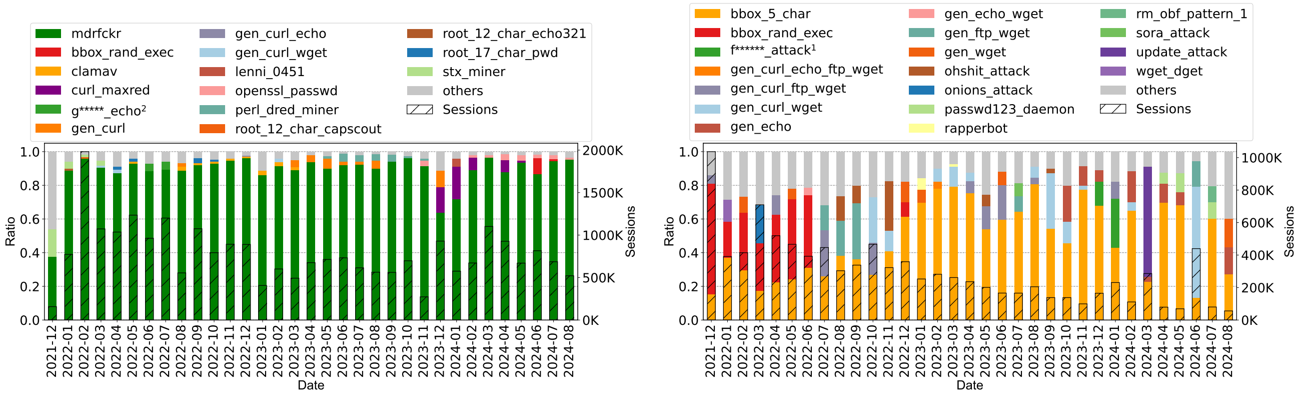
Figure 2: Sessions that do not change the state of the honeypot showing the Top 3 bots/month.

Figure 3 presents a temporal overview of the sessions that modify the state of the honeypot. On the left, Figure 3(a) provides a detailed analysis of the bot categories that add, modify, or delete files without executing them over time. As with the sessions that do not alter the honeypot's state, the majority of this activity (over 90%) is dominated by a single bot–`mdrfckr`–which will be discussed in detail in Section 9.

We observe a high number of sessions each month (>500k) in which attackers attempt to add, write, or delete files without executing them. This tactic may indicate that intruders are more focused on the following: (1) establish persistent access, e.g. by adding a public key, and as such "storing" the compromised device for later use; (2) placing a malicious script on the device for execution at a later time (e.g. coordinating with other devices for a DDoS attack); (3) inserting a script intended for silent execution upon a specific trigger, such as modifying a `.init` file or the crontab; (4) generating a random file and verifying its presence in a subsequent session to test system consistency, as inconsistency may signal to attackers that the device could be a honeypot. To date, no research has been published analyzing SSH attacks using stateful honeypots. However, there are existing proposals for the design of stateful honeypots, as discussed in [26].

Web attacks: At the start of 2024 we observe a wave of a different bot (`curl_maxred`). Closer analysis of the bot's sessions reveals that the honeypots repeatedly execute `curl` commands aimed at specific domains or IP addresses. This activity originates from four client IPs associated with a Russian hosting provider. These clients connect via SSH to 180 of the 221 honeypots in our honeynet, generating nearly 200k sessions in total. In each session, the attacker executes approximately 100 `curl` commands to various destinations. As a result, this group of four IP addresses is responsible for generating 20 million `curl` requests between January and April 2024. For more details see Appendix C.

The targeted domains reveal shared characteristics: most are Russian or Ukrainian sites associated with sectors such as economy, trade, cryptocurrency, e-commerce, Telegram bots, drugs, and gaming. Essentially, the attacker appears to be using the honeypot as a "proxy" with the intent to launch attacks against these sites. From our perspective, the nature of these attacks likely involves



(a) Sessions that add, modify, or delete files without executing them.

(b) Sessions that attempt to execute files.

Figure 3: Sessions that change the initial state of the honeypot.

either DDoS attempts or credential exploitation through the use of stolen cookies.

Both medium- and high-interaction SSH honeypots are intentionally designed to properly execute commands like *wget* or *curl* in order to retrieve and store malicious files for later analysis. Although honeypots are inherently non-malicious and serve solely to collect threat intelligence, here, their ability to execute such commands enable attackers to misuse them as intermediaries for malicious activity.

File exec: We continue our analysis by examining the sessions that attempt to execute files (see Figure 3(b)). First, we observe a higher diversity of bots, with the top three accounting for only about 50% of all sessions. Notably, the leading bots—`bb_5_diff_char_v2` and `bbox_unlabelled`— leverage the `/bin/busybox` [101] tool to execute potentially malicious scripts. This is not surprising; due to its lightweight implementation, *busybox* has been widely reported by multiple IoT honeypot studies as a preferred method for deploying and executing malware on IoT devices [28, 48, 98]. We also observe that the `bbox_unlabelled` campaign abruptly ends in mid-2022, which may indicate either a takedown or an intentional cessation of activity—especially since no other bot category appears to take its place thereafter.

Another important trend is visible in the number of sessions over time (secondary Y-axis), which shows a clear decline. Starting from late 2022, we observe a marked downward trend, with no new bots executing files. Approximately 60% of activity during this period can be attributed to the `bb_5_diff_char_v2` bot. This decline suggests that attackers may be shifting toward more stealthy attacks.

Honeypot files: We continue our investigation by delving into the sessions we consider most intrusive—those containing commands that attempt to execute a file. Figure 4 displays the distribution of sessions that run a command which tries to execute files over time. Sessions where we successfully identify the file's hash, are labeled as "file exists", while sessions labeled "file missing" indicate no associated hashes. In the later case, the filename being executed was not found in any previous honeypot sessions, nor was it downloaded through commands (on the shell offered by the honeypot) in the current or prior sessions. This implies that attackers may try to copy the file via `scp`, `ftp` or `rsync` on the honeypot from a remote location. As these methods are not emulated by the deployed Cowrie implementation, the honeypots cannot catch the file.

We find more than 3 million sessions where the "file exists" and around 12 million sessions with "file missing". Figure 4(a) presents the temporal distribution of the session where the executed file exists, while Figure 4(b) represents the temporal distribution of sessions where the executed file is missing.

While there is a noticeable downward trend in sessions attempting to execute files, we observe an even more significant drop in sessions where the file actually exists. In 2022, the honeypots recorded over 100k "file exists" sessions per month. However, starting in 2023, this number dropped sharply to approximately 5k sessions per month.

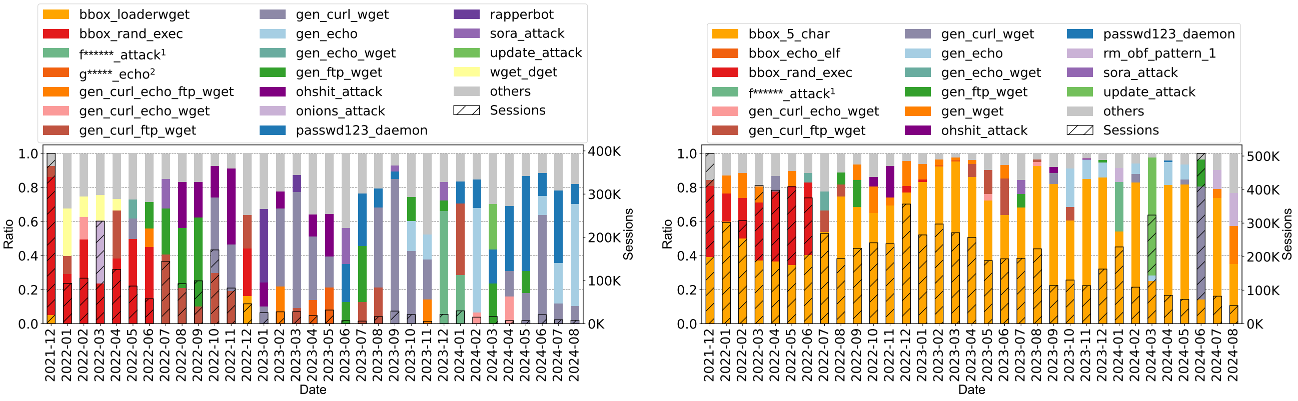
This shift could potentially be explained by two factors: first, bots may be misconfigured, resulting in incomplete or improperly executed attacks; second, malicious actors may have adapted their tactics. Specifically, the attackers either (a) recognized that honeypots are unable to capture files transmitted directly via protocols such as `scp`, `ftp`, or `rsync`, or (b) opted to simplify their operations by bypassing the need for a malware loader entirely, see Appendix D.

Another interesting observation concerns the bot category `bbox_unlabelled` (shown in red). This bot appears to have had multiple variants–some using protocols like *wget* and *tftp*, which allowed the honeypots to capture the dropped files, and others using protocols that prevented the honeypots from retrieving the files. It is plausible that this bot reached its end-of-life phase due to its extensive use of diverse protocols, which may have facilitated a deeper understanding by defenders and, ultimately, the development of effective countermeasures–especially in contrast to a similar bot, `bb_5_diff_char_v2`, which remains active.

This longitudinal analysis of command types over time reveals a clear change in the modus operandi (MO) of SSH brute-force attackers. We observe a clear preference for bots exhibiting scouting behavior. By categorizing the executed commands, we can effectively visualize that the number of bots displaying aggressive behavior decreases over time, in contrast to those that refrain from

[1] Slur against Jewish people redacted.

[2] Slur against homosexual men redacted.



(a) Sessions with exec commands, files exist

(b) Sessions with exec commands, files missing

Figure 4: Sessions that attempt to execute files.

executing files. Additionally, we identify a scenario in which the honeypots are vulnerable to misuse, serving as potential proxies for launching further attacks. These trends underscore the critical need for continual adaptation in anti-malware defenses, especially in honeypot technologies.

6 MALWARE FILES

To gain more insight into the attacker behavior, in this section, we focus specifically on sessions in which files are loaded onto the honeypot. In more than 3 million sessions, the intruder explicitly executes commands on the honeypot to download a file. The honeypots do not retain the original files after download; instead, they generate a SHA-256 hash based on each file's content. Over the duration of our study, we identified a total of 16,257 unique hashes. By cross-referencing these file hashes with abuse databases, we are able to classify them as follows:

Malicious: An exec file or a script that reads/writes protected OS files. Typically, a Virus or a Trojan.

Mirai: One of the first significant botnets targeting exposed networking devices running Linux. Nowadays, it targets a wide range of networked embedded devices such as IP cameras, home routers (many vendors involved), and other IoT devices. Since the source code was made public many variants of the Mirai family appeared, infecting many networked devices around the world [12, 40].

Dofloo: (Aka AESDDoS) is a popular malware used to create large scale botnets that often launch DDoS attacks and run cryptocurrency miners on the infected machines [33].

Gafgyt: A malware family which infects Linux systems to launch distributed denial-of-service attacks (DDoS) [33].

CoinMiner: An unwanted malicious software which uses the victim's computational power (CPU and RAM mostly) to mine coins (for example Monero or Zcash) [25].

XorDDos: A Linux Trojan malware with rootkit capabilities that is used to launch large-scale DDoS attacks [33].

Unfortunately, with the data from the abuse databases we are able to identify less than 700 hashes (≈5%), leaving more than 95% of all collected hashes unlabeled. One of the reasons is that abuse databases are unlikely to include all variants of the same malware, particularly since even a single bit flip can alter the hash and render the variant unrecognizable. Another reason is the lack of reports, since not every user reports malicious activity.

Clustering. To gain deeper insights into the attacks and to mitigate these limitations, we employ a clustering algorithm to identify sessions with similar attack behaviors. We classify each session based on the commands executed. Our clustering algorithm analyzes the sets of executed commands across sessions and assigns each session to a cluster based on a calculated similarity score. The process for determining the similarity score between the commands of two sessions involves the following steps.

(1) The commands in each session are split into a set of tokens. For example, the session commands "mkdir /tmp;cd /tmp" are tokenized into ["mkdir", "/tmp", "cd", "/tmp"]).

(2) We calculate the similarity score between two sessions using the Damerau-Levenshtein Distance (DLD) [32, 52, 63]. For this calculation, each token is treated as a single character. For instance, for two sessions, one that executes the command "mkdir /tmp" and the other executes "cd /tmp", the DLD would be 1, as only one token needs to be modified. This token-based approach offers increased robustness against frequent changes to IP addresses, filenames, or other forms of obfuscation, enhancing the effectiveness of the clustering.

For our classification, we employ the K-Means algorithm using the scoring function described earlier. To determine the optimal number of clusters, we utilize the elbow method, which involves plotting the "within-cluster sum of squares" (WCSS) against the number of clusters. The "elbow point", where the rate of decrease significantly slows, indicates the ideal number of clusters. Additionally, we consider the Silhouette Score, which evaluates how well each point fits within its cluster compared to other clusters. Based on these metrics, we select 90 clusters.

Figure 5 presents the distance matrix derived from the normalized DLD. The clusters are sorted based on the average number of tokens per cluster. As a result, "Cluster 1" contains sessions with the shortest average number of tokens, while "Cluster 90" contains sessions with the longest average number of tokens. Each of these clusters tries to represent a different type of bot based on the commands that it executed. We use this classification as a fingerprinting technique to identify different malicious bots based on the sequence of commands that they execute.

Clustering. To gain deeper insights into attacker behaviors and overcome the limitations of simple regex-based classification, we apply an unsupervised clustering approach to group sessions that exhibit similar command execution patterns. Each session is represented by the sequence of commands it executes, which we transform into a tokenized representation prior to clustering.

(1) *Tokenization.* Commands within each session are first split into tokens corresponding to meaningful substrings (e.g., commands, arguments, file paths). For example, the session string "mkdir /tmp;cd /tmp" is tokenized into ["mkdir", "/tmp", "cd", "/tmp"]. This token-level representation allows us to capture both command usage and argument structure, while remaining resilient to superficial variations such as spacing, reordering, or minor obfuscations.

(2) *Similarity measure.* To quantify the similarity between two sessions, we employ the Damerau–Levenshtein Distance (DLD) [32, 52, 63] on their token sequences. Each token is treated as a single unit, analogous to a character in a string. For example, the sessions "mkdir /tmp" and "cd /tmp" differ by only one token, yielding a DLD of 1. This token-based formulation is robust against frequent attacker obfuscation strategies (e.g., varying IP addresses, file names, or temporary directory paths), since such changes typically affect only isolated tokens without altering the overall behavioral pattern.

(3) *Clustering algorithm.* The token-based DLD provides a distance value that expresses how similar two sessions are. To apply K-Means, we make use of these pairwise distances by constructing a distance matrix across all sessions. This matrix captures how close or far apart the sessions are in terms of their command sequences. Based on this representation, K-Means partitions the sessions into groups such that sessions with lower distance values (i.e., more similar command sequences) are assigned to the same cluster, while dissimilar sessions are placed in different clusters.

(4) *Cluster selection.* To determine the optimal number of clusters, we combine two established approaches: (i) the elbow method, which identifies the point at which additional clusters yield diminishing improvements in the within-cluster sum of squares (WCSS), and (ii) the Silhouette Score, which measures how well each session fits within its assigned cluster relative to neighboring clusters. The convergence of these two metrics indicated that $k = 90$ provides a good balance between capturing behavioral diversity and avoiding over-fragmentation.

Figure 5 presents the distance matrix derived from the normalized DLD. The clusters are ordered by the average number of tokens per session: "Cluster 1" contains sessions with the shortest average length, while "Cluster 90" contains those with the longest. Each cluster thus represents a distinct type of bot, characterized by the commands it executes. We leverage this classification as a fingerprinting technique to identify and differentiate malicious bots based on their command execution sequences.

Analyzing the cluster formation allowed us to associate the largest clusters with specific bot types.

Cluster 2: Associated with *Gafgyt* bots.
Cluster 3: Associated with *Mirai*.
Cluster 4: A mix of *Mirai* and *CoinMiner*.
Cluster 6: Associated with *XXorDDoS*.

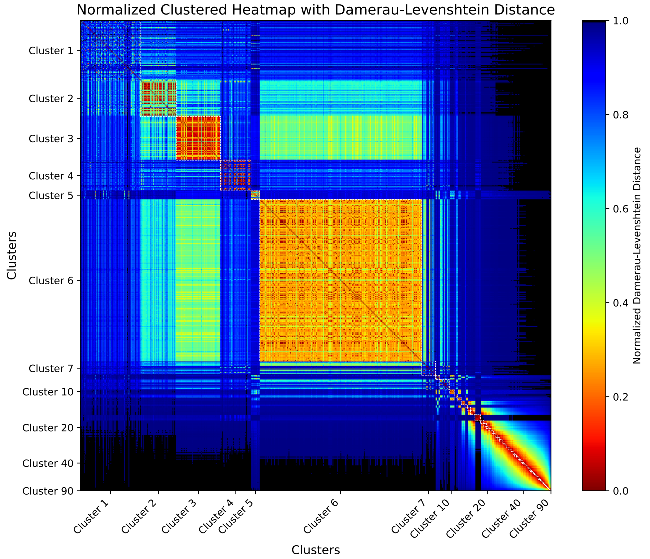


Figure 5: Normalized DLD Matrix.

We observe that **Cluster 1** exhibits the highest variability in terms of normalized DLD. This cluster also has the lowest average number of tokens. Typically, sessions in this cluster start with a `cd` command to change to a specific folder (sometimes preceded by a `mkdir` command to create the folder), followed by a `wget` or `curl` command to download a file, a `chmod` command to modify its permissions, and then the execution of the file, optionally followed by an `rm` command to clean up. The significant variation within this cluster arises from the diversity in folder names, IPs/domains, and downloaded filenames. The next step involves taking all the hashes within each cluster and cross-referencing them with abuse databases. When we correlate the hashes from sessions in **Cluster 1** with abuse databases, we identify multiple types of malware, including *Mirai*, *Dofloo*, *CoinMiner*, and *Gafgyt*. This diversity is expected, as the observed commands represent the minimum required to spread a malicious file.

The remaining clusters contain hashes that are either unknown to abuse databases or generically labeled as malicious, offering limited additional insights. Notably, the five labeled clusters account for over 90% of all sessions involving files. The temporal distribution of these sessions across clusters is shown in Figure 6.

Certain cluster/bots, such as "C-1 (*Mirai, Dofloo, CoinMiner, Gafgyt*)" and "C-6 (*XXorDDoS*)", are observed consistently throughout the entire period, while others, like "C-2 (*Gafgyt*)" and "C-3 (*Mirai*)", appear in intermittent waves. This wave-like behavior aligns with previous research [12, 62]. The recent rise in Mirai instances corresponds with findings by Griffioen et al. [40], who reported increased Mirai scanning activity seen in 2024. This trend points to a resurgence of the original Mirai variant, as verified by abuse databases showing that multiple hashes from this period match the classic Mirai botnet. The Mirai samples identified in spring 2024 include Corona, Kyton, and Ares [11, 92].

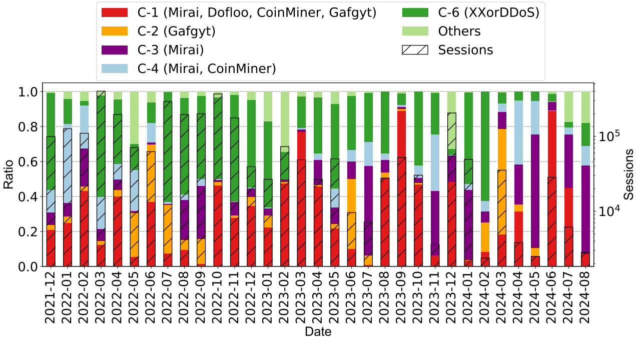


Figure 6: Top 5 clusters (bots) over time.

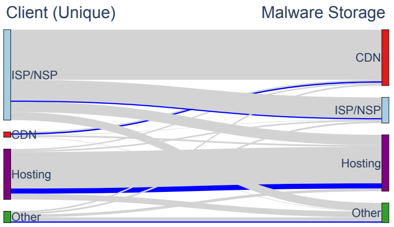


Figure 7: Sankey plot: # of client IPs by AS type vs. # malware storage IP by AS type. Flows correspond to client IP, malware storage IP pairs.

Another interesting observation is the sudden stop in activity from the *XXorDDoS* bot in 2024. This suggests one of two possibilities: (a) the bot was stopped or taken down, either by the attacker himself or through other means, or (b) the bot's behavior changed.

From our data, the first possibility appears more likely. We observe no clear evidence supporting the latter, as there is no spike in other clusters, nor do we find *XXorDDoS* hashes reappearing in other clusters. However, it remains possible that the bot's behavior changed along with its variant, and the new variant's hash has not yet been reported to abuse databases. Either way, this represents a clear indication of a change in the activity of this malicious actor.

As a takeaway, clustering sessions based on executed commands and correlating the results with file hash information from abuse databases proves to be an effective method for gaining a comprehensive overview of bot activity. This approach shows potential for identifying bots and understanding their behavior over time.

7 MALWARE STORAGE LOCATIONS

In this section, we examine the IP addresses involved in download commands that serve as malware storage locations. We start by investigating whether the honeypot client IP—the IP address responsible for brute-forcing and gaining access—is the same as the malware storage location IP. Surprisingly, we find that in 80% of sessions involving downloads, the malware storage location IP differs from the connected client IP. Furthermore, although we see the download commands originating from over 32k unique IPs connecting to the honeypot during our study, only around 3k IPs served as malware storage locations. Given this one-order-of-magnitude difference, we examined whether these malware storage IPs had been previously reported. Abuse databases (recall Section 3.4) reveal that 56% of the malware loader IPs have been reported.

Enriching the dataset with additional attributes, such as the origin AS and AS type associated with each IP (refer to Section 3.5), allows us to leverage this supplementary information effectively. We group the client IPs and the malware location IPs by AS type and generated a Sankey diagram of IP pairs, as shown in Figure 7. On the left side, we display the client IPs of the sessions, and on the right, the malware location IPs. Flows where the client IP matches the loader IP are shown in blue, while flows where they differ are shown in gray. Most client IPs are in ISP/NSP AS types, which aligns with expectations, as attackers often use end-hosts for attacks. Conversely, the majority of malware storage IPs are hosted in cloud environments, particularly in CDNs or Hosting ASes, with only a small fraction within ISP/NSP networks. For more details, see Appendix E.

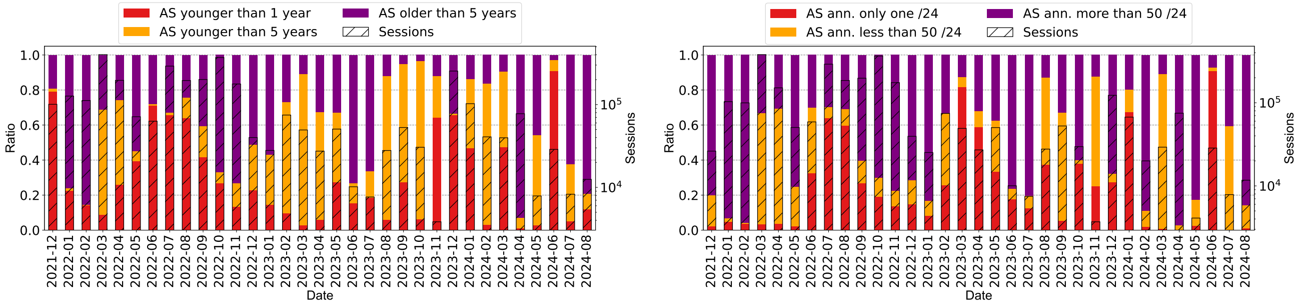
Storage ASes. We conducted a deeper analysis of the ASes used to host malicious files and identified a total of 388 ASes. Among these, 358 are either labeled as hosting ASes or provide hosting services (e.g., renting VMs or website hosting), while 30 are ISPs. Additionally, 36 of these ASes are currently "down", meaning they do not announce any prefixes.To ensure accuracy, AS information was collected using historical WHOIS data [81], meaning that the classification reflects the state of each AS at the time the session was recorded.

Next, we examined the age of the ASes at the time of the malware download on the honeypot (see Figure 8(a)). We found that in more than 35% of cases, the AS was registered in the last year, and in more than 70% of cases, the AS was registered in the last 5 years. It is also important to note that during the data collection period, approximately 1,500 new ASes were registered globally. Out of the 358 hosting ASes we identified, around 35% (≈125 ASes) were registered in the last year. Meaning that roughly 10% of all newly registered ASes during this period were involved in malicious file distribution, representing a non-negligible share of the overall AS growth. This led us to hypothesize that attackers are more successful in abusing recently registered, or smaller ASes, or may even deploy their own AS.

We also analyzed the size of the ASes based on the number of /24 prefixes they announce, focusing only on IPv4 since our investigation centers on attacks targeting IPv4 (see Figure 8(b)). We found that approximately 20% of ASes announce only a single /24 prefix, and around 50% of ASes announce fewer than 50 /24 prefixes. For this analysis, we deaggregated the announced prefixes to enable a fair comparison of AS sizes. These findings further support our hypothesis that attackers may prefer using recently registered and smaller ASes for malicious storage purposes.

To gain a deeper understanding of the malware storage ecosystem, we analyzed the duration for which an IP remains active in the honeynet. Specifically, we examined the reappearance of IPs across different time intervals: one week, four weeks, one year, and the entire dataset collection time window (see Figure 9).

For the one-week recall interval, we observe that in 50% of cases, the IP storing the malicious file is active for only one day. Another 20% remain visible for up to four days, while approximately 30% are active for the entire week. When analyzing longer recall intervals, we find a significant number of IPs being reused over time. On



(a) AS age of malware storage locations.

(b) AS size of malware storage locations.

Figure 8: AS analysis of malware storage locations.

average, 25% of IPs reappear after at least six months. However, this number can reach nearly 50%, as observed during the spring of 2023.

The repeated use of the same IP for hosting malicious files over extended periods suggests that the machines storing these files were not taken down. Additionally, the reappearance of these IPs after long intervals (exceeding six months) suggests that the attacker may employ a pool of machines, rotating their usage to evade temporary blocklists. Another important takeaway from this figure is that the long duration of our investigation enables us to better observe the extent of IP reusability over time.

8 ATTACKER LOGIN ATTEMPTS

In this section we focus on specific login attempts as well as top login credentials used by malicious actors.

We start our investigation with the most commonly used passwords for accessing the honeypots. Recall that the honeypots are configured to allow password-based SSH authentication using the username “root” and by supplying any password except “root”. Figure 10 reveals the most frequent passwords used over the entire observation period. While passwords “*admin*” or “*1234*” are not at all surprising (most like the combination “root:admin” is now on the top of all brute-force dictionary lists), the prominence of other passwords was unexpected.

The plot also highlights an interesting correlation between the passwords “dreambox” and “vertex25ektks123”, which appear to be synchronized in usage—a pattern that is not coincidental. Both passwords are known default credentials for popular TV boxes. The password “dreambox” is commonly set as the default for all `Dreambox Enigma(1)` models—specifically models 500, 500+, 5600, 5620, 600, 7000, and 7020 [4, 5]. Similarly, “vertex25ektks123” is the default password for the Dasan H660DW TV Box [59].

Sessions using these passwords could potentially be part of the same botnet structure, as they (1) appear to target similar device types (e.g., TV boxes) and (2) exhibit a consistent modus operandi: the bot first attempts to download a file from a remote server using the `wget` command and then executes it. Abuse database checks reveal a small number of hashes associated with these sessions, all labeled as “Mirai”. This finding indicates that this botnet leverages default credentials to exploit specific IoT devices, likely incorporating them into the broader Mirai botnet.

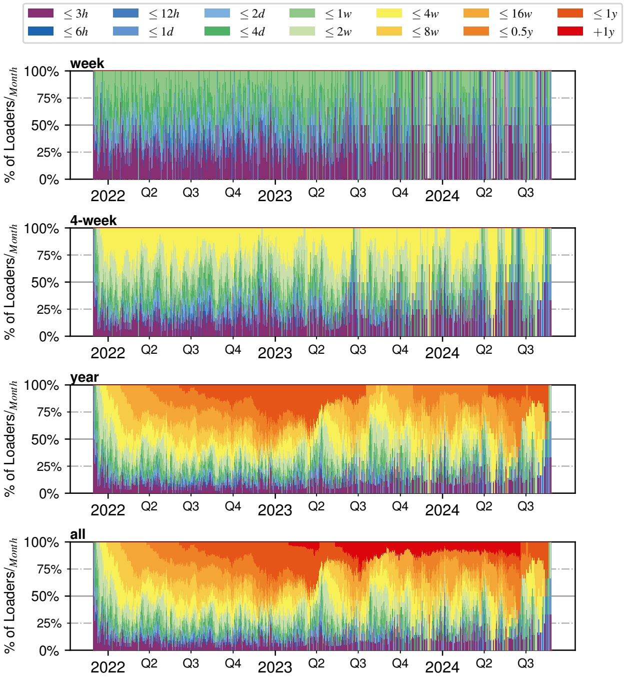


Figure 9: Malware storage activity days over time.

The most intriguing case is the top password in our dataset, “3245gs5662d34”. We observe no commands executed after login with this password, suggesting that some attackers may focus on gaining access without further interaction—potentially marking targets for future attacks or testing for vulnerabilities.

Our data shows a substantial number of sessions using this password—over 24 million—starting from December 8, 2022, at 18:00 (UTC). Despite the scale of these login attempts, no commands are executed post-login, as the bot simply logs in and performs no additional actions. Analyzing the client IPs reveals that this attack originates from over 125k unique IPs. This volume is notable and is only comparable in scale to one other significant attack in our dataset, which we refer to as “*mdrfckr*” and discuss in Section 9.

This password appears in multiple blog posts and forums, yet specific information about the actor or the intended purpose remains unclear. One study [105] suggests that it might be a default credential for the Polycom CX600 IP telephone.

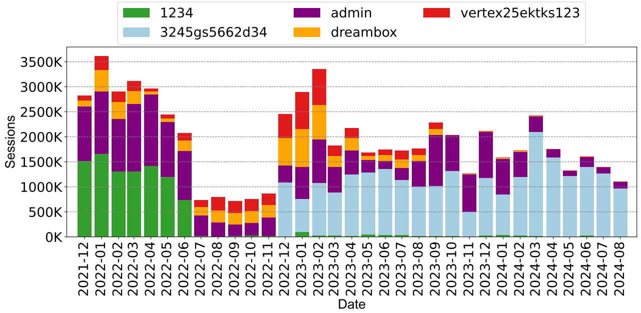


Figure 10: Sessions that record intrusions using one of the Top 5 passwords recorded by the honeynet.

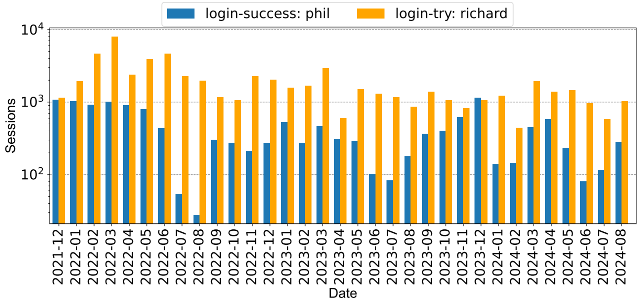


Figure 11: Login attempts using default Cowrie usernames, "richard" and "phil", observed over time.

The sudden appearance of this password might indicate a more targeted attack against specific devices or infrastructure. If the Polycom CX600 IP telephone is indeed the target, the attacker might exploit this default credential to gain unauthorized access and potentially use the device to spy or gather information. Moreover, if this is the case, the timing of the attack could be linked to the acquisition of Poly (formerly Polycom) by HP [41]. The Polycom CX600 IP telephone is considered a legacy device, originating from the era when Polycom used Microsoft Lync and has not received updates since 2017 [57].

Cowrie Default Credentials. While investigating intrusion logins, we identified that the Cowrie honeypots allowed logins for a user other than "root", specifically the username "phil". This behavior is not accidental: the usernames "richard" and "phil" are well-known defaults in Cowrie honeypot deployments [66]. In 2020, a Cowrie update replaced the earlier default username "richard" with "phil". Analysis of these usernames revealed consistent login attempts throughout the observation period. Since our honeynet runs a later version of Cowrie, we observed successful connections using the "phil" username. Figure 11 shows the number of sessions connecting with these default credentials ("phil") as well as attempts to connect with "richard" over time.

Although the sessions connecting with the "phil" credential are relatively low in volume (≈30k), they originate from over 10k unique IPs and span more than 1k ASes, suggesting a broad and distributed probing activity. Our analysis indicates that most of these IPs are located in ISP/NSP or Hosting ASes.

Most interestingly, in the vast majority of cases (over 90% of all sessions), attackers terminate the connection immediately after logging in with the "phil" credential and do not issue any further commands. Moreover, the same client IP does not typically reconnect afterwards. This strongly suggests that the purpose of these logins is not to compromise the host, but rather to detect the presence of Cowrie honeypots by exploiting the fact that "phil" is a default credential. In other words, the attackers appear to use these connections as a reconnaissance or fingerprinting technique to identify honeypot deployments, rather than to engage in sustained malicious activity.

This example indicates that attackers in the field are knowledgeable and adaptive. It is well-documented [53, 69, 78] that attackers employ strategies to evade anti-malware defenses, such as obfuscation, hiding file extensions, encoding in base64, or promptly removing files. As previously discussed in Section 2, honeypots play a crucial role in enhancing Internet security. We see that attackers are increasingly aware of the deployments of honeypots and the risks associated with their methods being recorded. Malicious actors appear to perform reconnaissance activities, like scanning for honeypot-specific usernames, rather than indiscriminately launching attacks, signaling an evolution in attacker tactics toward more cautious approaches.

9 "*MDRFCKR*"—CASE STUDY

The largest attack in our dataset, both by session count and unique client IPs, is from a bot we refer to as "*mdrfckr*". The name is derived from the SSH public key label that the attacker installs to maintain persistence. Over the observation period, we recorded more than 46 million sessions from over 270k unique client IPs associated with this attack (see Figure 12). The "*mdrfckr*" bot is believed to be linked to the Outlaw Hacking Group [86, 103], which has been active since 2018.

Analysis of commands collected by the honeypots shows that this attack installs a public key, ensures the victim cannot easily remove it (by locking out the victim through "root" password changes), and conducts reconnaissance on the machine. External sources, such as reports from actual victims [1], indicate that after establishing persistence, the attacker uses `rsync` to load malicious files and modifies the crontab file to execute them. This method allows the attacker to evade honeypot logs by minimizing detectable activity.

This attack, often associated with cryptocurrency mining, has been analyzed multiple times. Researchers have examined both the executed commands [21, 22] and the specific malicious files downloaded by the attacker [56, 86]. Abuse databases label the public key hash used in this attack as either "CoinMiner" or "Malicious", consistent with its association with cryptocurrency mining activities.

Cross-checking the IPs associated with the credential "3245gs5662d34" (see Section 8) attack against those used by the "*mdrfckr*" bot during the same period reveals a 99.4% overlap. Additionally, the number of sessions recorded monthly for both attacks is nearly identical, as shown in Figure 13. This finding suggests two possible scenarios. First, it could imply that the "3245gs5662d34" credential attack and the "*mdrfckr*" bot are managed by the same actor, indicating that the "*mdrfckr*" actor has expanded its operations in a new direction. Alternatively, both attacks may be run by

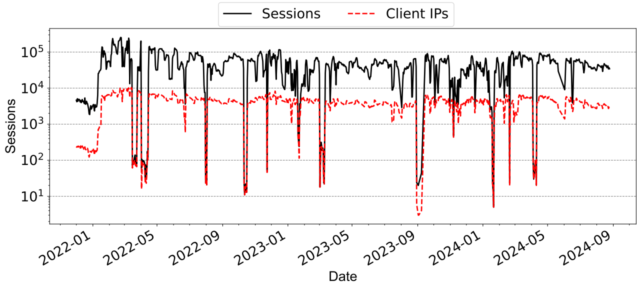


Figure 12: A temporal view of all sessions involving "*mdrfckr*" actor recorded by the honeynet.

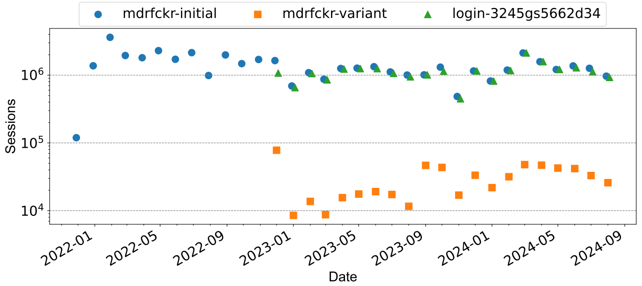


Figure 13: Behavior change in the "*mdrfckr*" bots and correlation to "3245gs5662d34" credential attack.

separate actors who share the same infrastructure, pointing toward a shared "malicious infrastructure as a service" model.

Another notable observation is a behavior change in part of the "*mdrfckr*" sessions, coinciding precisely with the onset of the "3245gs5662d34" attack. This new variant, which we refer to as "mdrfckr-variant", displays several key changes from the original "*mdrfckr*" attack: (1) it no longer changes the root password, (2) removes the files `/tmp/auth.sh` and `/tmp/secure.sh`, (3) kills any processes associated with `auth.sh` and `secure.sh`, and (4) clears the `/etc/hosts.deny` file. As shown in Figure 13, this variant—orange colored squares—is at least an order of magnitude smaller in scale than the original ("mdrfckr-initial", blue colored circles).

Further investigation into these modifications reveals that `/tmp/auth.sh` and `/tmp/secure.sh` are associated with another "CoinMiner" botnet known as *WorkMiner* [2], which first appeared in 2021 as a successor to the *Mozi* botnet [87]. These scripts are designed to defend infected machines against other brute-force attacks by adding offending IPs to the `/etc/hosts.deny` file. This behavior suggests that the "*mdrfckr*" actor seeks to disable *WorkMiner's* blocking functionality, likely to ensure uninterrupted access to the machine. However, only these two specific scripts are removed, leaving *WorkMiner's* mining operations unaffected.

To further understand the "*mdrfckr*" attacker, we cross-referenced their IPs with labeled malicious IP lists, including the C2-Daily Feed [3] and the Killnet Proxy IP list [61]. When correlating the "*mdrfckr*" IP list with the Killnet list, we discovered 988 overlapping IPs. Killnet [6] is a known pro-Russian group, and IPs in their list are often associated with DDoS attacks, indicating that the "*mdrfckr*" actor may be involved in more than just cryptocurrency mining.

Examining the typical behavior of the "*mdrfckr*" bot over the entire observation period (see Figure 12), we observe that it generally generates around 100k sessions per day from approximately 7k unique IPs. However, there are brief periods where this activity drops to around 100 sessions per day from only ≈10 unique IPs. The low activity at the end of 2021 likely resulted from the recent deployment of the honeynet, which needed time to become a known target.

We confirm that the honeypots were fully operational during these periods. Furthermore, the rapid return to "normal" activity—recovering within hours—suggests a deliberate reduction rather than a takedown, which would likely exhibit a more gradual recovery in activity levels. Our hypothesis is that these drops could correlate with other coordinated attack activities or strategic pauses, suggesting that the "*mdrfckr*" actor may intermittently reallocate resources, possibly toward other attacks. This hypothesis is supported by following findings. Only during these low-activity periods do we observe sessions where the "*mdrfckr*" bot executes base64-encoded scripts after installing the SSH key. We decoded and analyzed these scripts, finding that they are variations of three distinct functionalities:

Cryptominer setup: Similar to the cryptominer found in the repository of a previously compromised victim.

Shellbot installation: This backdoor uses IRC to provide botnet control and is often associated with DDoS attacks.

Cleanup script: A script that thoroughly terminates a set of specific processes, likely to hinder detection.

Note, that since the honeypot cannot capture files transferred via `rsync`—a technique known to be used by this actor—it is likely that we are missing some malicious files that could further clarify this out-of-the-ordinary behavior.

To analyze this further, we examined the IPs responsible for uploading base64-encoded scripts and identified 1,624 unique IPs. These IPs primarily belong to Hosting AS and ISP/NSP providers. Most of these IPs appeared only once, and there was no overlap among IPs across different periods of reduced activity, indicating a dispersed and dynamic infrastructure, even during out-of-the-ordinary attacks.

Focusing on base64 scripts, the "cleaning" script appears to specifically target and remove processes and scripts related to a set of 8 IPs. These IPs exhibit a variety of open ports:

- Four IPs have port 22 open (SSH).
- One IP has ports 1337 and 9999 open and is running ZNC [8], an IRC bouncer often used in botnet command-and-control (C&C) infrastructure.
- One IP has ports 80 and 3306 (MariaDB) open.
- One IP has port 8080 open.
- One IP has port 43, 80, and 443 open.

Given the consistent inclusion of these IPs in the script, their strategic network positioning, and the nature of their open ports, it is plausible that these IPs function as part of the botnet's C&C infrastructure. The continued presence of these IPs in the script, highlights their critical role in maintaining control and coordination across the botnet.

In collaboration with the Shadowserver Foundation, we analyzed their Special Report on compromised SSH servers [85]. This report identifies hosts running SSH services that are known to be compromised due to the presence of malicious public SSH keys facilitating unauthorized remote access. Our analysis revealed that the "*mdrfckr*" key has compromised over 13k servers globally, making it the most prevalent malicious SSH key identified in their dataset.

10 DISCUSSION

Limitations. In this study, we utilized Cowrie honeypots, which inherently restrict our observations to the SSH service. Additionally, Cowrie is a medium-interaction honeypot: it can emulate certain system behaviors, but it cannot execute actual operating system processes. As observed, this limitation can sometimes alert attackers to the presence of a honeypot. Another important constraint of the honeypots that are used – they only permit logins using the username "root". Consequently, our dataset does not capture attacks that specifically target non-root accounts, potentially overlooking behaviors associated with other common usernames. An exception to this rule is the inclusion of Cowrie's built-in default accounts (e.g., "richard" and "phil"), which are used primarily for fingerprinting the honeypot rather than for sustained interactions. Finally, although the honeyfarm offers broad geographic distribution, there are still notable gaps in sensor coverage in certain regions, which may bias the representativeness of our dataset.

Call for Better Honeypots. The current state of the art in honeypot technology appears to be insufficient to address evolving attacker strategies. For example, Cowrie honeypots are increasingly being targeted by attackers who exploit default credentials to identify them. Moreover, certain attackers bypass the honeypot's defenses entirely by abusing it as a proxy for their attacks. Others employ techniques such as `rsync` to evade detection, transferring files without being captured during the attack process.

Several improvements could help address these limitations. First, providing persistent storage would enable honeypots to capture and analyze long-term attacker activities rather than only ephemeral interactions. Second, moving beyond pure emulation toward controlled, real operating system environments would significantly increase realism and reduce the likelihood that attackers detect the deception. This could be achieved through lightweight virtualization or containerization, while still enforcing strict isolation and monitoring to prevent collateral damage. Finally, enhancing coverage of less common protocols and broadening the range of supported user accounts could yield richer insights into attacker behavior that current honeypots fail to expose.

Understanding malicious actors. Our analysis of the "mdrfckr" botnet exemplifies how attackers obfuscate their true objectives by engaging in multiple malicious activities simultaneously. This adaptability and evolution in tactics, likely in response to changing goals or environmental conditions, underscores the complexity of profiling malicious actors. By diversifying their actions, attackers can obscure their overall intentions, making it difficult to fully understand the scope and motivations of their operations through a single dataset.

Longitudinal analysis. This study underscores the growing importance of longitudinal analysis in understanding SSH attacks. While short-term snapshots can highlight immediate trends or specific incidents, they often fail to capture the gradual evolution of attacker strategies and behaviors. Long-term analysis reveals patterns such as the reuse of infrastructure, shifts in tactics, and the emergence of more exploratory methods, which are critical for comprehensively understanding the threat landscape. Without extended observation, key insights—such as the reappearance of malicious resources after dormancy or the adaptation of bots to evade detection—can be easily overlooked. As attackers become increasingly adaptive, longitudinal methodologies are essential for developing effective, forward-looking defense mechanisms that address these evolving threats.

Events correlation. During our investigation of the *mdrfckr*, we sought to identify any significant global events that coincided with the observed drops in bot activity. Our analysis reveals a strong correlation between these periods of reduced activity and several documented attacks during the same timeframes.

2022.03.16—2022.03.24 and 2022.04.02—2022.04.12 A series of (pro-Russian) attacks targeting Ukrainian infrastructure. An actor called "IRIDIUM" suspected in attacks, performed a massive DDoS attack against targets [58].

2022.08.01—2022.08.02: Hits on Infrastructure of a European country supporting Ukraine [90, 91].

2022.10.10—2022.10.16: "Sandworm" (a pro-Russian hacker group) attack against power grid of Ukraine [13, 68, 71], as well as Killnet DDoS attacks against US Airports [37].

2023.03.02—2023.03.10: Attack against "KyivStar"–largest Ukraine mobile operator [14].

2023.09.01—2023.09.08: DDoS attacks against Ukraine public administration and media [30].

2024.01.19—2024.01.21: APT29 (aka Midnight Blizzard)—data theft attack [27].

2024.04.04—2024.04.10: Another "Sandworm" attack against infrastructure of Ukraine [55, 79].

While the correlation between these events and bot activity does not imply causation, we find the pattern both interesting and concerning. In any case, these findings suggest that continuous monitoring of the bot's activity is essential.

11 CONCLUSION

In this study, we conducted a comprehensive longitudinal analysis of SSH-based attacks over a three-year period, shedding light on the evolving strategies and behaviors of attackers. By examining intrusive sessions, executed commands, malicious files, and the infrastructure supporting these attacks, we identified significant shifts in attacker tactics, including increased exploratory behavior and changes in how files are loaded onto compromised targets. Our findings also revealed a trend toward the use of recently registered ASes as malicious storage IPs, as well as evidence of attackers actively scanning for honeypots and, in some cases, exploiting them for their own purposes.

These insights highlight the importance of adapting defense mechanisms to address the evolving tactics of attackers. Our study emphasizes the need for continuous observation and deeper analyses of malicious activity to uncover trends and changes in behavior. While our findings demonstrate the potential of integrating data from diverse sources to gain broader insights, further collaboration and research are necessary to enhance our understanding of attacker strategies and improve the effectiveness of defensive measures in an ever-changing threat landscape.

ACKNOWLEDGMENT

We would like to thank Global Cyber Alliance for sharing the data with us. This work was supported by the European Commission under the Horizon Europe Programme as part of the projects SafeHorizon (Grant Agreement #101168562) and RECITALS (Grant Agreement #101168490).

REFERENCES

[1] 2020. Git, Crypto Miner Hack. https://github.com/dangoldin/crypto-miner-hack.

[2] 2021. WorkMiner Bot. https://www.a1ee.cn/medium/workminer/.

[3] 2024. C2-Daily-Feed. https://github.com/criminalip/C2-Daily-Feed.

[4] 2024. Dreambox Configuration. https://dreamboxedit.com/en/workshop-2/konfiguration/.

[5] 2024. Dreambox Enigma 1. https://dreambox.de/board/index.php?board/15-enigma-1-alle-themen/.

[6] 2024. Killnet: Inside the World's Most Prominent Pro-Kremlin Hacktivist Collective. https://flashpoint.io/intelligence-101/killnet/.

[7] 2024. PeeringDB. https://www.peeringdb.com.

[8] 2024. ZNC. https://wiki.znc.in/ZNC.

[9] AbdelRahman Abdou, David Barrera, and Paul C. van Oorschot. 2016. What Lies Beneath? Analyzing Automated SSH Bruteforce Attacks. In *International Conference on Passwords*. Springer International Publishing, Cham, 72–91.

[10] abuse.ch. 2024. abuse.ch: Fighting Malware and Botnets. https://abuse.ch/.

[11] Antonia Affinito, Stefania Zinno, Giovanni Stanco, Alessio Botta, and Giorgio Ventre. 2023. The evolution of Mirai Botnet Scans over a Six-year Period. *Journal of Information Security and Applications* 79 (2023), 103629.

[12] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. 2017. Understanding the Mirai Botnet. In *USENIX Security Symposium*.

[13] Daryna Antoniuk. 2023. Ukraine energy facility took unique Sandworm hit on day of missile strikes, report says. https://therecord.media/sandworm-attack-ukraine-energy-facility-missile-strikes.

[14] Daryna Antoniuk. 2024. Russian hackers infiltrated Ukrainian telecom giant months before cyberattack. https://therecord.media/russians-infiltrated-kyivstar-months-before.

[15] ArmstrongTechs. 2024. ArmstrongTechs Indicators-of-compromise-IOCs. https://github.com/ArmstrongTechs/Indicators-of-compromise-IOCs?tab=readme-ov-file.

[16] P. Baecher, M. Koetter, and G. Wicherski. 2023. Nepenthes on GitHub. https://github.com/jrwren/nepenthes.

[17] Timothy Barron and Nick Nikiforakis. 2017. Picky attackers: Quantifying the role of system properties on intruder behavior. In *Annual Computer Security Applications Conference*.

[18] Fabian Bäumer, Marcus Brinkmann, and Jörg Schwenk. 2024. Terrapin Attack: Breaking SSH Channel Integrity By Sequence Number Manipulation. In *33rd USENIX Security Symposium (USENIX Security 24)*. 7463–7480.

[19] Melike Başer, Ebu Yusuf Güven, and Muhammed Ali Aydın. 2021. SSH and Telnet Protocols Attack Analysis Using Honeypot Technique: Analysis of SSH AND TELNET Honeypot. In *6th International Conference on Computer Science and Engineering (UBMK)*. 806–811. https://doi.org/10.1109/UBMK52708.2021.9558948

[20] Pamela Beltrán-García, Eleazar Aguirre-Anaya, Ponciano Jorge Escamilla-Ambrosio, and Raúl Acosta-Bermejo. 2019. IoT botnets. In *Telematics and Computing: 8th International Congress, WITCOM 2019, Merida, Mexico, November 4–8, 2019, Proceedings 8*. Springer, 247–257.

[21] Blog Port22. 2022. mdrfckrs – part one. https://blog.port22.dk/mdrfckrs-part-one/.

[22] Blog Port22. 2022. mdrfckrs – part two. https://blog.port22.dk/mdrfckrs-part-two/.

[23] Matteo Boffa, Giulia Milan, Luca Vassio, Idilio Drago, Marco Mellia, and Zied Ben Houidi. 2022. Towards NLP-based Processing of Honeypot Logs. In *IEEE European Symposium on Security and Privacy Workshops*.

[24] Matthew L Bringer, Christopher A Chelmecki, and Hiroshi Fujinoki. 2012. A survey: Recent advances and future trends in honeypot research. *International Journal of Computer Network and Information Security* 4, 10 (2012), 63.

[25] Jonah Burgess, Domhnall Carlin, Philip O'Kane, and Sakir Sezer. 2019. MANiC: Multi-step Assessment for Crypto-miners. In *2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*. 1–8. https://doi.org/10.1109/CyberSecPODS.2019.8885003

[26] Alessandro Cantelli-Forti and Michele Colajanni. 2018. Adversarial fingerprinting of cyber attacks based on stateful honeypots. In *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 19–24.

[27] CERT-EU: The Cybersecurity Service for the Union institutions, bodies, offices and agencies. 2024. Cyber Security Brief 24-02 - January 2024; APT29 cyberattacks. https://cert.europa.eu/publications/threat-intelligence/cb24-02/.

[28] Andrei Costin and Jonas Zaddach. 2018. IoT malware: Comprehensive survey, analysis framework and case studies. *BlackHat USA* 1, 1 (2018), 1–9.

[29] Cowrie. 2024. Cowrie on GitHub. https://github.com/cowrie/cowrie.

[30] Cyber Peace Institute. 2023. Cyber Dimensions of the Armed Conflict in Ukraine. https://cyberconflicts.cyberpeaceinstitute.org/report.

[31] David Dagon, Xinzhou Qin, Guofei Gu, Wenke Lee, Julian Grizzard, John Levine, and Henry Owen. 2004. Honeystat: Local worm detection using honeypots. In *Recent Advances in Intrusion Detection: 7th International Symposium, RAID 2004, Sophia Antipolis, France, September 15-17, 2004. Proceedings 7*. Springer, 39–58.

[32] Fred J Damerau. 1964. A technique for computer detection and correction of spelling errors. *Commun. ACM* 7, 3 (1964), 171–176.

[33] Michele De Donno, Nicola Dragoni, Alberto Giaretta, and Angelo Spognardi. 2018. DDoS-Capable IoT Malwares: Comparative Analysis and Mirai Investigation. *Security and Communication Networks* 2018, 1 (2018), 7178164. https://doi.org/10.1155/2018/7178164 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1155/2018/7178164

[34] DutchSec B.V. 2023. Honeytrap on GitHub. https://github.com/honeytrap/honeytrap.

[35] Tobias Fiebig. 2013. Getting back at Trudy: SSH Botnet Member Credential Collection using Connect Back Honeypots. (2013). Universiteit van Amsterdam.

[36] Basil Fillan, Ben Cartwright-Cox, Cynthia Revström, Elimalko Saado, eraters, Igloo22225, Jeroen Massar, Job Snijders, Molly Miller, Puck Meerburg, Roelf Wichertjes, Tim Stallard, and Tommy Bowditch. 2023. BGP.tools. https://bgp.tools/.

[37] Jason Firch. 2022. Russian Hacktivists, Killnet, Take Down US Airport Websites. https://purplesec.us/breach-report/killnet-ddos-airport-websites/.

[38] Xinwen Fu, Wei Yu, Dan Cheng, Xuejun Tan, Kevin Streff, and Steve Graham. 2006. On Recognizing Virtual Honeypots and Countermeasures. In *2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing*. IEEE, 211–218.

[39] Vincent Ghiette, Harm Griffioen, and Christian Doerr. 2019. Fingerprinting Tooling used for SSH Compromisation Attempts. In *RAID*.

[40] Harm Griffioen, Georgios Koursiounis, Georgios Smaragdakis, and Christian Doerr. 2024. Have you SYN me? characterizing ten years of Internet scanning. In *ACM Internet Measurement Conference*.

[41] Hewlett Packard. 2022. HP Poly acquisition. https://www.hp.com/us-en/newsroom/press-releases/2022/hp-inc-completes-acquisition-of-poly.html.

[42] Raphael Hiesgen, Marcin Nawrocki, Alistair King, Alberto Dainotti, Thomas C. Schmidt, and Matthias Wählisch. 2022. Spoki: Unveiling a New Wave of Scanners Through a Reactive Network Telescope. In *USENIX Security Symposium*.

[43] Shane Huntley. 2023. Fog of war: how the Ukraine conflict transformed the cyber threat. (2023). https://blog.google/threat-analysis-group/fog-of-war-how-the-ukraine-conflict-transformed-the-cyber-threat-landscape/

[44] Liz Izhikevich, Manda Tran, Michalis Kallitsis, Aurore Fass, and Zakir Durumeric. 2023. Cloud Watching: Understanding Attacks Against Cloud-hosted Services. In *Internet Measurement Conference*. 313–327.

[45] John P John, Fang Yu, Yinglian Xie, Arvind Krishnamurthy, and Martín Abadi. 2011. Heat-seeking honeypots: design and experience. In *Proceedings of the 20th international conference on World wide web*. 207–216.

[46] Jeremy Kepner, Jonathan Bernays, Stephen Buckley, Kenjiro Cho, Cary Conrad, Leslie Daigle, Keeley Erhardt, Vijay Gadepally, Barry Greene, Michael Jones, et al. 2022. Zero Botnets: An Observe-Pursue-Counter Approach. *arXiv preprint arXiv:2201.06068* (2022).

[47] Pratibha Khandait, Namrata Tiwari, and Neminath Hubballi. 2021. Who is trying to compromise your SSH server? An analysis of authentication logs and detection of bruteforce attacks. In *Adjunct Proceedings of the 2021 International Conference on Distributed Computing and Networking*. 127–132.

[48] Joseph Khoury, Morteza Safaei Pour, and Elias Bou-Harb. 2022. A Near Real-Time Scheme for Collecting and Analyzing IoT Malware Artifacts at Scale. In *Proceedings of the 17th International Conference on Availability, Reliability and Security* (Vienna, Austria) *(ARES '22)*. Association for Computing Machinery, New York, NY, USA, Article 32, 11 pages. https://doi.org/10.1145/3538969.3539009

[49] Kippo. 2019. Kippo on GitHub. https://github.com/desaster/kippo.

[50] Ioannis Koniaris, Georgios Papadimitriou, and Petros Nicopolitidis. 2013. Analysis and Visualization of SSH Attacks using Honeypots. In *Eurocon 2013*.

[51] Igor Kotenko, Alexey Konovalov, and Andrey Shorov. 2010. Agent-based modeling and simulation of botnets and botnet defense. In *Conference on Cyber Conflict. CCD COE Publications. Tallinn, Estonia*. Citeseer, 21–44.

[52] VI Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Proceedings of the Soviet physics doklady* (1966).

[53] Vector Guo Li, Gautam Akiwate, Kirill Levchenko, Geoffrey M Voelker, and Stefan Savage. 2021. iClairvoyance: Inferring blocklist use on the internet. In

Passive and Active Measurement. Springer, 57–75.

[54] Tongbo Luo, Zhaoyan Xu, Xing Jin, Yanhui Jia, and Xin Ouyang. 2017. Iotcandyjar: Towards an intelligent-interaction honeypot for iot devices. *Black Hat* 1 (2017), 1–11.

[55] Jessica Lyons. 2024. Kremlin's Sandworm blamed for cyberattacks on US, European water utilities. https://www.theregister.com/2024/04/17/russia_sandworm_cyberattacks_water/.

[56] Malware News. 2019. Mdrfckr: How to stay safe online this festive season. https://malware.news/t/dota-campaign-analyzing-a-coin-mining-and-remote-access-hybrid-campaign/30326.

[57] Microsoft. 2017. Polycom Lync Update 2017. https://support.microsoft.com/en-us/topic/april-2017-cumulative-update-for-microsoft-lync-phone-edition-for-polycom-cx500-polycom-cx600-and-polycom-cx3000-telephones-kb4019529-7c85b54c-e4b1-44a8-b07b-e61d3eb92359.

[58] Microsoft. 2022. An overview of Russia's cyberattack activity in Ukraine. https://www.microsoft.com/en-us/security/security-insider/intelligence-reports/special-report-ukraine.

[59] Nguyen Quang Minh. 2019. Dasan H660DW. https://www.minhng99.cloud/Exploring-router-Dasan_H660DW/.

[60] Iyatiti Mokube and Michele Adams. 2007. Honeypots: Concepts, Approaches, and Challenges. In *Annual Southeast Regional Conference*.

[61] Stephen Mondiguing. 2024. KillNet DDoS Blocklist. https://github.com/securityscorecard/SSC-Threat-Intel-IoCs/tree/master/KillNet-DDoS-Blocklist.

[62] Cristian Munteanu, Said Jawad Saidi, Oliver Gasser, Georgios Smaragdakis, and Anja Feldmann. 2023. Fifteen Months in the Life of a Honeyfarm. In *ACM IMC*.

[63] Gonzalo Navarro. 2001. A guided tour to approximate string matching. *ACM computing surveys (CSUR)* 33, 1 (2001), 31–88.

[64] Marcin Nawrocki, John Kristoff, Raphael Hiesgen, Chris Kanich, Thomas C. Schmidt, and Matthias Wählisch. 2023. SoK: A Data-driven View on Methods to Detect Reflective Amplification DDoS Attacks Using Honeypots. In *IEEE Euro S&P*.

[65] Marcin Nawrocki, Matthias Wählisch, Thomas C. Schmidt, Christian Keil, and Jochen Schönfelder. 2016. A Survey on Honeypot Software and Data Analysis. *CoRR* (2016). http://arxiv.org/abs/1608.06249

[66] Michel Oosterhof. 2025. Cowrie Docs. https://readthedocs.org/projects/cowrie/downloads/pdf/latest/.

[67] Yin Minn Pa Pa, Shogo Suzuki, Katsunari Yoshioka, Tsutomu Matsumoto, Takahiro Kasama, and Christian Rossow. 2016. IoTPOT: A novel honeypot for revealing current IoT threats. *Journal of Information Processing* 24, 3 (2016), 522–533.

[68] James Pearson. 2023. Russian spies behind cyber attack on Ukraine power grid in 2022 - researchers. https://www.reuters.com/technology/cybersecurity/russian-spies-behind-cyberattack-ukrainian-power-grid-2022-researchers-2023-11-09/.

[69] Andreas Pitsillidis, Chris Kanich, Geoffrey M Voelker, Kirill Levchenko, and Stefan Savage. 2012. Taster's Choice: A Comparative Analysis of Spam Feeds. In *Internet Measurement Conference*. 427–440.

[70] Anton O Prokofiev and Yulia S Smirnova. 2019. Counteraction against Internet of Things botnets in private networks. In *2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*. IEEE, 301–305.

[71] Ken Proska, John Wolfram, Jared Wilson, Dan Black, Keith Lunden, Daniel Kapellmann Zafra, Nathan Brubaker, Tyler McLellan, and Chris Sistrunk. 2023. Sandworm Disrupts Power in Ukraine Using a Novel Attack Against Operational Technology. https://cloud.google.com/blog/topics/threat-intelligence/sandworm-disrupts-power-ukraine-operational-technology/.

[72] Niels Provos. 2023. Developments of the Honeyd Virtual Honeypot. https://www.honeyd.org/.

[73] Gokul Kannan Sadasivam, Chittaranjan Hota, and Bhojan Anand. 2016. Classification of SSH Attacks Using Machine Learning Algorithms. In *2016 6th International Conference on IT Convergence and Security (ICITCS)*. 1–6. https://doi.org/10.1109/ICITCS.2016.7740316

[74] Farhan Sadique and Shamik Sengupta. 2021. Analysis of Attacker Behavior in Compromised Hosts During Command and Control. In *ICC 2021 - IEEE International Conference on Communications*. 1–7. https://doi.org/10.1109/ICC42927.2021.9500859

[75] Gabriel Salles-Loustau, Robin Berthier, Etienne Collange, Bertrand Sobesto, and Michel Cukier. 2011. Characterizing Attackers and Attacks: An Empirical Study. In *17th IEEE Pacific Rim International Symposium on Dependable Computing*. 174–183. https://doi.org/10.1109/PRDC.2011.29

[76] Zain Shamsi, Daniel Zhang, Daehyun Kyoung, and Alex Liu. 2022. Measuring and Clustering Network Attackers using Medium-Interaction Honeypots. In *IEEE European Symposium on Security and Privacy Workshops*.

[77] Sachin Kumar Singh, Shreeman Gautam, Cameron Cartier, Sameer Patil, and Robert Ricci. 2024. Where The Wild Things Are: Brute-Force SSH Attacks In The Wild And How To Stop Them. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*. USENIX Association, Santa Clara, CA, 1731–1750. https://www.usenix.org/conference/nsdi24/presentation/singh-sachin

[78] Sushant Sinha, Michael Bailey, and Farnam Jahanian. 2008. Shades of Grey: On the effectiveness of reputation-based "blacklists". In *3rd International Conference on Malicious and Unwanted Software (MALWARE)*. IEEE, 57–64.

[79] SOC Radar. 2024. Major Cyberattack April 2024: Sandworm. https://socradar.io/major-cyber-attacks-in-review-april-2024/.

[80] Brett Stone-Gross, Marco Cova, Lorenzo Cavallaro, Bob Gilbert, Martin Szydlowski, Richard Kemmerer, Christopher Kruegel, and Giovanni Vigna. 2009. Your botnet is my botnet: analysis of a botnet takeover. In *Proceedings of the 16th ACM Conference on Computer and Communications Security* (Chicago, Illinois, USA) *(CCS '09)*. Association for Computing Machinery, New York, NY, USA, 635–647. https://doi.org/10.1145/1653662.1653738

[81] Florian Streibelt, Martina Lindorfer, Seda Gürses, Carlos H Gañán, and Tobias Fiebig. 2023. Back-to-the-Future Whois: An IP Address Attribution Service for Working with Historic Datasets. In *International Conference on Passive and Active Network Measurement*. Springer, 209–226.

[82] Cymru Team. 2024. IP to ASN mapping. *http://www.team-cymru.org/IP-ASN-mapping.html* (2024).

[83] Simon Nam Thanh Vu, Mads Stege, Peter Issam El-Habr, Jesper Bang, and Nicola Dragoni. 2021. A survey on botnets: Incentives, evolution, detection and current trends. *Future Internet* 13, 8 (2021), 198.

[84] The Honeynet Project. 2023. The Honeynet Project. https://www.honeynet.org/.

[85] The ShadowserverFoundation. 2024. Compromised SSH Host Special Report. https://www.shadowserver.org/what-we-do/network-reporting/compromised-ssh-host-special/.

[86] Trendmicro. 2018. Trendmicro, Outlaw Hacking Group. https://www.trendmicro.com/de_de/research/18/k/outlaw-group-distributes-botnet-for-cryptocurrency-mining-scanning-and-brute-force.html.

[87] Alex Turing, Hui Wang, and Genshen Ye. 2021. The death of Mozi. https://blog.netlab.360.com/the_death_of_mozi_cn.

[88] Shreya Udhani, Alexander Withers, and Masooda Bashir. 2019. Human vs Bots: Detecting Human Attacks in a Honeypot Environment. In *2019 7th International Symposium on Digital Forensics and Security (ISDFS)*. 1–6. https://doi.org/10.1109/ISDFS.2019.8757534

[89] Joni Uitto, Sampsa Rauti, Samuel Laurén, and Ville Leppänen. 2017. A survey on anti-honeypot and anti-introspection methods. In *Recent Advances in Information Systems and Technologies: Volume 2 5*. Springer, 125–134.

[90] Ukrainska Pravda. 2024. US charges 6 Russians with cyberattack on Ukraine and NATO before 2022 full-scale invasion. https://www.pravda.com.ua/eng/news/2024/09/5/7473672/.

[91] US Department of Justice. 2024. Russian National Charged for Conspiring with Russian Military Intelligence to Destroy Ukrainian Government Computer Systems and Data. https://www.justice.gov/opa/pr/russian-national-charged-conspiring-russia-military-intelligence-destroy-ukrainian.

[92] Ramyapandian Vijayakanthan, Karley M Waguespack, Irfan Ahmed, and Aisha Ali-Gombe. 2023. Fortifying IoT Devices: AI-Driven Intrusion Detection via Memory-Encoded Audio Signals. In *2023 IEEE Secure Development Conference (SecDev)*. IEEE, 106–117.

[93] VirusTotal. 2024. VirusTotal. https://www.virustotal.com/.

[94] Michael Vrable, Justin Ma, Jay Chen, David Moore, Erik Vandekieft, Alex C Snoeren, Geoffrey M Voelker, and Stefan Savage. 2005. Scalability, fidelity, and containment in the potemkin virtual honeyfarm. In *Proceedings of the twentieth ACM symposium on Operating systems principles*. 148–162.

[95] Jan Vykopal. 2013. Flow-based brute-force attack detection in large and high-speed networks. *Masaryk University (Brno, Czech Republic), PhD Thesis* (2013).

[96] Jan Vykopal, Tomas Plesnik, and Pavel Minarik. 2009. Network-Based Dictionary Attack Detection. In *2009 International Conference on Future Networks*. 23–27. https://doi.org/10.1109/ICFN.2009.36

[97] Matthias Wählisch, Sebastian Trapp, Christian Keil, Jochen Schönfelder, Thomas C Schmidt, and Jochen Schiller. 2012. First insights from a mobile honeypot. *ACM SIGCOMM Computer Communication Review* 42, 4 (2012), 305–306.

[98] Aohui Wang, Ruigang Liang, Xiaokang Liu, Yingjun Zhang, Kai Chen, and Jin Li. 2017. An Inside Look at IoT Malware. In *Industrial IoT Technologies and Applications*, Fulong Chen and Yonglong Luo (Eds.). Springer International Publishing, Cham, 176–186.

[99] Binbin Wang, Yilian Zhang, Minjie Zhu, and Yan Chen. 2020. Design and Implementation of Web Honeypot Detection System Based on Search Engine. In *2020 International Conference on Intelligent Computing, Automation and Systems (ICICAS)*. IEEE, 130–135.

[100] Majda Wazzan, Daniyal Algazzawi, Omaima Bamasaq, Aiiad Albeshri, and Li Cheng. 2021. Internet of Things botnet detection approaches: Analysis and recommendations for future research. *Applied Sciences* 11, 12 (2021), 5713.

[101] Nicholas Wells. 2000. Busybox: A swiss army knife for linux. *Linux Journal* 2000, 78es (2000), 10–es.

[102] Yuming Wu, Phuong M Cao, Alexander Withers, Zbigniew T Kalbarczyk, and Ravishankar K Iyer. 2020. Mining Threat Intelligence from Billion-scale SSH

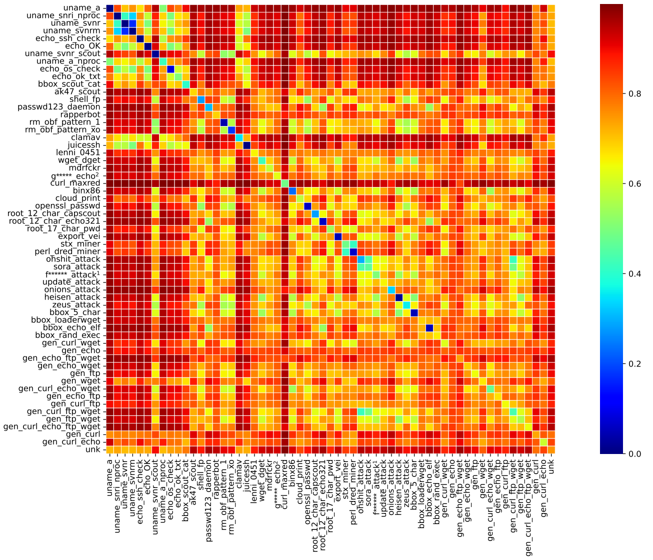


Figure 14: Inter-bot category normalized DLD.

Brute-Force Attacks. (2020).

[103] Yoroi. 2020. Yoroi Outlaw Hacking Group. https://yoroi.company/research/outlaw-is-back-a-new-crypto-botnet-targets-european-organizations/.

[104] Zhenxin Zhan, Maochao Xu, and Shouhuai Xu. 2013. Characterizing Honeypot-Captured Cyber Attacks: Statistical Framework and Case Study. *IEEE Transactions on Information Forensics and Security* 8, 11 (2013), 1775–1789. https://doi.org/10.1109/TIFS.2013.2279800

[105] Zetong Zhao, Shreyas Srinivasa, and Emmanouil Vasilomanolakis. 2023. SweetCam: an IP Camera Honeypot. In *Proceedings of the 5th Workshop on CPS&IoT Security and Privacy*. 75–81.

[106] Jiawei Zhou, Zidong Zhang, Lingyun Ying, Huajun Chai, Jiuxin Cao, and Haixin Duan. 2025. Hey, Your Secrets Leaked! Detecting and Characterizing Secret Leakage in the Wild. In *IEEE Symp. on Security and Privacy*.

APPENDIX

A ETHICS

We discuss the ethical consideration for our work in Section 4.

B COMMANDS ANALYSIS

In Figure 14, we present the average DLD between different command categories. Additionally, Table 1 shows all the category labels and the regex commands used to select the commands. Note the clear separation between the ten command categories located in the top-left corner and the rest. These commands do not alter the honeypots' state (downloading or writing files), but collect system information. Generally, *clamav* and *juicessh*, could also be considered part of this cluster.

C CURL DDOS ATTACK

A sample of the commands used in this campaign is shown in Figure 15. The `curl` commands are harmless to the honeypot, as they do not introduce malicious software. These commands attempt all HTTP methods with varying cookies and target over 100 domains or IP addresses, with some accessed more than 300k times throughout the campaign. Examination of the cookies shows each one is unique, suggesting that this campaign is aimed at either conducting a Denial of Service (DoS) attacks or possibly testing/exploiting stolen cookies.

```
curl https://<X.X.X.X>/ -s -X GET --max-redirs 5 --compressed
--cookie '<hidden-cookie>' --raw --referer '<hidden-URL>'
curl https://<X.X.X.X>/ -s -X POST --max-redirs 5 --compressed
--cookie '<hidden-cookie>' --raw --referer '<hidden-URL>'
...
```

Figure 15: Snippet of a curl with "cookie attack". Referrer, cookies, domain and IP redacted.

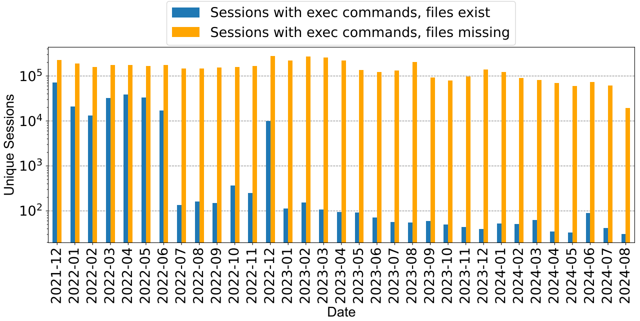


Figure 16: Sessions that execute files. Blue color represents sessions where the executed file is found (and a hash of the file is recorded), while orange color shows sessions where the file was missing (and no hash is recorded). Unique number of sessions (based on commands).

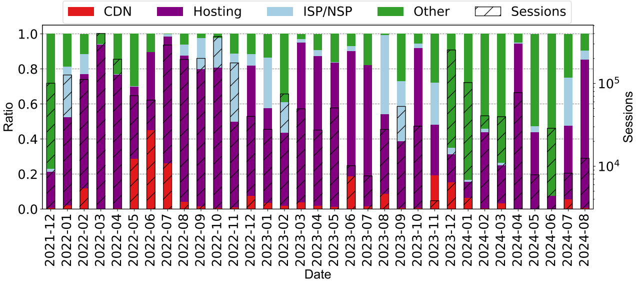


Figure 17: AS Types of Malware storage location seen over time.

D UNIQUE COMMANDS

In analyzing command uniqueness, see Figure 16, we find that sessions labeled as "file missing" exhibit a higher number of unique commands, indicating greater variability compared to sessions where the file is present. This suggests an increased use of obfuscation techniques in these sessions. The spikes in command uniqueness for sessions with files correspond to two distinct bot waves: a prolonged campaign at the start of 2022 and a shorter, intense burst of attacks in December 2022. A thorough analysis showed that both of the waves are most likely related to Mirai, as we are able to find the hashes recorded in those waves labeled as "Mirai" in the abuse datasets.

<table>
<tr><th>Label</th><th>Regular Expresion</th></tr>
<tr><td>mdrfkr</td><td>r"mdrfckr"</td></tr>
<tr><td>echo_ok</td><td>r"\\\\x6F\\\\x6B"</td></tr>
<tr><td>echo_ok_txt</td><td>r"echo ok"</td></tr>
<tr><td>echo_ssh_check</td><td>r"SSH check"</td></tr>
<tr><td>echo_os_check</td><td>r"\becho\b\s+[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}"</td></tr>
<tr><td>uname_a</td><td>r"’uname\s+-a’"</td></tr>
<tr><td>uname_svnrm</td><td>r"’uname\s+-s\s+-v\s+-n\s+-r\s+-m’"</td></tr>
<tr><td>uname_svnr</td><td>r"’uname\s+-s\s+-v\s+-n\s+-r’"</td></tr>
<tr><td>uname_a_nproc</td><td>(̂?=.*nproc)(?=.*\buname\s+-a\b).*</td></tr>
<tr><td>uname_snri_nproc</td><td>(̂?=.*nproc)(?=.*\buname\s+-s\s+-n\s+-r\s+-i\b).*</td></tr>
<tr><td>bbox_5_char_v2</td><td>(̂?=.*\/bin\/busybox\s+([a-zA-Z0-9]{5}\’))(?=.*tftp;\s+wget).*</td></tr>
<tr><td>bbox_scout_cat</td><td>r"/bin/busybox\s+cat\s+/proc/self/exe\s*\|\|\s*cat\s+/proc/self/exe"</td></tr>
<tr><td>bbox_loaderwget</td><td>r"loader\.wget"</td></tr>
<tr><td>bbox_echo_elf</td><td>r"\\\\x45\\\\x4c\\\\x46’"</td></tr>
<tr><td>bbox_unlabelled</td><td>/bin/busybox\s|busybox\s</td></tr>
<tr><td>juicessh</td><td>r"juicessh"</td></tr>
<tr><td>passwd123_daemon</td><td>(̂?=.*Password123)(?=.*daemon).*</td></tr>
<tr><td>pattern_7</td><td>r"cd\s+/tmp\s*;\s*rm\s+-rf\s+/tmp/\*\s*\|\|\s+cd\s+/var/run\s*\|\|\s+cd\s+/mnt\s*\|\|\s+cd\s+/root\s*;\s*rm\s+-rf\s+/root/\*\s*\|\|\s+cd\s+/"</td></tr>
<tr><td>rapperbot</td><td>r"ssh-rsa\s+AAAAB3NzaC1yc2EAAAADAQABA"</td></tr>
<tr><td>root_17_char_pwd</td><td>r"root:[A-Za-z0-9]{15,}\|chpasswd"</td></tr>
<tr><td>pattern_5</td><td>(̂?=.*rm\s+-rf\s+\*;\s*cd\s+/tmp\s*;\s*rm\s+-rf\s+\*)(?=.*xoxox0\\\\nxoxox0).*</td></tr>
<tr><td>curl_maxred</td><td>r"–max-redir"</td></tr>
<tr><td>lenni_0451</td><td>r"lenni0451"</td></tr>
<tr><td>binx86</td><td>(̂?=.*CPU\(s\):)(?=.*bin\.x86_64).*</td></tr>
<tr><td>export_vei</td><td>r"export VEI"</td></tr>
<tr><td>clamav</td><td>r"\bclamav\b"</td></tr>
<tr><td>g*****_echo</td><td>r"\\\\x67\\\\x61\\\\x79\\\\x66\\\\x67\\\\x74"</td></tr>
<tr><td>dget_4</td><td>(̂?=.*wget\s+-4)(?=.*dget\s+-4).*</td></tr>
<tr><td>openssl_passwd</td><td>r"openssl passwd -1 \S{8}"</td></tr>
<tr><td>cloud_print</td><td>r"cloud\s+print"</td></tr>
<tr><td>shell_fp</td><td>r"(?=.*\$\bSHELL\b)(?=.*bs=22)"</td></tr>
<tr><td>root_12_char_capscout</td><td>r"(?=.*root:[A-Za-z0-9]{12})(?=.*awk\s+’{print\s+\$4,\$5,\$6,\$7,\$8,\$9;}’)"</td></tr>
<tr><td>perl_dred_miner</td><td>r"(?=.*perl)(?=.*dred)"</td></tr>
<tr><td>stx_miner</td><td>r"(?=.*stx)(?=.*LC_ALL)"</td></tr>
<tr><td>f******_attack</td><td>r"fuckjewishpeople"</td></tr>
<tr><td>ohshit_attack</td><td>r"ohshit"</td></tr>
<tr><td>onions_attack</td><td>r"onions1337"</td></tr>
<tr><td>sora_attack</td><td>r"sora"</td></tr>
<tr><td>heisen_attack</td><td>r"Heisenberg"</td></tr>
<tr><td>zeus_attack</td><td>r"Zeus"</td></tr>
<tr><td>update_attack</td><td>r"update\.sh"</td></tr>
<tr><td>ak47_scout</td><td>r"(?=.*\\\\x41\\\\x4b\\\\x34\\\\x37)(?=.*writable)"</td></tr>
<tr><td>uname_svnr</td><td>r"(?=.*’uname\s+-s\s+-v\s+-n\s+-r’)(?=.*model\s+name)"</td></tr>
<tr><td>gen_curl_wget</td><td>r"(?=.*curl)(?=.*wget)"</td></tr>
<tr><td>gen_echo</td><td>r"(?=.*echo)"</td></tr>
<tr><td>gen_echo_ftp_wget</td><td>r"(?=.*echo)(?=.*ftp)(?=.*wget)"</td></tr>
<tr><td>gen_echo_wget</td><td>r"(?=.*echo)(?=.*wget)"</td></tr>
<tr><td>gen_ftp</td><td>r"(?=.*ftp)"</td></tr>
<tr><td>gen_wget</td><td>r"(?=.*wget)"</td></tr>
<tr><td>gen_curl_echo_wget</td><td>r"(?=.*curl)(?=.*echo)(?=.*wget)"</td></tr>
<tr><td>gen_echo_ftp</td><td>r"(?=.*echo)(?=.*ftp)"</td></tr>
<tr><td>gen_curl_ftp</td><td>r"(?=.*curl)(?=.*ftp)"</td></tr>
<tr><td>gen_curl_ftp_wget</td><td>r"(?=.*curl)(?=.*ftp)(?=.*wget)"</td></tr>
<tr><td>gen_ftp_wget</td><td>r"(?=.*ftp)(?=.*wget)"</td></tr>
<tr><td>gen_curl_echo_ftp_wget</td><td>r"(?=.*curl)(?=.*echo)(?=.*ftp)(?=.*wget)"</td></tr>
<tr><td>gen_curl</td><td>r"(?=.*curl)"</td></tr>
<tr><td>gen_curl_echo</td><td>r"(?=.*curl)(?=.*echo)"</td></tr>
</table>

Table 1: Regular expressions used to categorise the commands. For reproducibility, slurs used in regular expressions are not redacted here, see Section 5. Reader discretion is advised.

E MALWARE STORAGE LOCATION

Figure 17 presents the distribution of AS types for these malware storage locations over our investigation period.

As shown, the majority of malware downloads originate from Hosting ASes, with sporadic appearances of IPS/NSP and CDN ASes, which aligns with expectations. A somewhat surprising finding is the number of “Other” ASes, particularly the significant spike at the end of 2023. We categorized as “Other” any AS that was governmental, academic, personal networks, corporation or unlabeled.

Upon manual verification of these “Other” ASes, we found that all provide some form of hosting service, whether web-hosting or generic VM-hosting. This reinforces the observation that attackers predominantly rely on Hosting ASes for malware storage.