Fifteen Months in the Life of a Honeyfarm

Munteanu, Cristian; Saidi, Said Jawad; Gasser, Oliver; Smaragdakis, Georgios; Feldmann, Anja

DOI
10.1145/3618257.3624826

Publication date
2023

Document Version
Final published version

Published in
Proceedings of the 2023 ACM on Internet Measurement Conference

Fifteen Months in the Life of a Honeyfarm

Cristian Munteanu
Max Planck Institute for Informatics

Said Jawad Saidi
Max Planck Institute for Informatics

Oliver Gasser
Max Planck Institute for Informatics

Georgios Smaragdakis
Delft University of Technology
Max Planck Institute for Informatics

Anja Feldmann
Max Planck Institute for Informatics

ABSTRACT

Honeypots have been used for decades to detect, monitor, and understand attempts of unauthorized use of information systems. Previous studies focused on characterizing the spread of malware, e.g., Mirai and other attacks, or proposed stealthy and interactive architectures to improve honeypot efficiency.

In this paper, we present insights and benefits gained from collaborating with an operational honeyfarm, i.e., a set of honeypots distributed around the globe with centralized data collection. We analyze data of about 400 million sessions over a 15-month period, gathered from a globally distributed honeyfarm consisting of 221 honeypots deployed in 55 countries. Our analysis unveils stark differences among the activity seen by the honeypots—some are contacted millions of times while others only observe a few thousand sessions. We also analyze the behavior of scouters and intruders of these honeypots. Again, some honeypots report orders of magnitude more interactions with command execution than others. Still, diversity is needed since even if we focus on the honeypots with the highest visibility, they see only a small fraction of the intrusions, including only 5% of the files. Thus, although around 2% of intrusions are visible by most of the honeypots in our honeyfarm, the rest are only visible to a few. We conclude with a discussion of the findings of work.

CCS CONCEPTS

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

KEYWORDS

Honeypot, Intrusion Detection, Network Scanning, Cyber Security.

ACM Reference Format:
Cristian Munteanu, Said Jawad Saidi, Oliver Gasser, Georgios Smaragdakis, and Anja Feldmann. 2023. Fifteen Months in the Life of a Honeyfarm. In *Proceedings of the 2023 ACM Internet Measurement Conference (IMC '23), October 24–26, 2023, Montreal, QC, Canada.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3618257.3624826

1 INTRODUCTION

Honeypots are network-connected systems that are deployed to detect and monitor attempts for unauthorized access, command execution, and control of information systems. Honeypots have been used for decades as a security mechanism [38, 47] as well as to shed light on the security status of networks, discover new threats, and to investigate attack strategies in the wild [26, 34, 35, 44]. Honeypots can either be deployed alongside production systems to inform network administrators about specific attack strategies, or in the wild to detect more attack campaigns. Usually, honeypots record metadata (e.g., client IP address, port numbers) as well as behavioral data (e.g., executed commands, communication with other devices, created files). Honeypots have been very successful in analyzing malware code and practices, e.g., Mirai [1, 18].

Today, a myriad of honeypot software is freely available [2, 9, 13, 39]. There are also commercial companies that run large collections of honeypots (called "honeyfarms"), e.g., GreyNoise [51], NetScout [36]. These companies use the honeypots to gather information that is then used as one of the input for their security products. However, the data gathered by these systems is usually unavailable to researchers and/or the public. Therefore, it remains unclear what these honeyfarms see and if the view of a single honeypot differs substantially from those of the full honeyfarm. Note, getting information about honeypots or honeyfarms is difficult, as most operators–both commercial and research–prefer to keep the system hidden to be able to catch further data.

For this paper, we use the unique opportunity to work with Global Cyber Alliance (GCA) [17], a major non-profit honeyfarm operator. In collaboration, we analyze data from a large honeyfarm operator with 221 honeypots deployed in 55 countries in 64 networks that became operational at the end of November 2021 using IPv4 addresses that had not been used as honeyfarms or darknets before. We investigate data collected by these honeypots over 15-months period, shed light on different views of honeypots, and report on the types of threats that are visible to honeypots, *individually*, as well as the honeyfarm, *collectively*. Since the used honeypot software is a medium interaction honeypot we can distinguish scan, scouting, and intrusion behavior. This gives us a unique view of the unsolicited and unwanted activities happening on the Internet. Our contributions can be summarized as follows:

- We analyze more than 400 million sessions from November 2021 to March 2023, collected at 221 honeypots in the studied honeyfarm. We characterize the sessions as no credential, failed login, no command execution, command execution, and command execution with file download.
- We study the characteristics of around 2.1 million honeypot client IPs from more than 17.7 thousand networks worldwide. We notice that more than 40% of these IPs participate in more than one type of activity (e.g., port scans vs. executing commands), and 20% of all activity in our dataset is observed for more than a week.

IMC '23, October 24–26, 2023, Montreal, QC, Canada

ACM ISBN 979-8-4007-0382-9/23/10.
https://doi.org/10.1145/3618257.3624826

- We report that around half of the client IPs are located in a different geographical region than the targeted honeypot. However, 25% of client IPs that connect and execute commands or download malicious code are in close proximity to the honeypots.
- We analyze more than 64 thousand unique hashes of files generated by clients on the honeypots. We find striking differences in the number of IPs associated with each hash: from tens of thousands to only a handful of IPs. We conclude that some high-profile attacks can be mitigated by blocking a handful of IPs, while others are more difficult to mitigate.
- Some of the popular hashes are visible for a year or more. However, the dominance of hashes varies over time. On average more than 500 unique hashes are visible per day, with the number of first-seen hashes varying between 2% and 60% of all daily hashes.
- Our study shows striking differences between honeypots regarding the number of sessions, IPs, hashes of files generated by commands, and first-seen hashes in the honeypot. Around 20% of the honeypots observe 5× to 30× more unique file hashes than the rest of the honeypots. Yet, even the honeypots with the highest number of observed hashes contribute less than 10% of the overall hashes in the honeyfarm.
- We also report that honeypots that observe the highest number of sessions or client IPs are not the ones that collect the highest number of files hashes generated by intruders generating or modifying files. However, the honeypots that collect the highest number of file hashes are typically the ones that observe the hashes earlier than the rest in our honeyfarm.

2 BACKGROUND

This section provides background information on honeypots, different types of honeypots, and honeyfarms.

In general, a *honeypot* mimics a real system to attract, log, and inspect potentially malicious activities. To facilitate access to attackers, honeypots enable logins without any or with easily guessable credentials. After a successful login, a honeypot tries to imitate a real system, e.g., by providing shell access, allowing to execute commands, or enabling downloads from remote servers. At the same time, the honeypot system logs all activity to be then able to analyze the behavior of potential malware.

Honeypots can be classified into different types using a variety of metrics. One commonly used metric to distinguish honeypots is the possible interactions that they provide. This allows us to classify them into low-interaction, medium-interaction, and high-interaction honeypots [29]. Low-interaction honeypots such as Honeyd [39] or Nepenthes [2] simulate only a very restrictive set of system behaviors. High-interaction honeypots such as The Honeynet Project [49], on the other hand, try to imitate a much broader set of services and interactions, providing also access to real-world operating systems without simulation. Medium-interaction honeypots such as Honeytrap [13] or Cowrie [9] can be seen as somewhere in between these two extremes, i.e., they are more sophisticated than low-interaction honeypots but less sophisticated than high-interaction honeypots.

Honeypots can also be run as a fleet to detect potentially malicious behavior more easily. When running more than one honeypot within the same network, this is referred to as a *honeynet*. Similarly, when multiple honeypots are deployed in different networks, this is referred to as a *honeyfarm*.

In our study, we make use of a newly deployed honeyfarm. The honeyfarm consists of 221 medium-interaction *Cowrie* [9] honeypot instances distributed over 65 networks in 55 countries worldwide.

3 RELATED WORK

Researchers have deployed honeypots to gain insights into attacks on various vulnerabilities in applications, devices, or protocols. These tools are deployed on various scales and network types, such as in clouds [3, 12], on campuses [3, 6, 14], or in eyeball networks [18]. Studies have used *vertical honeypots* exposing particular services and protocols, or *horizontal honeypots* offering multiple services and applications [14]. Various studies have also used logs from real-world operational systems to study unsolicited or malicious requests [20, 41].

Honeypot Detection: If a honeypot is easily detected as such, this can negatively affect its effectiveness since adversaries may avoid honeypots to hide their malicious activities. Multiple studies have investigated these honeypot detection techniques [33, 52, 53], and presented techniques to measure the deployment of popular honeypot systems. In addition, there are also online services [46] that provide estimates on the probability of an IP being a honeypot.

Studying Attackers: Medium and high interaction SSH honeypots can capture many potentially useful information (e.g., executed commands) about adversaries. Hence, recent works apply different clustering [45] and natural language processing (NLP) techniques [5] to group attackers' IPs. In work by Shamsi et al. [45], they set up honeypots for 5 protocols (including SSH) in 3 regions of the AWS cloud for a duration of 20 months. While they covered more protocols, they observed fewer sessions than our honeyfarms, i.e., 5.5 million for SSH. They applied different clustering algorithms and identified features to cluster the attacker IPs. The work by Barron et al. [3] deployed 102 honeypots for 4 months and varied their characteristics, e.g., location and difficulty to break, to study the response by attackers.

Another body of work focuses on the initial phase of SSH connections to gain insights into the attacker's IPs. Ghiëtte et al. [16] deployed 4,500 honeypots for a month, focusing on fingerprinting the software stacks and tools used by adversaries to establish the SSH connection. They identified 49 tools used for SSH compromising attempts.

Wu et al. [56] deployed a honeypot on a /16 IPv4 prefix and reported their analysis of data spanning 1,000 days. However, due to the massive number of IP addresses within that prefix, i.e., 65k, their deployment is a lightweight low-interaction honeypot that does not record the executed commands [6]. Hence, their analysis includes only those attributes of the attacker that are observable upon connection, for example, SSH clients and IP addresses. Our Cowrie-based system is a medium-interaction honeypot deployed in 55 countries. This allows us to observe localized attacks as well as the commands and files generated during an attack session.

Amplification Attacks: Amplification DDoS attacks are one of the popular areas where researchers used honeypots for their studies [34]. Since amplification attacks mostly make use of IP address

spoofing, they are limited to UDP-based protocols such as NTP, CharGen, and DNS [19, 25–27, 50].

IoT Honeypots: In response to the increasing number of IoT devices, researchers have also deployed honeypots that simulate vulnerable IoT devices to study attacks on the IoT ecosystem. For example, Griffoen et al. [18] deployed 7,500 specialized Telnet-based IoT honeypots across eight networks for three weeks, primarily focusing on Mirai botnets [1] and their underlying infrastructure. Another study by Dang et al. [12] deployed 108 honeypots across multiple cloud providers and two ISPs to study fileless attacks (attacks that do not require downloading malware) on Linux-based IoT devices, categorizing the attacks based on the executed commands.

Companies Operating Honeypots: Maintaining and operating a large global fleet of honeypots across different types of networks for extended periods of time can become prohibitively expensive for individual research groups. Therefore, we observe cyber-security-focused companies [40], non-profit organizations [37], or their alliances operate such deployments and occasionally share their insights via threat intelligence tools [23], dashboards [23, 48] or white papers. To the best of our knowledge, our study is the first research work that takes an in-depth look into unsolicited SSH and Telnet connection attempts by leveraging a globally distributed honeyfarm across 15 months that is available to researchers.

Network Telescopes: These network infrastructures monitor traffic reaching Internet address space that is not assigned to hosts but is advertised to the global routing system [32]. Although network telescopes can provide insights about scanners [10], attacks [30, 31], and other Internet phenomena and characteristics [4, 11, 55], they are typically passive. Thus, no connections are established or monitored. In recent years, reactive network telescopes have been proposed to address this issue. In the work of Hiesgen et al. [21], a reactive network telescope "Spoki" responds to asynchronous TCP SYN packets and engages in TCP handshakes initiated in the second phase of two-phase scans. With the collected data, they investigated Mirai attacks based on the SYN packet and malicious behavior based on the payload of the initial TCP packet. However, honeypots can report the intruder's full activity and thus are more powerful in detecting malware, unauthorized access, and command execution. Large networks or content providers also operate network telescopes. Richter et al. [41, 42] have a deep look into the scanning activity on the Internet. Using the logs of a large content delivery network (CDN) they show that about 87% of the incoming connection are scanners.

4 HONEYFARM DATASET

Our work relies on data from an operational honeyfarm that we obtain by establishing a collaboration with Global Cyber Alliance (GCA) [17], a large honeyfarm operator. The data is processed and analyzed in situ, using the provided interface. We anonymize sensitive data to comply with the requirements set forth by the collaboration. One of their projects involves building and operating an SSH/Telnet honeyfarm. In this paper, we leverage 15 months of data from their honeyfarm, which has been operational since November 2021 to March 2023.

The goal of deploying honeypots in this honeyfarm is to have them geographically distributed across different countries as well as in different networks, with a focus on residential networks. The honeyfarm consists of 221 identically configured honeypots in 55 countries and 65 Autonomous Systems (ASes). Each honeypot is realized using a customized version of the Cowrie Honeypot suite [9], a medium interaction SSH and Telnet honeypot that is designed to log possible brute force attacks as well as shell interactions that a possible intruder executes. The selection of the Cowrie honeypot software is driven by its ease of use and because it covers a relevant attack vector for IoT devices: SSH and Telnet. In Figure 1, we show a visualization of the 55 countries where the 221 honeypots are deployed. While most countries host a single honeypot, some, e.g., the US and Singapore, host multiple ones.

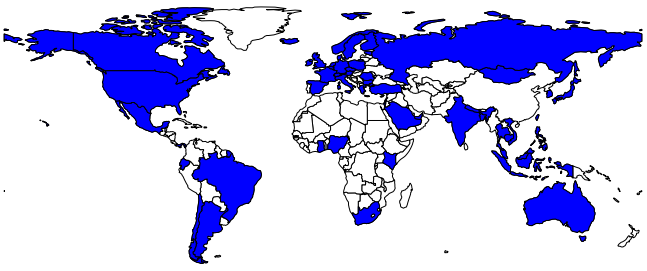


Figure 1: Presence of 221 honeypots in 55 countries.

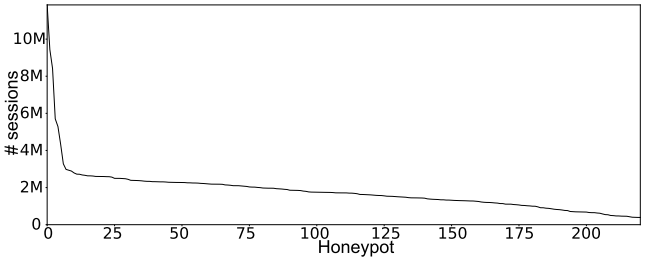


Figure 2: Honeypot activity: Number of sessions per honeypot (sorted by # of sessions).

During our observation period from December 1, 2021, until March 31, 2023 all 221 honeypots are active. Within this time frame, each honeypot reports summaries for each SSH or Telnet session. Each successful TCP connection handshake by a client on either the SSH port 22 or the Telnet port 23 creates a new session in the honeyfarm 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 from the SSH handshake if available.

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

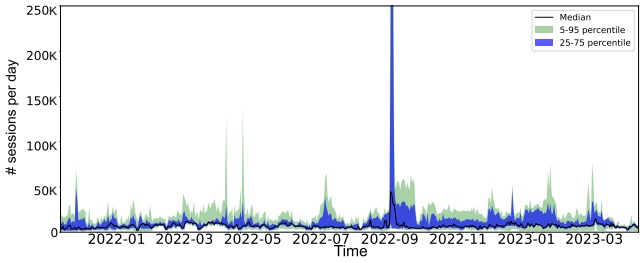


Figure 3: Honeyfarm activity across time for top 5% of honeypots with the most sessions: median, IQR, and 5th/95th percentile ranges.

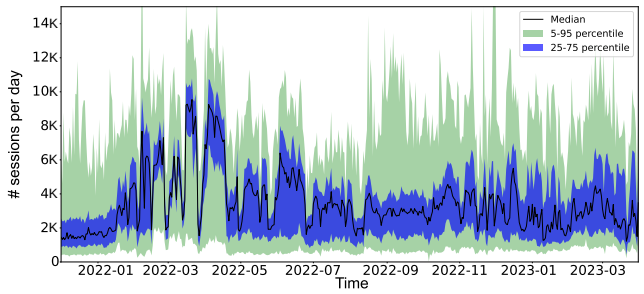


Figure 4: Honeyfarm activity across time for honeypots, showing median, IQR, and 5th/95th percentile ranges for the number of sessions per honeypot.

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 FTP, HTTP, SCP, 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.

Overall, a total number of more than 402 million sessions are recorded throughout the measurement period. A majority of 75.84% of these sessions use SSH, while 24.16% use Telnet to connect. The median daily number of sessions across all honeypots is roughly 1.6 million. We note that the daily activity differs greatly across honeypots, see Figure 2, which shows the number of sessions per honeypot sorted by activity. We find that the top 10 honeypots see 14% of all sessions. Moreover, we observe a knee in the distribution around 11, after which the difference in activity for the remaining honeypots is drastically smaller. Indeed, even the least targeted honeypot still sees more than 360,000 sessions. The 10 most popular honeypots are located in 10 different countries in 7 different ASes. As such, their popularity does not appear to be linked to a specific network or geographical location. Furthermore, it is also not an artifact of a specific time interval, as we see next.

Indeed, the minimum/maximum daily activity of a single honeypot involves 94/1,634,886 sessions. Note that the observed activities change across time according to the activity of scanners and possible attackers. Given the large differences in popularity for the top 5% of honeypots and the remaining ones, we show the median number of daily sessions over time in Figure 3 (black line) as well as the interquartile range (IQR, 25% – 75%, blue line) together with the 5th and 95th percentile range (5% – 95%, light green line) for the top 5% of honeypots only. Figure 4 shows the same analysis for all the honeypots.

We clearly see activity spikes, e.g., on September 5, 2022, when some honeypots see a substantial increase in activity, see Figure 3. Other spikes are visible in May 2022. Still, we see that the median typically follows the 75% and 95% line, as shown in Figure 4. The 25% and the 5% lines are often smoother and do not show as many spikes, with the exception of the 25% line for spring 2022, where it follows the spikes.

To summarize: First, not all honeypots get the same attention from scanners and attackers in terms of the number of sessions. We see that the most targeted honeypot sees more than 30× the number of sessions than the least targeted one. Moreover, the top 10 honeypots account for 14% of all sessions in the dataset. Finally, the daily activity can go above 1.5M sessions per honeypot.

5 ETHICAL CONSIDERATIONS

The honeypot data is processed and analyzed in situ at the collector of the honeyfarm using the provided interface by the major non-profit honeyfarm operator (GCA), with which we have established a research collaboration. No data has been moved outside the premises of the honeyfarm operator.

Sensitive data regarding the honeypots hosting information (as well as our collaborator) is anonymized to avoid collateral damage to all the parties involved. Indeed, providing detailed information about the hosting networks where honeypots are deployed or the exact number of honeypots per country may be utilized to discover the honeypots of the honeyfarm. We do not divulge any information about the configuration of the honeypots (we only report that they all run the very popular Cowrie honeypot software) to prevent malicious actors from detecting or abusing the honeypots.

We anonymize sensitive data information that may be included in files and commands to comply with the requirements set forth by the collaboration. In this study, we do not "name and blame" network operators that host clients participating in the sessions. However, we disclose the information required to improve our cybersecurity understanding, namely the number of IPs and hashes associated with anonymized ASes and each network type.

6 HONEYPOT SESSION CATEGORIES

From our data, we can identify different classes of client honeypot interactions. Namely, we use the following subcategories for our sessions, see Figure 5:

NO_CRED: This category includes all sessions where the client never attempts to log in. Accordingly, the honeypot never sees any credentials. Such sessions can, e.g., be the result of "scans" for open ports without a "login attempt".

FAIL_LOG: Sessions in this category contain login attempts which do *not* succeed. The honeypot only allows login via username and password and not via keys (see Section 4). Hereby, the username must be "root", and the password must be any string except "root". The motivation for this password policy is to check for "root login" attempts rather than "regular user login" attempts.

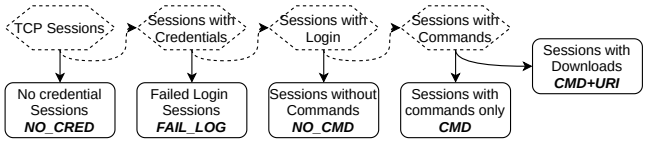


Figure 5: Flow diagram: Session classification.

Protocol	NO_CRED	FAIL_LOG	NO_CMD	CMD	CMD+URI	Total
SSH and Telnet	27.7%	42%	11.6%	18%	0.7%	100%
SSH	21.82%	99.24%	98.30%	93.69%	62.45%	75.83%
Telnet	78.18%	0.76%	1.70%	6.31%	37.55%	24.17%

Table 1: Percentage of total sessions per category (top row), and per protocol (SSH/Telnet only) percentage of sessions in each category (second and third column).

NO_CMD: This category includes sessions with a successful login that never executed any follow-up commands. Note, that there might have been unsuccessful login attempts prior to the successful one within the same session.

CMD: Sessions in this category have a successful login and execute commands (known or unknown ones). However, they do not access any external resources via a URI.

CMD+URI: These sessions have a successful login and executed commands by the client. In addition, the client tries to explicitly access an external resource via a URI.

Based on these categories we define the following different client behaviors:

Scanning: The sessions in the NO_CRED category can best be described as scanning behavior, as no credentials are being sent.

Scouting: The sessions in the FAIL_LOG category can be characterized as scouting behavior, since the clients tried to log in using some credentials.

Intrusion: The sessions in the NO_CMD, CMD, and CMD+URI categories can be described as intrusion behavior, since the clients were able to get access to a shell, with some even executing commands.

Table 1 summarizes the results of the session classification. The top row contains the results for all sessions. The bottom ones are the fractions for SSH and Telnet, respectively. Note that SSH accounts for more than three-quarters of all sessions (see the last column). Overall, more than a quarter of all sessions only check if the ports are open without attempting to log in. This category is dominated by Telnet accounting for more than three fourths. Failed login attempts account for 42% of all sessions. This is fully dominated by SSH with more than 99%. In 11.6% of all sessions, the clients successfully log in, but do not execute any commands—again dominated by SSH. In more than 18.5% of all sessions, the client executes commands, and in more than 0.7%, the client accesses external resources. Here, SSH accounts for more than 60% of all these sessions.

For those sessions involving commands, we note that about one third create or modify files, and 0.5% create or modify multiple files (at least two). The remaining ones do not involve file system access. Table 2 lists the ten most popular successful passwords. Some of them are predictable and are to be expected (e.g., "admin", "1234", or "passw0rd"), while others are very specific. This may indicate that

Password	
admin	1234
3245gs5662d34	dreambox
vertex25ektks123	12345
h3c	1qaz2wsx3edc
passw0rd	GM8182

Table 2: Top 10 most used successful passwords (row-major order).

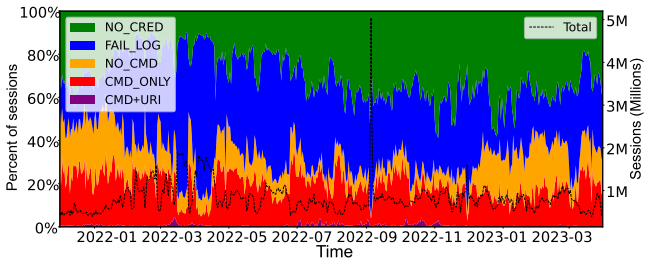


Figure 6: Honeyfarm activity: Across time (black line, right axis) and by category (in percentages, left axis).

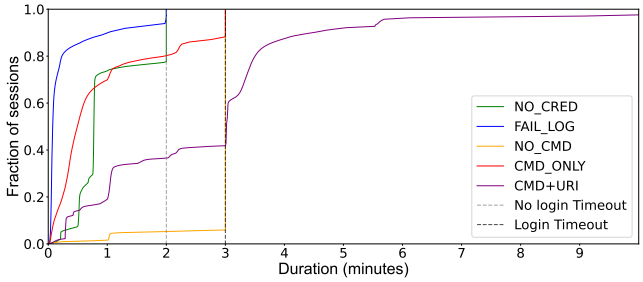
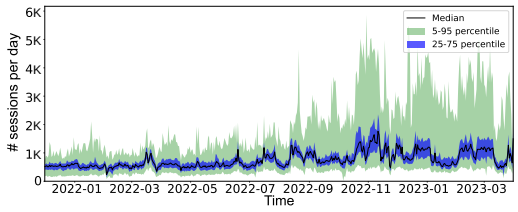


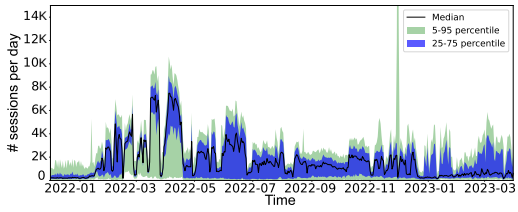
Figure 7: ECDF of honeyfarm session duration by category.

these specific passwords might be part of a larger attack campaign, or have access to a leaked password database. Among the most often used usernames we see the strings "nproc", "admin", and "user". However, as they are not root, the login is unsuccessful in these cases.

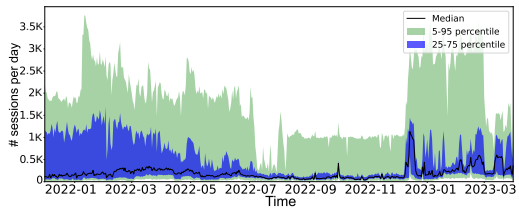
Next, in Figure 6, we show how the fraction of the sessions in each category changes over time using a stacked area plot (left y-axis). In addition, the figure shows the total activity in the number of sessions as a black line (right y-axis). We note that over time the fraction of sessions in the NO_CRED category increases, indicating that the fraction of scanning activity increases. Additionally, at the beginning and end of our observation period, the fraction of sessions with successful logins that never execute any commands (NO_CMD) makes up more than 20%. Upon closer inspection, we find that it is a single prefix that originates most of these sessions, which is mainly active during these time periods. According to RIPEstat [43], this prefix is originated by a Russian datacenter. The fraction of sessions with commands is relatively constant throughout our observation period. It only decreases during the spikes in Spring 2022, December 2023, as well as the large spike on September 5, 2022, when the overall activity is dominated by sessions with failed logins. We also



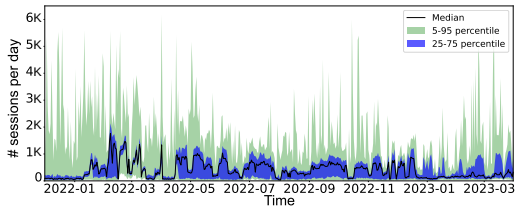
(a) NO_CRED



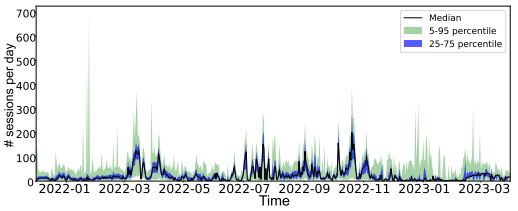
(b) FAIL_LOG



(c) NO_CMD



(d) CMD



(e) CMD+URI

Figure 8: Honeypots activity by category across time: Showing median, IQR, and 5th/95th percentile ranges for the number of sessions per honeypot.

see that the sessions with URIs occur in bursts, most likely related to specific types of attacks.

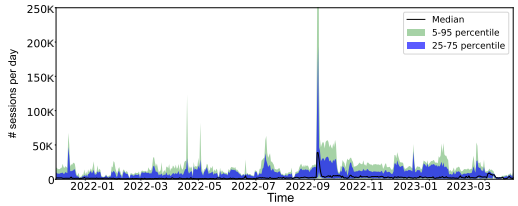
In Figure 7 we show the empirical cumulative distribution function (ECDF) of the session durations by category. In addition, the plot highlights two timeout values: the one for no login (dashed gray line) and the successful login (dashed black line). Overall, we see that the session durations increase with increasing interactions between the client and the honeypot. We confirm that the honeypots terminate many sessions due to timeouts. This is, in particular, the case for more than 90% of all sessions that log in successfully but never execute any command (NO_CMD). Interestingly, most sessions which never attempted to log in (NO_CRED) or with failed login (FAIL_LOG) are terminated after 3 unsuccessful tries (0.3% of all SSH session by the honeypot) or by the client. In addition, we note that a substantial number of sessions with commands (CMD) trigger the login timeout and that some sessions with URIs (CMD+URI) cross the timeout—they last longer than three minutes. This is due to the reset of the timeout period while waiting for the external resource.

So far, we looked at the overall activity of the honeyfarm. Next, we show how the number of sessions in each category varies by honeypot across time in Figure 8. Therefore, we reuse our earlier visualization for all sessions, recall Figure 3 and 4, but this time we have separate plots for each of the categories.

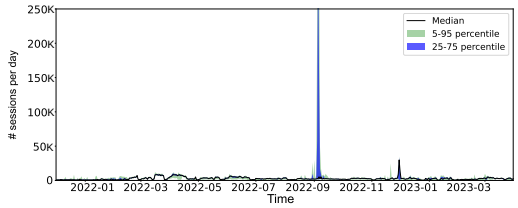
We show the activity for the top 5% honeypots (sorted by session activity). in Figure 9. We notice that for both NO_CRED and FAIL_LOG, three of the honeypots observed a significant amount of sessions on September 9, 2022. Further investigation did not show any difference between the sessions received by these honeypots and the rest. In the case of CMD and CMD+URI, we see that only a single or two honeypots are reporting spikes during the entire period of our study. In Figure 9(c) we notice that the top 5% of the honeypots observe a similar activity. An intense activity from the start of our study period (December 2021) until July 2022, followed by a drop and another rise in the first three months of 2023.

Overall, we see that the activity in each category differs; this is underlined also by the fact that the y-axis scale differs. The most obvious observations are (a) that NO_CRED has a substantial baseline activity for all honeypots at all times, indicating that scanning does not stop; (b) that FAIL_LOG has a similar shape to the overall one, which relates to the point that it contributes a substantial share of sessions; (c) spikes are often due to activity seen by only a small subset of the honeypots, e.g., the spike on November 5, 2022, in FAIL_LOG; for CMD there is a large variation with little baseline activities across all honeypots except during the spring of 2022; (d) there is a large variation in terms of targeted honeypots for NO_CMD sessions, and there are almost no spikes visible in the other categories; and (e) both sessions in the CMD as well as the CMD+URI category are quite spiky, indicating that there are time periods with substantial activity. We also note that towards the end of 2022, a subset of the honeypots are scanned more often, i.e., the variance increases for NO_CRED sessions.

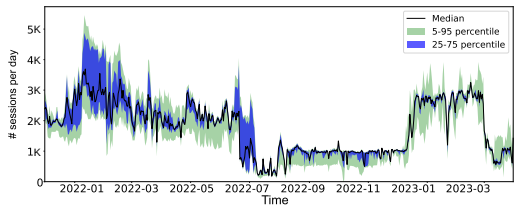
To summarize: By splitting the honeypot sessions into different categories, we notice the first characteristic behaviors. We observe that the NO_CMD sessions tend to be terminated by timeouts while most of the FAIL_LOG and NO_CRED sessions are closed before the one minute mark. Another observation is that the NO_CRED category is more stable over time, while the rest of the categories tend to have multiple spikes across our observation period. The first hints at honeypots keep being scanned at a constant rate, while the latter suggests that the scouting and more specifically, the intrusion categories are more volatile over time. These bursts of activities might be linked to different attack campaigns.



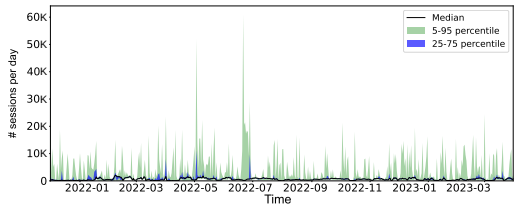
(a) NO_CRED



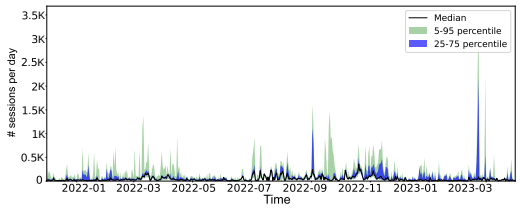
(b) FAIL_LOG



(c) NO_CMD



(d) CMD



(e) CMD+URI

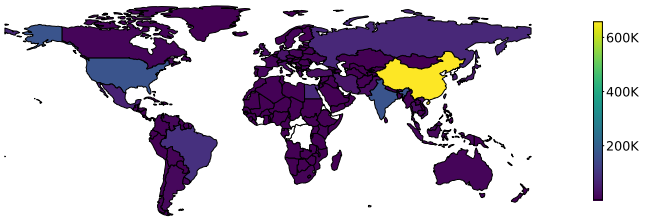
Figure 9: Honeypots activity by category across time for the top 5% honeypots: Showing median, IQR, and 5th/95th percentile ranges for the number of sessions per honeypot.

7 HONEYPOT CLIENTS

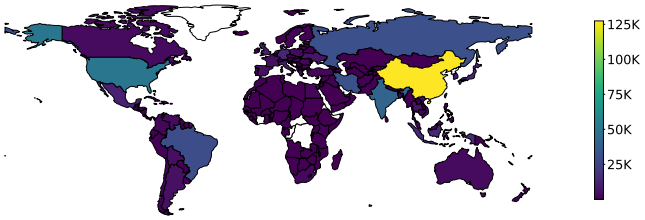
Next, we turn our attention to the characteristics of the clients that establish connections to the honeyfarm.

7.1 Client IPs per country

Overall, the monitored honeypot sessions involve more than 2.1 million unique IPv4 addresses from more than 17.7k ASes during the 15 months observation period. Note, these IPs are unlikely to be spoofed as all sessions involve a successful TCP handshake. To determine the physical location of the clients, we use the commercial API of Maxmind [28] to map the IPs to countries. Figure 10(a) shows the distribution of the clients on the world map using a logarithmic scale. Overall a large fraction of the honeyfarm client IPs originate from China (31%), India (9%), the US (8%), Russia (5%), Brazil (5%), Taiwan (5%), Mexico (3%), and Iran (3%).



(a) All sessions (log scale).



(b) Sessions with commands (CMD + CMD+URI) (log scale).

Figure 10: Honeypot client IPs per country.

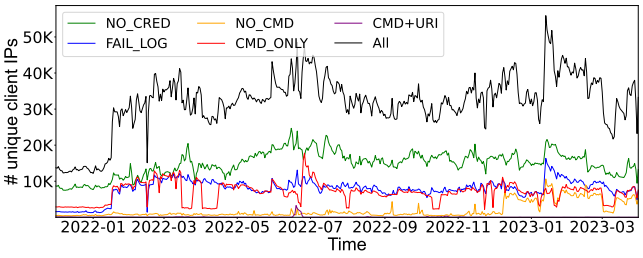


Figure 11: Number of unique IPs per session type.

7.2 Client IPs Over Time

Then, we study how the daily population of client IPs changes over time. Figure 11 shows the number of unique IPv4 addresses observed per day for each category for the duration of our study. The first surprise is the substantial increase in the scanning activity (NO_CRED) after about two months. It seems that it takes scanners some time to discover the honeypots and include them in "regular" scans. Indeed, at the same time, the command execution is also much more prominent (see CMD line). Overall, we notice that the number of client IPs varies substantially per category. The number of client IPs with NO_CRED sessions is higher than those with FAIL_LOG and CMD sessions. The number of client IPs involved in NO_CMD is small, except during the last five months of our study.

The number of client IPs with CMD+URI is very low for most of our study period.

There is also a substantial variation in the number of client IPs per category. For example, the number of unique IPs for NO_CRED varies between 10k and 25k during the 15 months of our study. We also notice that the number of unique IPs for FAIL_LOG and CMD are of the same order of magnitude, around 10k IPs between the beginning of 2021 and the beginning of 2023. However, there is a noticeable change at the end of 2021 from around 3k to 10k and a substantial increase for FAIL_LOG in 2023 from 10k to more than 12k. The daily number of client addresses that are categorized as NO_CMD is relatively low, around 1k IPs, for most of of our study period. However, there is a noticeable increase after December 2022 reaching the levels of CMD at the level of 10k IPs. The number of client IPs that are categorized as CMD+URI sessions is overall small, with a noticeable spike in June 2022 with more than 2,500 IPs.

We note that the number of daily unique IPs is small compared to the number of IPs observed during the full period of our study. This implies that, typically, the same set IPs do not contact the honeyfarm continuously. This indicates that a large number of client IPs can be utilized for scouting as well as other more suspicious interactions, and with low frequency. This makes these IPs more difficult to be detected as potential malicious ones when the window of observation is small.

7.3 Client IPs across countries by category

Next, we revisit the geographic distribution of the clients but this time per category. Overall, we observe similar distributions but with some variations. Figure 10(b) shows the distribution of IPs on the world map for CMD + CMD+URI. For comparison, we show the maps for all categories in the Appendix A. NO_CRED sessions involve 1.7 million unique IPv4 addresses in 14 thousand ASes during our study period. Again, the largest fraction of them originates from the US, China, Taiwan, Russia, and Iran. A large fraction of client IPs are attributed to FAIL_LOG sessions, see Figure 23(b) in the Appendix, are hosted in Asia. Although the US is still the most popular origin, China, Japan, Vietnam, Singapore, and India are at the top of the list. Moreover, the set of these IPs is relatively smaller compared to NO_CRED. We observe 420 thousand unique IPv4 addresses in 11.7 thousand ASes. The number of client IPs that are classified as CMD are similar to those classified as FAIL_LOG. Indeed, we observe 450 thousand unique IPv4 addresses in 10.6 thousand ASes, see Figure 10(b).

Around 222 thousand of these IPs also establish FAIL_LOG sessions with honeypots in the honeyfarm. This is to be expected as compromised hosts are used to establish connections with different credentials. At the top of the list of countries that host IPs that establish CMD sessions are the US, China, Japan, India, and Brazil, see Figure 10(b). The number of IPs involved in the NO_CMD session is smaller compared to CMD. Indeed, we observe only 160 thousand IPs in 8.5 thousand ASes. This can be attributed to the fact that the majority of client IPs that successfully login to any of the honeypots in the honeyfarm execute commands. At the top of the list are Russia, Germany, the US, Vietnam, and Sweden. Finally, only a small number of client IPs are involved in CMD+URI sessions, as such sessions are only a very small fraction of all sessions (around 0.18%). The clients in this category are 16 thousand IPv4 addresses in 1.3 thousand ASes. Among the top countries that host these client IPs are the US, Netherlands, France, Bulgaria, and Romania, see Figure 23(e) in the Appendix. The main take-away is that as the interactions become more involved, the top countries hosting the clients changes.

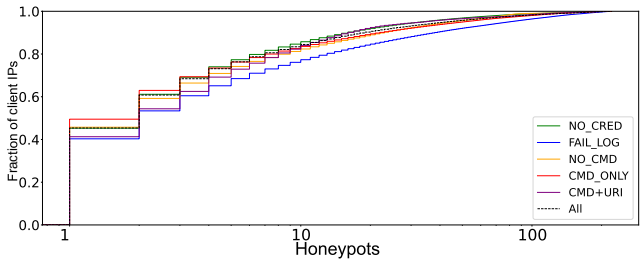


Figure 12: ECDF of number of honeypots contacted per client by category (log scale).

7.4 SSH vs. Telnet Activity

Next, we look at differences per protocol. Overall, recall that SSH is much more common than Telnet. However, for NO_CRED we observe 78% Telnet and only 22% SSH sessions. This is also true when looking at the country granularity. However, some countries differ. For example, for Vietnam, Brazil, and Lithuania, SSH dominates. For other countries, e.g., Italy and Canada, SSH contributes less than 10%. NO_CMD is clearly dominated by SSH except for some special cases, e.g., Switzerland and Bulgaria. Manual cross-checks show that these are due to one spike in activity.

7.5 Client IPs vs. Honeypots

Given our earlier observation regarding the reuse of client IPs, we now investigate the relationship between client IPs and honeypots more closely. In Figure 12 we show an ECDF of the number of honeypots within our honeyfarm that are contacted by each client IP. We acknowledge that dynamic IP address assignment may skew the distribution towards lower numbers.

Overall, we see that more than 40% of all client IPs only contact a single honeypot. Yet, 18% contact more than 10, and 2% contact more than half of the honeypots (> 110). Surprisingly, clients that interact more with the honeypots are likely to contact more of them. The largest exception, though, is the FAIL_LOG category. One possible explanation is that FAIL_LOG sessions can indicate a reconnaissance behavior of an intruder in order to detect more potentially vulnerable hosts.

Next, we ask how long client IPs are interacting with our honeyfarm. Figure 13 shows an ECDF of the number of days that a client IP contacts any of the honeypots in our honeyfarm. Note, that most client IPs are only visible for a single day. Yet, more than 100 client IPs are active almost every day (> 90%) during our study period. We again see that FAIL_LOG sessions are most prominent. On the other hand, IPs that involve CMD+URI are seen for the lowest number of days. Indeed, these are mainly consecutive days. This hints at the possibility that potential attackers are trying out different strategies.

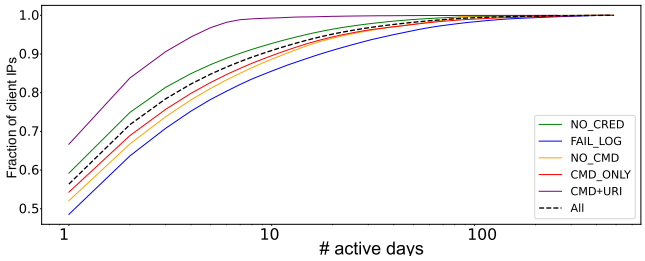


Figure 13: ECDF of number of days that a client IP is observed by category (log scale).

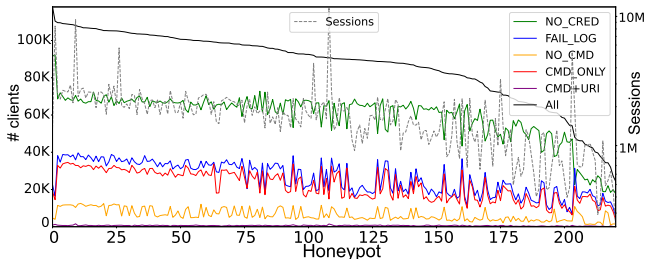


Figure 14: Number of clients per honeypot per category.

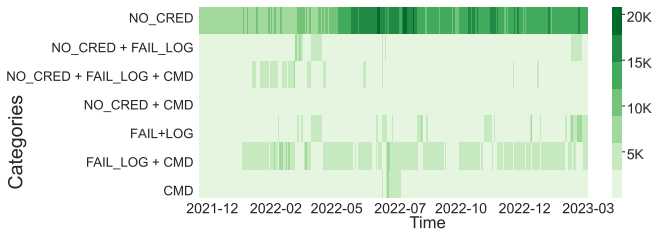
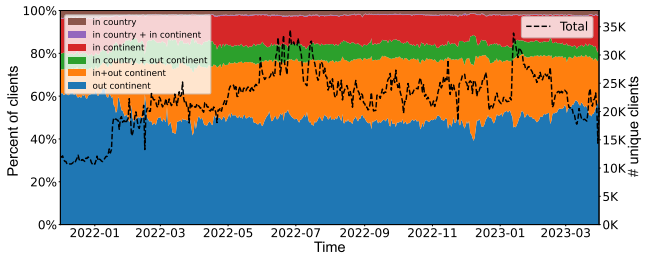


Figure 15: Number of clients per combinations of category across time (per day).

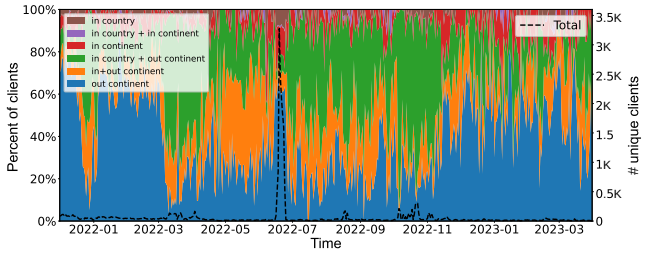
So far, we have taken the viewpoint of a client IP. Now we are switching to the the viewpoint of honeypots and ask how many different clients are contacting each one of the honeypots within our honeyfarm. Figure 14 shows the number of unique client IPs per honeypot sorted by the number of clients. We see that a few honeypots (the top 10 ones) are contacted by many more clients than others. Surprisingly, these are not the same as the ones that see the largest number of sessions, recall Figure 2. For comparison, we add the number of sessions to Figure 14 using the right y-axis using log scale.

We also include the number of client IPs per category within the figure. Note, that these do not add up, as some clients have sessions in multiple categories. Overall, the number of clients involved in scanning is more than twice as many as those involved in more advanced honeypot interactions. The curves for clients with sessions in FAIL_LOG and CMD sessions follow each other closely, whereby the one for FAIL_LOG is just slightly larger. Sessions with NO_CMD and CMD+URI are initiated from just a small fraction of the clients.

To underline that some IPs are indeed active in multiple categories, we check how many client IPs are involved in sessions



(a) All sessions.



(b) CMD+URI.

Figure 16: Regional diversity: client vs. honeypot across time (left y-axis) and number of clients (right y-axis).

from multiple categories on the same day in Figure 15. We pick three of our categories. We investigate NO_CRED as this indicates scanning, FAIL_LOG as this may be the first step of interaction with the honeypot, and CMD as this involves substantial interactions with the honeypot. We find that more than 700k IPs are only involved in scanning (NO_CRED). We notice that scanning is often combined with login attempts as well as command execution, e.g., in Spring 2022. Scanning combined with commands rarely occurs on the same day. However, failed logins together with commands are very common but hardly occur on the same day as failed login only. Commands only complement those that also do failed login, e.g., in Summer 2022.

7.6 Regional Diversity

Next, we take advantage of the geographic diversity of the honeyfarm. This allows us to check if the client and the contacted honeypot are in the same country, on the same continent, or in completely different geographic regions. In Figure 16(a), we show the regional diversity of client/honeypot interactions across time as a fraction of clients. In addition, the plot also shows the number of clients observed per day. Overall, we see that most clients contact honeypots that are not on the same continent. This makes up more than 50% of all interactions every single day. The next most common class involves clients that contact a honeypot on the same continent and on another continent. This indicates that locality is not the primary motivation when clients choose address space for scanning or testing for vulnerabilities. The fraction of clients whose interactions stay within a single country is tiny. Even if we include clients that interact within the country as well as outside, the fraction still remains small. One caveat is that the honeyfarm

has no deployment in China, yet a substantial share of clients (31%) is geolocated in China.

When further digging into the data by looking at the different categories, we note similar behavior for NO_CRED, FAIL_LOG, NO_CMD, as well as CMD, see plots Figure 24 in the Appendix. However, those sessions that involve URIs show more geographic proximity, see Figure16(b). Here, the fraction of sessions out of the continent is substantially smaller, while the fraction in and out of the continent is substantially larger. This indicates that geographic locality may matter more when clients start picking targets for specific tasks.

To summarize: First, the number of unique client IPs increases over time. This can be explained by the fact that it takes a while until the honeypot IPs become well-known. Second, more than half of client IPs contact multiple honeypots. Also, we notice a substantial number of IPs involved in more than one type of session. This sort of behavior makes the differentiation between a "malicious" and a "harmless" actor much more difficult. Third, more than 50% of client IPs are seen active only for one day. In CMD+URI sessions, this value increases to almost 70%. This result shows that IP blocking might not be a good option, yet we will later show that this is not always the case. Finally, intrusion sessions, specifically the CMD+URI sessions, tend to originate closer to the honeypot. This could be linked to the "malicious actor" choosing a closer location to the target from where to initiate the attack.

8 COMMANDS

In this section, we look closely at the commands and associated file hashes resulting from a client's interaction with any of our honeypots. This involves roughly 19% of all sessions.

8.1 Common Commands

To understand common commands executed by honeypot clients, we take the recorded command strings, split them at command separators (";" and "|"), and look at the most popular ones, see Table 3. Among them, we find many typical Unix commands that give information about the system, such as `free`, `uname`, `w` for whois, etc., or access to files with such information, e.g., `cat /proc/cpuinfo`. Another class of popular commands relates to script execution such as `awk` or `shell`, or remote file access such as `wget` or `tftp`. Other commands relate to SSH and its handling of public keys, e.g., `mkdir .ssh`. Indeed, we also find the entry of a trojan horse SSH public key via `echo` into the SSH `authorized_keys` file among the most popular commands. Changing rights, e.g., `chmod 777`, can also be seen among popular commands. Another set of commands relates to changes of credentials, e.g., `chpasswd`. These observations are similar to those seen, e.g., by Koniaris et al. [24] and Dang et al. [12].

Overall, there is a lot of diversity in these commands with extra spaces, quotes, etc., which makes an overall classification challenging. Still, it is clear that typical Unix commands related to information gathering, script execution, file transfer, and credentials are among the most popular ones. In addition, about one-third of the commands involve creating or modifying a file for which the honeypot records a hash of the file content. These hashes can then be used to identify files with the same content across honeypot sessions. In effect, the hashes are signatures of the client honeypot interactions and, as such, can be used to identify specific attack campaigns. Almost all sessions only involve a single file. Only 0.5% of all the sessions involve two, e.g., when they create one and then, later on, change it. A few sessions, namely 282, involve more than 10 file operations, generating more than 10 hashes.

Commands		
grep name	awk	cat /proc/cpuinfo
free -m	crontab -l	w (whois)
uname -a	bash	chpasswd
mkdir .ssh	rm -rf .ssh	chmod -R go= /.ssh
top	echo "ssh-rsa AAA..."».ssh/authorized_keys	
shell	cat /proc/mounts	wget
tftp	history -c	chmod 777 bins.sh

Table 3: Top 20 commands (row-major order).

Hash	# Sessions ↓	# Unique Client IPs	# Days	Tag	# Honeypots
H1	25,688,228	118,924	484	trojan	221
H2	153,672	3	252	malicious	202
H3	110,280	12,698	119	trojan	150
H4	105,102	1,288	20	mirai	203
H5	96,523	1,027	451	mirai	221
H6	87,610	3	33	malicious	74
H7	64,762	1	3	unknown	180
H8	58,662	165	4	mirai	214
H9	57,726	43	220	trojan	173
H10	54,464	488	6	mirai	209
H11	54,262	354	65	trojan	221
H12	52,312	129	6	malicious	215
H13	47,240	1	31	miner	212
H14	40,274	29	9	malicious	210
H15	40,031	4	18	mirai	184
H16	39,688	200	12	miner	208
H17	36,049	3	8	mirai	183
H18	33,569	3	12	unknown	30
H19	33,400	2	4	unknown	184
H20	32,785	105	3	malicious	215

Table 4: Top 20 hashes sorted by the number of sessions.

8.2 Popular Hashes and Attack Campaigns

During the 15 months of our study, we observe 64,004 unique file hashes created by honeypot clients. To understand which files may correspond to known malware, we cross-check all hashes in the VirusTotal malware database [54]. Of the more than 64k hashes, we find information for only less than one thousand hashes. Of these, most are labeled as malicious or are associated with specific attack threats, e.g., Mirai, or attack families, e.g., Trojan.

Given this low coverage, we use additional databases for manual cross-checks for the most popular hashes, namely ClamAV [8], FileScan.IO [15], InQuest [22], CERT.PL MWDB [7], and YOROI YOMI [57]. Table 4 summarizes the results. Among the 20 most popular hashes—by number of sessions—we find 6 hashes related to Mirai, 5 malicious ones, 4 trojan ones, 3 unknown ones, and 2 miners (one for Bitcoin and one for Ethereum). We find that the number of sessions differs substantially. The first trojan is the one that is also among the top commands, recall Table 3. In terms of sessions, it dominates all other commands as it is more than 20

Hash	# Sessions	# Unique Client IPs ↓	# Days	Tag	# Honeypots
H1	25,688,228	118,924	484	trojan	221
H3	110,280	12,698	119	trojan	150
H21	16,670	5,897	9	suspicious	205
H22	4,680	2,213	16	unknown	206
H23	1,803	1,310	63	unknown	126
H4	105,102	1,288	20	mirai	203
H24	2,279	1,144	425	mirai	77
H25	2,250	1,126	424	mirai	77
H26	2,187	1,108	423	mirai	77
H27	1,208	1,067	30	malicious	113
H5	96,523	1,027	451	mirai	221
H28	1,485	752	305	mirai	76
H29	1,503	750	312	mirai	76
H30	1,443	736	305	mirai	76
H31	1,191	704	3	suspicious	185
H32	1,213	610	281	mirai	75
H33	29,227	575	456	mirai	221
H10	54,464	488	6	mirai	209
H34	761	448	301	trojan	118
H35	2,809	416	8	unknown	193

Table 5: Top 20 hashes sorted by number of client IPs.

Hash	# Sessions	# Unique Client IPs	# Days ↓	Tag	# Honeypots
H1	25,688,228	118,924	484	trojan	221
H33	29,227	575	456	mirai	221
H5	96,523	1,027	451	mirai	221
H24	2,279	1,144	425	mirai	77
H25	2,250	1,126	424	mirai	77
H26	2,187	1,108	423	mirai	77
H36	6,213	399	325	mirai	220
H29	1,503	750	312	mirai	76
H28	1,485	752	305	mirai	76
H30	1,443	736	305	mirai	76
H34	761	448	301	trojan	118
H32	1,213	610	281	mirai	75
H37	4,875	27	274	mirai	217
H2	153,672	3	252	unknown	202
H9	57,726	43	220	trojan	173
H38	10,834	4	172	trojan	197
H39	981	19	159	mirai	75
H40	7,532	5	151	unknown	4
H41	8,309	4	145	trojan	193
H42	660	13	145	trojan	63

Table 6: Top 20 hashes sorted by the number of active days.

times as popular as the next one. It also involves many unique client IPs that contact all possible honeypots in our honeyfarm. Moreover, it is active on all days throughout our observation period.

The next popular hash is a malicious one that has rather different characteristics. It only involves 3 client IPs, is active for roughly half of the time period (with some breaks in between), but still contacts almost all honeypots. Others, e.g., H6 and H7, also involve fewer than 5 clients but last much shorter or target only a subset of the honeypots.

Interestingly, there are multiple hashes associated with the Mirai attack, but they have all different characteristics, e.g., H8 lasts for 4 days, H4 lasts 20 days, H5 lasts almost the whole period. H4 and H5 involve more than 1000 client IPs, yet H8 only 165 client IPs. One of the miners is only observed from a single client but with a large number of sessions (for that one client). It is active for a month and contacts almost all honeypots. The second one is active for only 12

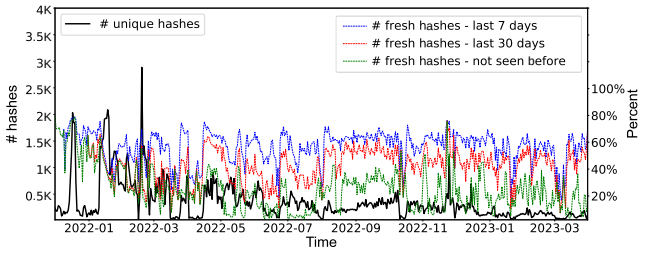


Figure 17: Unique Hash activity: Across time (black line, left axis) and fraction of fresh hashes (right axis). We distinguish between overall fresh hashes—not seen before, and hashes that are fresh within the last 30 or 7 days.

days but involves 200 clients. Overall, 8 of the top 20 hash involve less than 5 client IPs as seen by our honeyfarm, while 12 of them are seen by more than 200 (90%) of the honeypots.

To highlight the diversity, we show the top 20 hashes sorted by the number of unique client IPs in Table 5 and sorted by the number of days in Table 6. Interestingly, Mirai-related hashes are dominant among the long-lasting attacks, and for quite a number of them, they only contact 75–77 of the honeypots. Further investigation shows that the sessions with these hashes always use the same login credentials ("root":"1234") and run a similar set of commands.

Among the takeaways of this analysis are that attack vectors vary largely by the number of sessions, number of unique client IPs, active days, and number of contacted honeypots. As such, a honeyfarm needs to have diversity in terms of geographic coverage, network coverage, etc. This diversity implies that some of the attacks may be easy to block, e.g., if they only involve a small number of client IPs. Others are more difficult to stop, e.g., if they involve a large botnet of clients.

8.3 Campaign Timeline and Freshness

Figure 17 shows the number of unique hashes that are collected per day at the 221 honeypots of the honeyfarm. The number of hashes varies substantially from a few tens up to three thousand per day. Note, the spikes that can occur at any point and are not restricted to the first few months of the honeyfarm operation.

Given that attack campaigns have different durations, we also analyze if the hashes are fresh, i.e., that they were not observed before. To not bias the data by our observation period, we define two additional freshness metrics using a sliding window approach, namely not observed within the last 7 days and the last 30 days.

We find that a substantial fraction of unique hashes observed in a day are fresh. In Figure 17, we show the fresh hashes as a percentage of the unique hashes observed per day with a green dashed line. This percentage varies substantially from 2% up to 60%. There is no apparent correlation between the number of unique hashes and the number of fresh hashes.

When moving to the two additional freshness metrics, we see that the fraction of fresh hashes is increasing up to 60%. With less memory, i.a., as we move from all via 30 days to 7 days, the percentage of fresh hashes increases. Overall, this plot highlights that "new" attacks or variants of attacks that generate fresh hashes

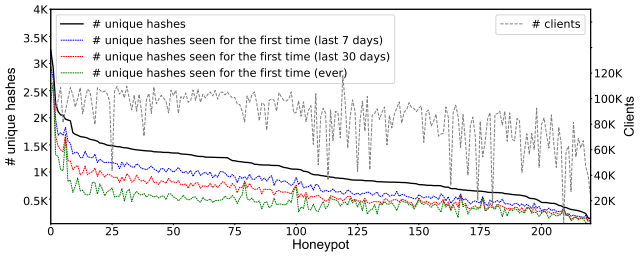


Figure 18: Unique (fresh) hashes: Per honeypot (left axis) and # of client IPs (right axis).

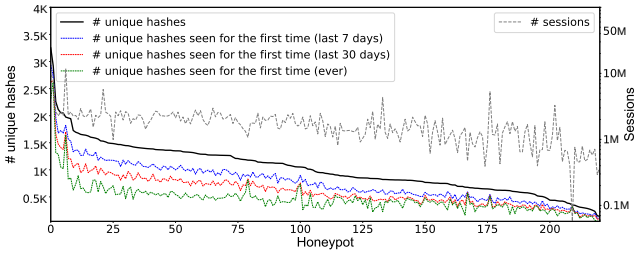


Figure 19: Unique (fresh) hashes: Per honeypot (left axis) and # of sessions (right axis).

are hitting the honeyfarm every day. We also see that some attacks are active for some time, then pause and restart.

8.4 Hashes vs. Honeypots

Figure 18 shows the number of unique hashes recorded by each honeypot sorted by the number of unique hashes. We again observe a large variability, i.e., the top 10 honeypots observe many more hashes than the rest. More concretely, they see a factor of 20 more hashes than the ones in the tail. However, the top honeypot does not see more than 5% of all the hashes, and the top 10 see less than 15% of all hashes. The grey dotted line shows the number of clients for each honeypot. Figure 19 is a similar figure where the grey dotted line shows the number of sessions for each honeypot. Interestingly, however, the honeypots that see the highest number of unique hashes are not necessarily the honeypots that are contacted by the largest number of clients (gray line in Figure 18), nor are the ones with the most sessions (gray line in Figure 19).

Still, when we look at the freshness of the hashes, the top 10 honeypots remain the same, even though the individual ranks change by a few spots. Thus, they still contribute most of the fresh hashes. The same holds for the time-limited freshness metrics with a memory of 7 resp. 30 days. Indeed, the ranking remains almost stable throughout the whole set of honeypots.

When looking at how many hashes are seen by how many honeypots, we find that more than 60% of all the hashes are observed by a single honeypot only. However, more than 6.8% are seen at more than 10 honeypots, and more than 200 are seen by more than half of the honeypots; among them are many of those presented in Table 4. We also check the distribution of the hashes among countries and find a similar long-tail distribution. These numbers indicate that besides those hashes that are observed everywhere, there is a very long tail. Given that security incidents often try to hide in the long

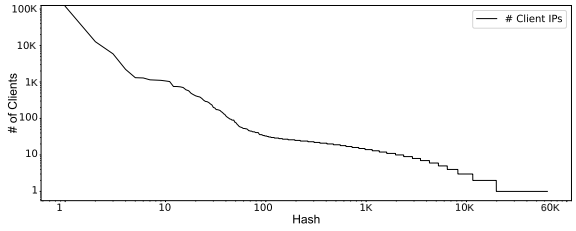


Figure 20: #clients IPs per hash (log log scale).

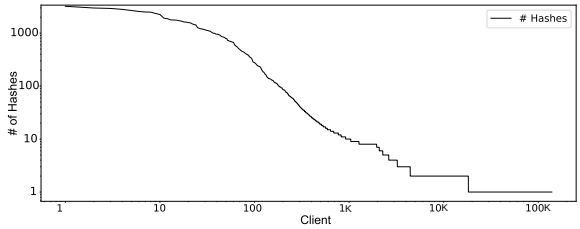


Figure 21: #Hashes per client IPs (log log scale).

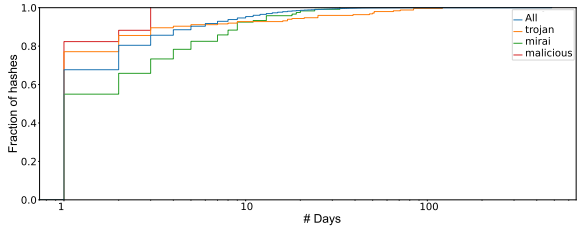


Figure 22: ECDF: length of attack campaigns by attack type.

tail, it is important to have a honeyfarm infrastructure that has sufficient vantage points and is geographically and topologically diverse.

8.5 Hashes vs. Clients

On the one hand, we observe hashes that involve many client IPs. On the other hand, some involve only a single client IP. To better understand this relationship, in Figure 20 we plot for each hash (log x-axis) the number of unique client IPs that involve this hash (log y-axis) sorted by the number of client IPs. We find the typical effects of a long-tailed distribution. Some hashes involve many clients, while others only have a few. The ones with many clients are likely easier to observe but may be harder to block. While the ones in the tail may be harder to observe but easier to block.

We also plot, in Figure 21, for each client IP (log x-axis) the number of unique hashes that involve this client (log y-axis) sorted by the number of hashes. Again we see the effects of the long-tail distribution. Some hashes/campaigns are driven by a small number of IPs others involve large sets of IPs. Such large sets of IPs may point to a sizable botnet.

8.6 Activity of Attack Campaigns

To better understand the activity of each attack campaign, we next investigate how many days we see a specific hash in any of our honeypots. Figure 22 shows the corresponding empirical cumulative distribution function (ECDF). Hereby, we look at all hashes and the subclasses we derived using VirusTotal. The latter distinguishes

hashes associated with the Mirai attack, hashes tagged as trojan, and hashes tagged as malicious. The main takeaway is that (a) most hashes are only active for a single day, (b) hashes tagged as trojan tend to be active on more days compared to others, and (c) hashes tagged as Mirai are typically active for less than 30 days.

9 DISCUSSION

On Multi-role Client IPs. Our analysis shows that 40% of all client IPs are involved in more than one type of connections. Thus, these devices, either compromised or purposefully provisioned as attack systems, serve different roles, ranging from scanning (checking for open ports) to scouting (trying login credentials), as well as intrusion and interaction (logging in and executing commands) with potentially vulnerable devices on the Internet. This shows that client IPs that are performing scanning, over time may perform security-threatening tasks. We were able to characterize such behavior as we utilize a highly distributed honeyfarm and collect connection data over a long period (15 months). Such observations highlight the value of a distributed honeyfarm and the need to collect data at honeypots over a long duration.

The Birth of a Honeyfarm. The analysis of our newly launched honeyfarm deployed on 221 hosts (whose addresses have never been used as honeypots before) in 55 countries and 64 networks shows, that from day one to the end of our study (15 months later) the level of activity remains overall similar. Interestingly, the levels of scanning and scouting are almost similar to the levels of intrusion and interaction with the honeypots at the beginning of our study. It takes more than a month until the level of scouting increases and more than 6 months for scanning. After a couple of months in the honeyfarm's lifetime, scanning and scouting activity combined surpasses the intrusion category. At the end of the 15 months of operation of the studied honeyfarm, we did not observe a noticeable drop in scouting or interaction activity. Thus, we conclude that most of the attackers did not bother detecting or blacklisting our honeypot IPs.

Federated Honeyfarms. Our study clearly shows the benefits of operating a honeyfarm over individual honeypots. However, even the top honeypots only observe a small fraction of the unique set of hashes observed in the complete honeyfarm. We believe that the research and security community should collaborate to share the data and intelligence collected by honeyfarms operated and deployed by independent organizations. We expect this will substantially improve the visibility of activities such as scouting, intrusion, and interaction but also has the potential to identify such activity earlier than independent honeyfarms can achieve today.

Honeyfarms and Security Reality. The analysis of the data collected in the honeyfarm also shows that the majority of the honeypots observe high-profile attack campaigns. Many of these campaigns last for weeks, and many of the top 20 campaigns in terms of activity in our honeyfarms even last for months, with some being visible throughout the entire duration of our study, i.e., fifteen months. Often, these campaigns are well orchestrated, and we see the same attack coming from numerous IPs.

However, this is not always the case. We also observe large campaigns which are active for more that half a year, that originate from only a handful of IPs. It is frustrating to see that no action is taken to block IPs participating in such campaigns. Unfortunately, our analysis shows strong indications that network and cloud providers are not well informed or do not have the appropriate filters in place to block client IPs that participate in scouting, intrusion, or interaction with potentially vulnerable hosts. It is even more concerning that long-lasting campaigns observed by a substantial fraction of the honeypot population in our honeyfarm utilize only a handful of IPv4 addresses for their campaign, and still no blocking or take-down of these IPs takes place for months. Thus, although honeyfarms are proven to be very effective in detecting attack campaigns, this is only a part of the chain of network defenses that alone can not be a panacea.

10 CONCLUSION

By analyzing data from a newly deployed and operational honeyfarm consisting of 221 honeypots deployed in 55 countries and 65 networks for 15 months, we shed light on the unwanted and unsolicited activity in the Internet. We performed our analysis per honeypot, i.e., *individually*, as well as per honeyfarm, i.e., *collectively*. Our analysis shows striking differences across identical honeypots operated worldwide.

Hereby, we taxonomized the different connections established with honeypots to illuminate different scanning, scouting, and intrusion behaviors. Depending on the metric, the top honeypots may receive more than 30× more sessions or sessions from 20× more client IPs than honeypots in the tail. Note, which honeypots are the top ones differs substantially by metric. As such, contrary to our intuition, those with the largest number of hashes do not have the most sessions or client IPs. However, we reported that the set of honeypots that observed the highest number of hashes is likely to observe new hashes earlier. These insights can be used to inform new installations of honeypots within honeyfarms depending on the objectives of the honeyfarm, e.g., early detection of new hashes vs. high visibility of scanning activity. We have also observed that the intruders generate many different files. However, none of the honeypots observed more than 5% of all seen unique files. To capture the tail, which are likely the more interesting intrusions, one has to have scale and diversity in the honeyfarm deployment.

Our analysis showed that many of the attacks, based on the hashes involved, are visible for an extended period so that they could in principle be easily detectable. Nevertheless attacks varied and new ones appeared on a daily basis. Moreover, some attacks always targeted the same subset of honeypots with just a few client IPs, while others used many client IPs likely from major botnets and contacted almost all honeypots. The former are in principle easy to take down, the latter are more difficult but they may be useful to track down botnets. Thus, our study provides insights on which attack infrastructures are easy to neutralize and which not within a relatively small observation window.

As part of our ongoing research we continue analyzing the logs and files of the honeyfarm and plan to investigate changes in the attack behavior and practices. Finally, we also plan to coordinate with the honeyfarm operator with the aim to jointly notify networks participating in connections to the honeyfarm.

ACKNOWLEDGMENTS

We would like to thank Global Cyber Alliance for sharing the data with us as well as our shepherd Paul Pearce and the anonymous reviewers for their valuable comments. This work was supported in part by the European Research Council (ERC) under Starting Grant ResolutioNet (ERC-StG-679158).

REFERENCES

[1] 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*.

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

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

[4] Karyn Benson, Alberto Dainotti, kc claffy, Alex C. Snoeren, and Michael Kallitsis. 2015. Leveraging Internet Background Radiation for Opportunistic Network Analysis. In *ACM IMC*.

[5] 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*.

[6] Phuong M. Cao, Yuming Wu, Subho S. Banerjee, Justin Azoff, Alex Withers, Zbigniew T. Kalbarczyk, and Ravishankar K. Iyer. 2019. CAUDIT: Continuous Auditing of SSH Servers To Mitigate Brute-Force Attacks. In *NSDI*.

[7] Cert.PL. 2023. Cert.PL. https://mwdb.cert.pl/. (2023).

[8] ClamAV. 2023. ClamAV. https://www.clamav.net/. (2023).

[9] Cowrie. 2019. Cowrie on GitHub. https://github.com/cowrie/cowrie. (2019).

[10] Alberto Dainotti, Alistair King, kc Claffy, Ferdinando Papale, and Antonio Pescapè. 2012. Analysis of a "/0" Stealth Scan from a Botnet. In *ACM IMC*.

[11] Alberto Dainotti, Claudio Squarcella, Emile Aben, Kimberly C. Claffy, Marco Chiesa, Michele Russo, and Antonio Pescapé. 2011. Analysis of Country-wide Internet Outages Caused by Censorship. In *ACM IMC*.

[12] Fan Dang, Zhenhua Li, Yunhao Liu, Ennan Zhai, Qi Alfred Chen, Tianyin Xu, Yan Chen, and Jingyu Yang. 2019. Understanding Fileless Attacks on Linux-Based IoT Devices with HoneyCloud. In *ACM MobiSys*.

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

[14] Thomas Favale, Danilo Giordano, Idilio Drago, and Marco Mellia. 2022. What Scanners do at L7? Exploring Horizontal Honeypots for Security Monitoring. In *IEEE European Symposium on Security and Privacy Workshops*.

[15] FileScan.IO. 2023. FileScan.IO. https://www.filescan.io/. (2023).

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

[17] Global Cyber Alliance. 2023. GCA AIDE – Automated IoT Defense Ecosystem. https://www.globalcyberalliance.org/. (2023).

[18] Harm Griffioen and Christian Doerr. 2020. Examining Mirai's Battle over the Internet of Things. In *ACM CCS*.

[19] Harm Griffioen, Kris Oosthoek, Paul van der Knaap, and Christian Doerr. 2021. Scan, Test, Execute: Adversarial Tactics in Amplification DDoS Attacks. In *ACM CCS*.

[20] Hwanjo Heo and Seungwon Shin. 2018. Who is Knocking on the Telnet Port: A Large-Scale Empirical Study of Network Scanning. In *ACM ASIACCS*.

[21] 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*.

[22] InQuest. 2023. InQuest. https://inquest.net/. (2023).

[23] SANS Internet Storm Center. 2023. DShield Honeypot. DShield Honeypot, https://isc.sans.edu/tools/honeypot/. (2023).

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

[25] Lukas Krämer, Johannes Krupp, Daisuke Makita, Tomomi Nishizoe, Takashi Koide, Katsunari Yoshioka, and Christian Rossow. 2015. Amppot: Monitoring and defending against amplification ddos attacks. In *RAID*.

[26] Johannes Krupp, Michael Backes, and Christian Rossow. 2016. Identifying the Scan and Attack Infrastructures Behind Amplification DDoS Attacks. In *ACM CCS*.

[27] Marc Kührer, Thomas Hupperich, Jonas Bushart, Christian Rossow, and Thorsten Holz. 2015. Going Wild: Large-Scale Classification of Open DNS Resolvers. In *ACM IMC*.

[28] MaxMind. 2023. MaxMind. https://www.maxmind.com/. (2023).

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

[30] David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford-Chen, and Nicholas Weaver. 2003. Inside the Slammer Worm. *IEEE Security and Privacy* 1, 4 (2003).

[31] David Moore, Colleen Shannon, Douglas J. Brown, Geoffrey M. Voelker, and Stefan Savage. 2001. Inferring Internet Denial-of-Service Activity. In *USENIX Security Symposium*.

[32] David Moore, Colleen Shannon, Geoffrey M Voelker, and Stefan Savage. 2004. *Network Telescopes: Technical Report*. Technical Report. Cooperative Association for Internet Data Analysis (CAIDA).

[33] Shun Morishita, Takuya Hoizumi, Wataru Ueno, Rui Tanabe, Carlos Gañán, Michel J.G. van Eeten, Katsunari Yoshioka, and Tsutomu Matsumoto. 2019. Detect Me If You... Oh Wait. An Internet-Wide View of Self-Revealing Honeypots. In *IFIP/IEEE Symposium on Integrated Network and Service Management*.

[34] 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*.

[35] 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

[36] NetScout ASERT Team. 2018. Dipping Into The Honeypot. https://www.netscout.com/blog/asert/dipping-honeypot. (Oct 2018).

[37] Honeynet Project. 2023. The Honeynet Project. https://www.honeynet.org/about/. (2023).

[38] Niels Provos. 2004. A Virtual Honeypot Framework. In *USENIX Security Symposium*.

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

[40] Rapid7. 2023. Project Heisenberg. Rapid7, https://www.rapid7.com/research/project-heisenberg/. (2023).

[41] Philipp Richter and Arthur Berger. 2019. Scanning the Scanners: Sensing the Internet from a Massively Distributed Network Telescope. In *ACM IMC*.

[42] Philipp Richter, Oliver Gasser, and Arthur Berger. 2022. Illuminating Large-Scale IPv6 Scanning in the Internet. In *ACM IMC*.

[43] RIPE. 2023. RIPE Stat. https://stat.ripe.net/. (2023).

[44] Christian Rossow. 2014. Amplification Hell: Revisiting Network Protocols for DDoS Abuse. *NDSS* (2014).

[45] 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*.

[46] Shodan. 2023. Honeypot Or Not? Shodan, ttps://honeyscore.shodan.io. (2023).

[47] Lance Spitzner. 2003. The Honeynet Project: trapping the hackers. *IEEE Security & Privacy* 1, 2 (2003), 15–23. https://doi.org/10.1109/MSECP.2003.1193207

[48] T-Systems. 2023. The T-Sec Radar shows cyber attacks happening worldwide on our and our partners' honeypot infrastructure. . T-Sec Radar, https://www.sicherheitstacho.eu/start/main. (2023).

[49] The Honeynet Project. 2023. The Honeynet Project. https://www.honeynet.org/. (2023).

[50] Daniel R. Thomas, Richard Clayton, and Alastair R. Beresford. 2017. 1000 days of UDP amplification DDoS attacks. In *Symposium on Electronic Crime Research (eCrime)*.

[51] Justin Varner. 2022. Contextualize honeypot alerts automatically with GreyNoise, runZero, Thinkst Canary, and Tines. https://www.runzero.com/blog/contextualize-honeypot-alerts/. (Oct 2022).

[52] Alexander Vetterl and Richard Clayton. 2018. Bitter Harvest: Systematically Fingerprinting Low- and Medium-interaction Honeypots at Internet Scale. In *USENIX Workshop on Offensive Technologies*.

[53] Alexander Vetterl, Richard Clayton, and Ian Walden. 2019. Counting Outdated Honeypots: Legal and Useful. In *IEEE Security and Privacy Workshops*.

[54] VirusTotal. 2023. VirusTotal. https://www.virustotal.com/. (2023).

[55] Daniel Wagner, Sahil Ashish Ranadive, Harm Griffioen, Michalis Kallitsis, Alberto Dainotti, Georgios Smaragdakis, and Anja Feldmann. 2023. How to Operate a Meta-Telescope in your Spare Time. In *ACM IMC*.

[56] Yuming Wu, Phuong M Cao, Alexander Withers, Zbigniew T Kalbarczyk, and Ravishankar K Iyer. 2020. Mining Threat Intelligence from Billion-scale SSH Brute-Force Attacks. (2020).

[57] YOROI YOMI. 2023. YOROI YOMI. https://yomi.yoroi.company/. (2023).

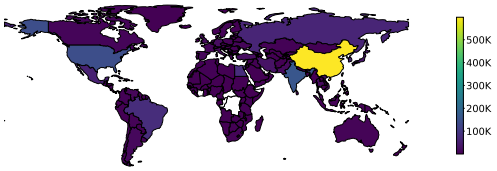
APPENDIX

A HONEYPOT CLIENTS

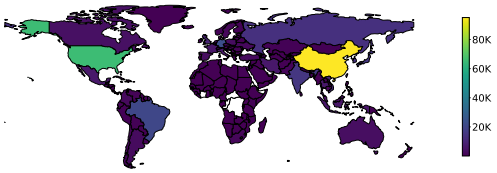
In this section, we present additional results about the geographical distribution of honeypot clients per category. Figure 23 provides an overview of clients distribution per category. Our analysis concludes that there is a similar distribution of the clients in NO_CRED

and CMD, with the majority of clients originating in China. One slight difference is noticed in the activity of Russia, Saudi Arabia, and Brazil, with a larger percentage of clients in CMD. The FAIL_LOG also follows the above distribution with one distinct difference—clients activity in the US. In Figure 23(d), we notice the domination of the China and US clients in the CMD category. Figure 23(e) shows that in the CMD+URI category, the US has the most clients while a significant amount of countries in Africa do not have any clients involved.

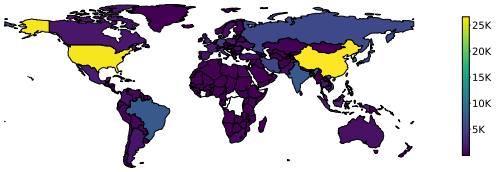
Figure 24 shows the regional diversity of clients vs. honeypots by category. In almost all categories, except CMD+URI, we notice that a significant amount of sessions are established between a honeypot and a client located on a different continent. In the case of CMD+URI, as we note in Figure 24(e), we see a significantly higher number of clients located in the same region as the honeypot.



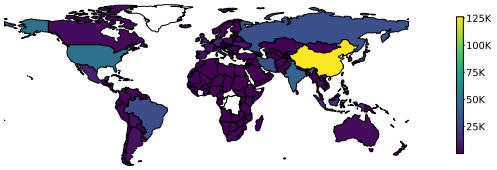
(a) NO_CRED.



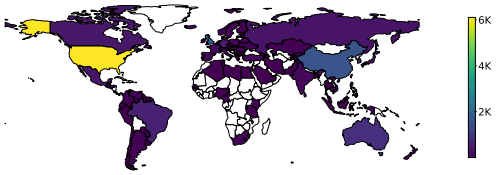
(b) FAIL_LOG.



(c) NO_CMD.

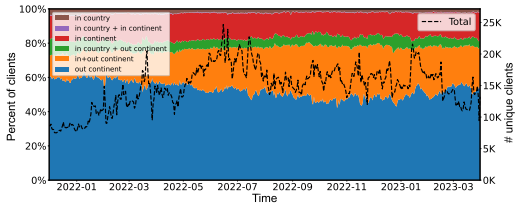


(d) CMD.

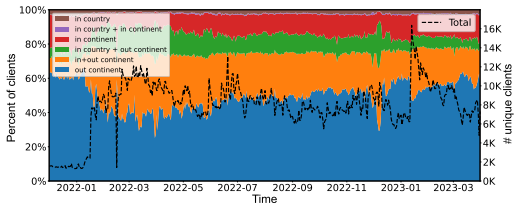


(e) CMD+URI.

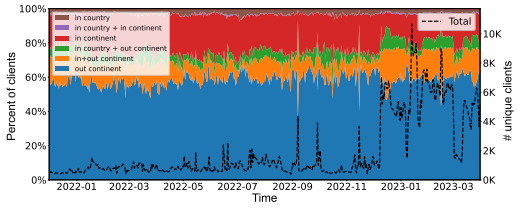
Figure 23: Honeypot client IPs that establish sessions per country (log scale).



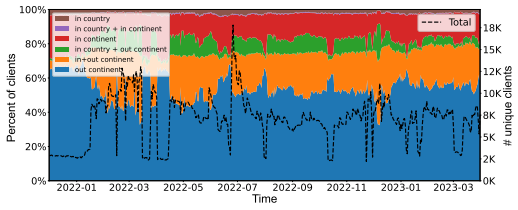
(a) NO_CRED.



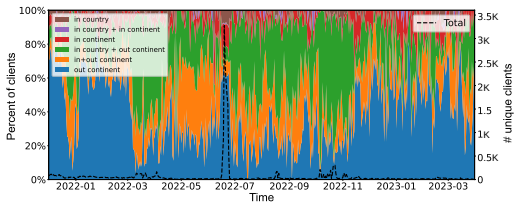
(b) FAIL_LOG.



(c) NO_CMD.



(d) CMD.



(e) CMD+URI.

Figure 24: Regional diversity: client vs. honeypot by category.