UNIVERSITÀ DI PISA

DIPARTIMENTO DI INFORMATICA

Corso di Laurea Triennale in Informatica

Decoding the Cyber Threat Landscape: A Honeypot Data Analysis Across Cloud Providers

Relatori:

Fabrizio Baiardi
Emanuele Briganti
Luca Deri

Candidati:

Filippo Boni
Giovanni Braccini

ANNO ACCADEMICO 2022/2023

Contents

1 Introduction

The widespread use of cloud computing technologies has changed the way modern industries operate, but it has also made them more vulnerable to complex cyber threats that can seriously harm organizations; it's crucial to understand how these attacks happen and how to defend against them.

In this thesis, we delve into the details of cyber threats on two industry-leading cloud providers, Azure and AWS. The main goal is to gain a better understanding of the security measures that exist and how effective they are in preventing threats. This serves two purposes: First, it helps identify which provider offers the most reliable defense against malicious entities, which can inform decisions when deploying cloud services. Secondly, it helps assess whether investing in honeypots is a viable way to understand the attackers' techniques and enhance server security. The deployment of honeypot software in a cloud environment is an already-explored concept. Past research [1] investigated the link between the popularity of cloud providers and the number of attacks. This current study, however, pivots from popularity to a more security-centric focus.

It was discovered that there is a significant contrast in the types of attacks between Azure and AWS. Azure encountered more SSH attacks, whereas AWS experienced fewer SMB attacks. These variations indicate different security obstacles for each platform. Moreover, the study highlights patterns in the timing and location of attacks. The highest number of attacks occurred during nighttime and on Mondays, and the primary sources of attacks were the United States, China, Singapore, and Hong Kong. It was noted that most attacks originated from Autonomous Systems such as Digital Ocean, Google, and Tencent, which implies that attackers utilize cloud services for their operations.

Additionally, the analysis reveals that most attacks are carried out automatically through scripts, often occurring precisely on the hour. A distinct categorization of attackers was made based on their interactions with honeypots, services, and providers, with a majority specializing in attacking one or more honeypots within a single provider and focusing on one service. However, a small yet highly dangerous group of attackers has orchestrated coordinated attacks across various cloud providers. The analysis of malware attacks has revealed that the most common types specifically target IoT devices and originate from the United States and China. This study underscores the importance of tailored security measures and a multilayered approach to cyber defense, emphasizing the value of honeypots as early warning systems in detecting and analyzing sophisticated, targeted attacks.
Due to limited resources, our study employs just two honeypots across distinct cloud providers, which prevents us from drawing certain conclusions in some analyses due to the restricted dataset. It's important to highlight that these limitations were established in advance and do not undermine the integrity of the thesis's findings.

The study begins by proposing the following hypotheses:

H1: No notable temporal dependency exists among attacks.

H2: No apparent spatial dependency exists among attacks.

H3: No dependency from the attacked protocol exists.

H4: Most of the attacking nodes run a Linux-based OS.

H5: Attackers can be segregated based on their behavior.

H6: All geographical regions contribute equally to the origins of malware activities.

H7: A significant correlation exists between the cloud provider's hosting choice and the observed activity.

H8: There is a significant volume of attacks that target IoT devices.

According to the data analysis of the data we collected, hypotheses H4, H5, H7, and H8 were accepted, while the remaining hypotheses were rejected.

The hypotheses H1-H8, which have been proposed, lay the foundation for addressing the study objective of this thesis, outlined on the following page. For a more precise understanding and a finer granularity in the response, we will define several subtypes of these hypotheses in the next chapter. This will include a detailed explanation of their meanings and the rationale behind their selection.

The thesis utilizes eight hypotheses to methodically examine cyber-attack patterns, each hypothesis targeting a specific facet of security concerns. By aligning with the twelve study objectives, these hypotheses pave a clear path for a comprehensive investigation. This systematic approach allows for an in-depth exploration of the security landscape, considering temporal, spatial, and behavioral aspects, and serves as a foundational structure for the research.

The research objectives are:

1. **Time and Day Impact on Attack Patterns [H1]:** Focusing on the temporal aspects of cyber attacks, this objective examines how time and day influence attack patterns.

2. **Geographical and Spatial Attack Patterns [H2]:** This aim delves into the spatial distribution of attacks, identifying the primary sources and implications.

3. **Protocol Dependency [H3]:** This concise goal examines the relationship between attacked protocols and attack patterns, seeking to uncover dependencies within specific protocols highlighting possible coordinated attacks.

4. **Linux's Predominant Role in Attacks [H4]:** Analyzing the role of Linux-based systems in cyber-attacks, this objective highlights the relationship between the OS type and attack landscape.

5. **Attackers' Skill Levels and Resources [H5]:** This goal focuses on the skills and resources utilized by attackers, with an emphasis on differentiating between automated and manual attacks.

6. **Behavior and Profiling of Attackers [H5]:** Understanding the behaviors, strategies, and specialization of attackers are at the core of this objective.

7. **Tor Nodes Relevance [H5]:** A brief objective focusing on the relevance of Tor nodes in cyberattacks, analyzing how attackers may utilize Tor nodes.

8. **Malware Origins Assessment [H6]:** Focused on identifying the origins, types, and purposes of various malware, this goal contributes to a broader understanding of the malware landscape.

9. **Coordinated Attacks [H7]:** Investigating the presence of organized and collaborative attacks across various cloud providers, this objective examines the complexity of multi-faceted cyber threats.

10. **Comparison Between AWS and Azure [H7]:** This objective contrasts the security architectures of AWS and Azure, identifying unique challenges and strengths.

11. **Understanding IoT-Specific Malware [H8]:** By exploring malware that specifically targets IoT devices, this objective emphasizes the need for robust IoT security.

1.1 Chapters Overview

- In Chapter 2, the study's background is established by describing the definition of a honeypot and the concept of cloud computing. A detailed explanation of the methodology used to deploy honeypots and collect data is provided.

- Chapter 3 examines when cyber-attacks happen. It looks at how an attacker's time zone might influence the timing. Findings from this study are then shared. Autocorrelation analysis, a method to find patterns, is used to see if the timing of one attack is linked to others. Finally, it wraps up with the results of this time-based analysis, revealing any patterns in the timing of attacks. Essentially, it's all about understanding when and why cyber attacks occur.

- Chapter 4 discusses how to figure out where cyber attacks come from. The first part goes deeper into the origins of these threats. Then, the chapter talks about making the data more understandable by considering the population and area of the places the attacks come from. The chapter then explores the AS from which most attacks come.

- Chapter 5 examines cyber-attacks based on specific protocols. Specifically, it looks at which services get attacked most often and compares this between different cloud providers identifying the most targeted services and comparing security across different cloud platforms.

- Chapter 6 focuses on creating profiles for attackers by identifying their skill levels, strategies, and the operating systems they use. It also investigates the role of TOR nodes, which can hide an attacker's location and identity.

- Chapter 7 explores the origin and types of malware. It looks at how malware is uploaded and downloaded and where it's hosted.

- Chapter 8 compares how different cloud providers experience attacks. It examines correlations between various factors, like the type of honeypots used, the cloud provider, and the metrics observed.

- Chapter 9 examines attacks specifically aimed at IoT devices. It delves into a particular type of malware called Mirai, known for its focus on IoT. The chapter explores the impact of Mirai attacks and where this malware appears in their data.

- Chapter 10 discusses the usefulness of a honeypot for corporations in protecting their production servers. It serves as both an early detection system and a decoy.

- Chapter 11 summarizes all the findings based on the collected data.

2 Background

2.1 Honeypot: definition and general use

A honeypot is a computer security mechanism designed to identify, divert, or counter unauthorized attempts to access information systems. Typically, a honeypot appears to be a genuine part of a network or website containing valuable information or resources that would attract potential attackers. However, an isolated and monitored system blocks or analyzes the attackers' actions.

Honeypots can be classified according to the low, medium, or high level of interaction they offer to the attacker. They are commonly categorized as low, medium, or high interaction. A low-interaction honeypot provides limited engagement, typically mimicking specific services or protocols. A medium-interaction honeypot offers a wider range of services but still doesn't provide full system access. Conversely, a high-interaction honeypot allows full system access, presenting a more authentic environment, often at the cost of increased risk and complexity in terms of management and analysis.

As reported in [1], an early and notable application of honeypots occurred in 1986 by Clifford Stoll, a system administrator at the University of California, Berkeley. In his book, 'The Cuckoo's Egg' [2], Stoll recounts discovering an unauthorized user with superuser access penetrating his system. He implemented a honeypot strategy to catch the intruder, using borrowed terminals connected to the system's incoming phone lines. This approach allowed him to discern the attacker's intentions, ultimately leading to the arrest of a German operative working for the KGB.

In 1990, the concept of honeypots gained significant traction through the influential work of Cheswick [3]. His pioneering efforts sparked the birth of the Honeynet Project [4], an initiative that continues to contribute to the field even today. [5]. This research aims to utilize honeypots to acquire a deeper understanding of the specific threat landscapes targeting cloud infrastructures. Furthermore, the research sought to analyze and highlight the differences between various cloud service providers, reinforcing our understanding of their unique security profiles.

2.2 Honeypot: Cloud Deployment

The term "*cloud*" suggests the data is airborne and accessible from anywhere, as it can migrate between different physical servers and locations [6]. The National Institute of Standards and Technology (NIST) outlines three distinct cloud service models:

- Software as a Service (SaaS)—This describes applications businesses subscribe to and operate on cloud infrastructure via the Internet.
- Platform as a Service (PaaS)—In this model, the vendor provides high-level application services, allowing the business to create its own custom applications. While the business can freely configure and deploy its application, it's still regulated by the provider with its hardware and resources.
- Infrastructure as a Service (IaaS)—This model provides on-demand access to computing resources such as servers, storage, networking, and virtualization.

In this analysis, we focus on the security aspects of Infrastructure as a Service (IaaS), since our honeypots were deployed as virtual servers properly configured.

The concept of deploying honeypot software in cloud environments has been previously examined. Earlier research [1] explored the correlation between the popularity of cloud providers and the frequency of attacks. Unlike past studies that focused on popularity, this current research shifts its emphasis to a more security-centered approach.

A similar shift can be observed in the recent work by *Orca Security* [7]. They deployed honeypots containing cloud access keys across multiple cloud providers, timing the time interval from deployment to the keys being discovered and exploited. Specifically, on the AWS cloud platform, it took just two minutes for cloud keys to be compromised.

This study has deployed a total of four honeypots, two of which were hosted on AWS and the other two on Azure cloud. The two honeypots hosted on Azure will be referred to in this study as Azure A and Azure B, and the two hosted on AWS will be designated as AWS A and AWS B. These honeypots all supported the SSH protocol, utilizing Cowrie [8] a medium-interaction honeypot. Furthermore, one honeypot on each cloud platform was configured to support a variety of other protocols such as FTP, SMB, and HTTP. For this multi-protocol support, the Dionaea low-interaction honeypot was utilized [9].

Honeypot	Cloud Platform	Supported Protocols	Location
Azure A	Azure	SSH, FTP, SMB, HTTP	Amsterdam
Azure B	Azure	SSH	Amsterdam
AWS A	AWS	SSH, FTP, SMB, HTTP	Amsterdam
AWS B	AWS	SSH	Amsterdam

Table 1: Deployment details of the honeypots

2.3 Methodology

The main goal is to understand how different security measures work and how well they can prevent threats. This analysis has two purposes. First, it helps us find the provider that offers the best protection against malicious actors, so we can make informed decisions when selecting a provider. Second, it helps us evaluate whether investing in honeypots is an effective way to enhance server security.

We have developed hypotheses that explain the tactics and patterns of different attack methods, and we will carefully study them to achieve our goal [10].

To initiate, an investigation is carried out on the time patterns and schemes that fit the honeypot data. Given the seemingly random nature of the collected data, one could infer:
H1: No notable temporal dependency exists among attacks.

Since most of the attack origins are inspected:
H2: No apparent spatial dependency exists among attacks.

Additionally, disparities among the targeted services are delved into:
H3: No dependency from the attacked protocol exists.

H3.1: Autonomous systems, from which attacks originate, are uniformly distributed across various cloud service providers

Since most open-source software is developed for Linux, the following hypothesis is examined:

H4: Most of the attacking nodes run a Linux-based OS.
It's evident that not all attackers behave in the same way. Gaining insights into these varying behaviors is a pivotal objective of this study. Thus, the following hypotheses are examined:
H5: Attackers can be segregated based on their behavior.

H5.1: The resources of the attacker do not exhibit a significant dependency.

H5.2: The skill level of the attacker does not exhibit a significant dependency.

H5.3: Attacks originating from TOR nodes are not significantly prevalent.

The study also investigates the origins, types, and goals of malware attacks:

H6:All geographical regions contribute equally to the origins of malware activities.

H6.1: The contribution of geographical regions to the origins of SSH malware attacks is not equal.

H6.2: All geographical regions contribute equally to the origins of malware uploads.

H6.3: The contribution of geographical regions to malware-hosting is not equal.

H6.4: The use of a common SSH key across multiple attacks does not suggest a coordinated operation.

H6.5: All types of malware are downloaded with equal frequency.

The study further delves into the differences in the observed data between the two cloud providers, Azure and AWS, predicated on the following hypothesis:

H7: There is a significant correlation between the choice of the cloud provider for hosting and the observed activity.

H7.1: The data observed on machines within the same cloud provider do not show a significant dependency.

H7.2: The likelihood of a successful SSH login attempt is relatively high

H7.3: The risk of an attacker downloading or uploading malware is substantial. itemize

Lastly, hypotheses H8 and H8.1 are proposed to investigate two insights:

- Attacks targeting IoT devices are the most common.
- Mirai botnets play a substantial role in the overall landscape of attacks.

H8: There is a significant volume of attacks that target IoT devices.

H8.1: A *significant volume of attacks originate from Mirai botnets.*

2.3.1 Deployment details

We have designed and implemented many scripts to perform our statistical analysis. The most important one implement correlation, trend detection, and autocorrelation computations on the time series. This customized approach was due to the specific requirements of our study, which included accessing data directly from the Prometheus[1] server and generating correlation graphs with data points.

The outputs of the computations are represented within a Pandas[2]data frame [11]. This structured data representation simplifies the management and manipulation of the acquired data. To further enhance the visualization and comprehension of our findings, we adopted *Pyplot* to transform the data into a time-series format. This rigorous and customized analytical process allowed us to explore correlations and trends within our dataset.

The system exploits both virtual machines (VMs) and Docker containers [12] to build a scalable and isolated environment for the honeypot servers. We use VMs as protective barriers between the network and the hosted honeypots. This approach effectively confines potential security breaches within the containers themselves, thereby preserving the host system's integrity.

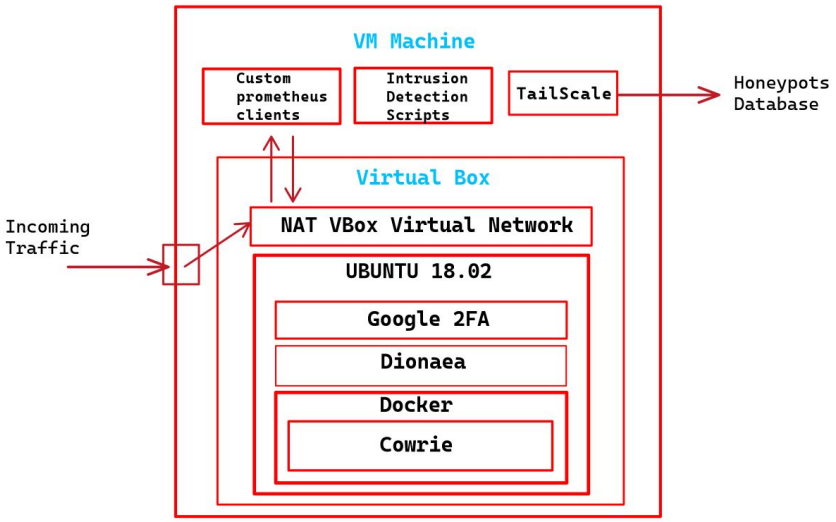

Figure 2.1: Honeypot machine Architecture

To improve the security standards of our system, we've incorporated a sophisticated alert system that activates immediately when access is granted to either the host or the virtual machine. This proactive system serves as an instant notification mech-

[1]Prometheus: An open-source time-series database used for real-time systems monitoring and alerting, originally developed at SoundCloud.

[2]Pandas: open-source data analysis library for Python, providing flexible data structures for handling and analyzing complex datasets.

anism, ensuring prompt responses to any potential security threats and preserving our security infrastructure's continuous robustness. In addition, we've deployed a protective protocol to suspend the operations of either the Docker container or the VM when an unauthorized login attempt is detected while the system is in an armed state. This automatic halt mechanism minimizes potential damage and provides ample time for security responses. For securing access to the VMs themselves, we've fortified them with a two-factor authentication (2FA) system [13], providing an added layer of protection against unauthorized access.

Because of the huge volume of logs produced by the honeypots, we've strategically implemented a Centralized Server, leveraging MongoDB [14] for optimized storage and proficient data management. Our preference for MongoDB stems from its inherent flexibility, which includes crafting custom scripts to safely export data from the secure confines of the honeypot environment. Moreover, it allows us to formulate efficient query scripts tailored for handling the extensive dataset. This synergy significantly streamlines and enhances the analytical process.

For real-time monitoring and time-series data analysis, we have integrated Prometheus [15]using custom scripts named Prometheus Collectors that enable the integration of this time-series database with our VMs and honeypots, measuring not only log data but also the VMs environment metrics. However, since Prometheus is primarily a metric collection tool, we have paired it with Grafana [16]. Grafana, with its customizable dashboards and MongoDB integration, allows us to visualize our metrics and analyze the data more comprehensively. The flawless integration between Grafana and MongoDB is achieved using custom scripts taken from a GitHub repository *mongodb-grafana* by James Osgood [17], the forked version of which is available by Nes Cohen [18].

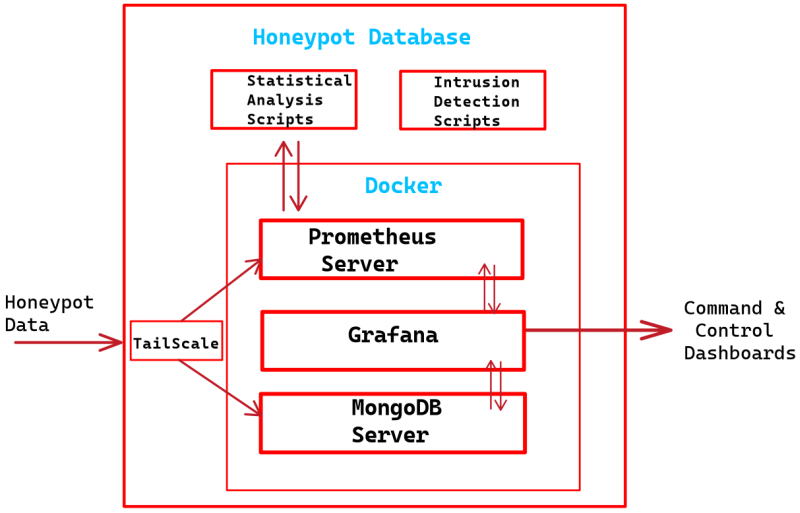

Figure 2.2: Honeypot Server Architecture

Finally, the data collected from each honeypot is securely transmitted to a server hosting both Prometheus and MongoDB through the Tailscale network [19] for enhanced privacy and security. Tailscale provides a reliable and encrypted network tunnel that protects the data transfer between the honeypots and the storage server from potential eavesdropping or unauthorized access.

In summary, the combination of Tailscale, for safe data transferring, Prometheus for time series generation, Grafana for data visualization as in fig 2.3 and 2.4, and MongoDB for efficiently storing the substantial dataset for more fined grained analysis, enabled us to create a robust and secure ecosystem for collecting, storing, and analyzing honeypot data.

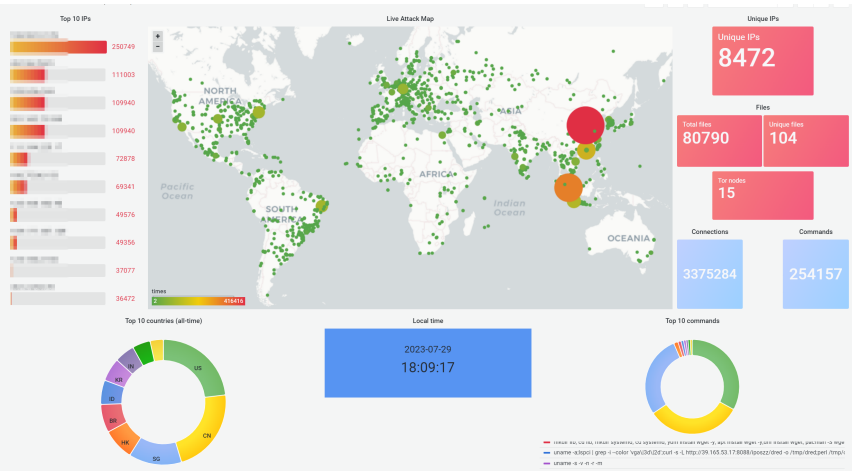

Figure 2.3: Grafana attacks Dashboard 1

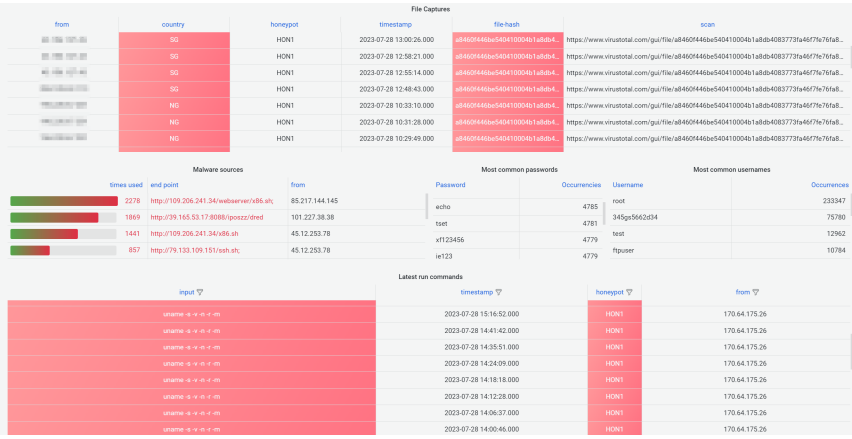

Figure 2.4: Grafana attacks Dashboard 2

2.4 Underlying Assumptions

These are the assumptions Underlying our analysis:

- Within the scope of this research, an "attack" is any inbound interaction towards the system housing the honeypot. We regard all these as malicious, since the system isn't hosting any applications beyond the honeypot software itself, and as such, there shouldn't be any legitimate connections taking place.
- The term "Malware Upload" denotes the execution of commands or a series of commands to exploit the SSH machine. The main goal of these commands is to hijack the SSH-authorized keys. After acquiring these keys, the attacker gains a backdoor into the system, which significantly eases subsequent access into the compromised system. Alternatively, where specified, we refer to Malware uploaded using SMB protocol to the Honeypots
- On the other hand, "Malware Downloads" denotes situations where malware is directly fetched and downloaded onto the host machine through web requests. These requests are usually implemented via '*wget*' or '*curl*' commands triggered by the attackers.
- When presenting time series data, the analysis zeroes in on the week from 06/19/2023 to 06/26/2023. Yet, when the data is represented in non-time series formats—like aggregated data or charts—it covers a more extensive time frame, specifically a two-month collection period.

3 Understanding Attack Timing

In this section, we delve into the temporal aspect of cyber attacks, a critical factor that is often neglected. While most security analyses focus on technological sophistication and sources of attacks, understanding 'when' attacks occur can provide valuable insights for threat mitigation.

This section delves into a detailed examination of the temporal patterns observed in cyber attacks (2.1), an intriguing contrast between the frequency of attacks during the day versus night (2.2), and a comprehensive analysis of timing patterns in attacks using autocorrelation methods (2.3).

3.1 The When of Cyber Attacks: Temporal Analysis

To initiate the analysis, we present and discuss the time series for each honeypot, isolated on a weekly basis. The time series illustrates the quantity of SSH attacks over time, captured with a sampling rate of 30 seconds. In addition, a detailed analysis is conducted for each honeypot, breaking down the data by day of the week and hour to provide further granularity.

- **3.1.1 Azure**

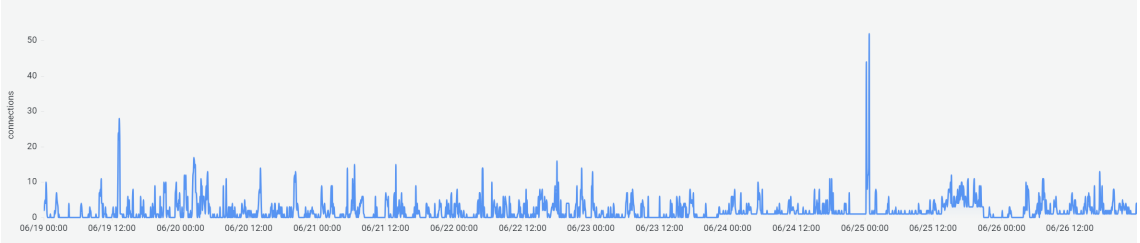

Figure 3.1: Attacks' distribution on Azure A over a week

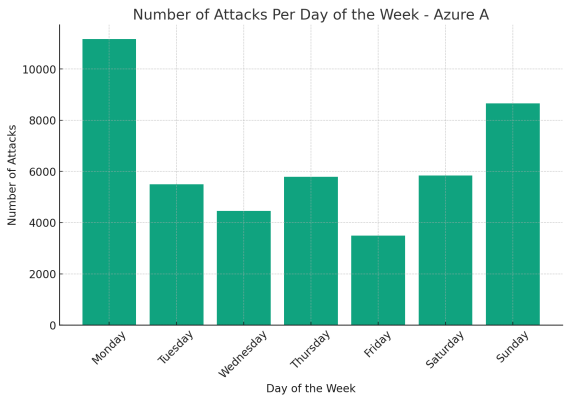

Figure 3.2: Attacks per weekday - Azure A

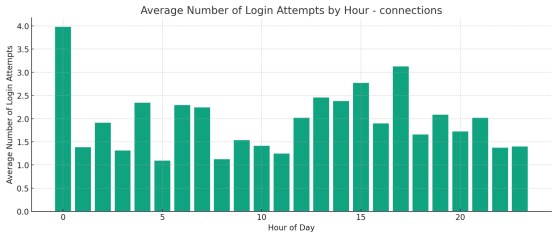

Figure 3.3: Attacks by hour - Azure A

Figure 3.2 shows that the distribution of login attempts throughout the week is not uniform on Azure A. It is noteworthy that the highest number of attacks occurred on Monday, decreasing throughout the week, and a slight increase over the weekend.

Turning our attention to Figure 3.3, a pronounced peak can be observed around midnight. This is when the average number of SSH login attempts significantly escalates, indicating that it is the preferred time for attacks.

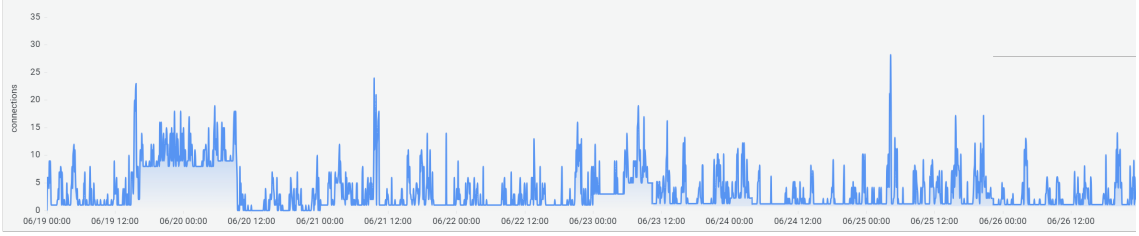

Figure 3.4: Attacks' distribution on Azure B over a week

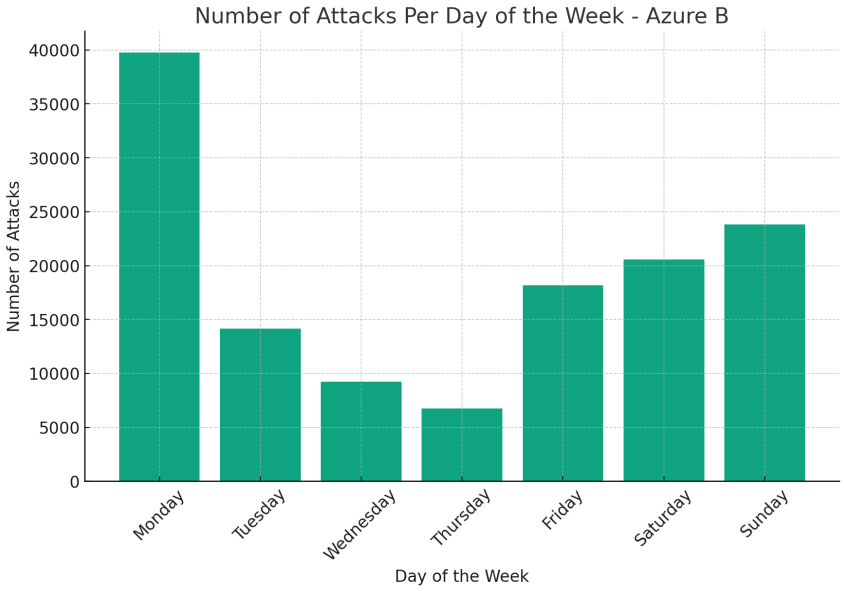

Figure 3.5: Attacks by weekday - Azure B

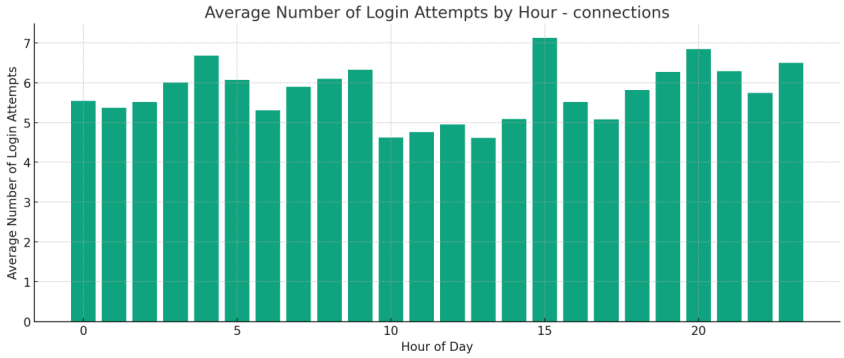

Figure 3.6: Attacks by the hour - Azure B

The frequency of access attempts exhibits a distinct variance based on the day of the week on Azure B as well, as shown in Fig3.4. Notably, Mondays register the highest average number of attempts, as already seen on Azure A. The number of attacks decreases until Thursday, with an increase over the weekend. The number of attacks on Sunday is particularly with respect to the other days.

In contrast to Azure A , the distribution of authentication attempts throughout the day in the current system is relatively uniform. However, a minor surge in the average number of login attempts can be discerned around 15:00 (3:00 PM) as shown in Fig 3.6. This peak, while observable, is markedly less conspicuous than the one in Azure A.

• 3.1.2 AWS

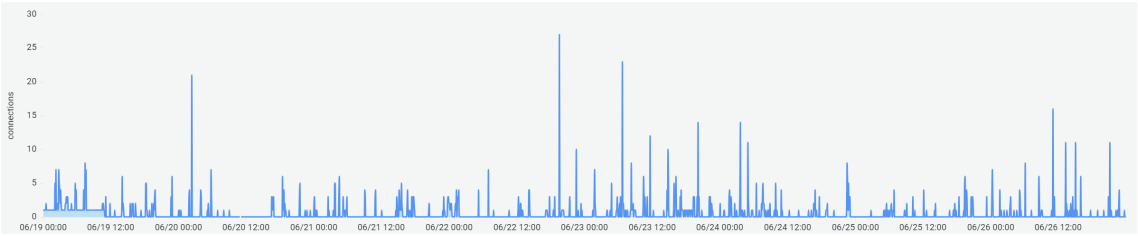

Figure 3.7: Attacks' distribution on AWS A over a week

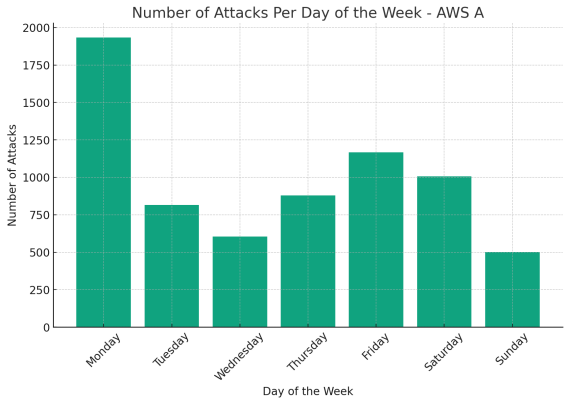

Figure 3.8: Attacks per weekday - AWS A

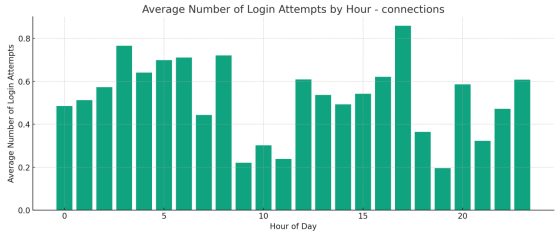

Figure 3.9: Attacks by the hour - AWS A

Figure 3.8, shows the connection pattern on AWS A, it is evident that the distribution of activity is not uniform throughout the week. The largest number of attacks occurs on Monday, followed by a sharp drop on Tuesday. The attacks then decrease gradually throughout the rest of the week, with a slight increase on Saturday and another drop on Sunday.

A further observation from Figure 3.9 shows that some timeframes, particularly around 17:00, result in a large average number of login attempts. This suggests a notable surge in activity during these periods

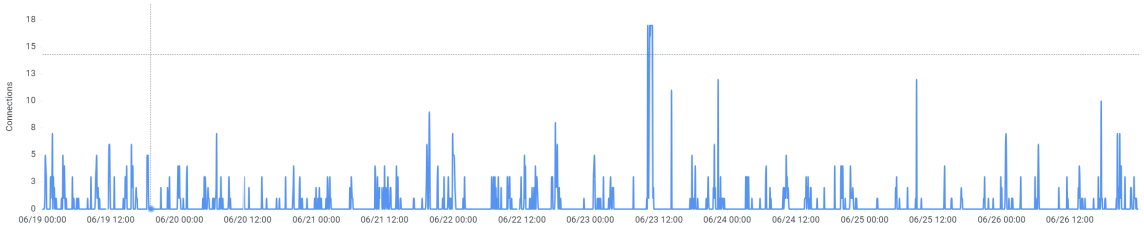

Figure 3.10: Attacks' distribution on AWS B over a week

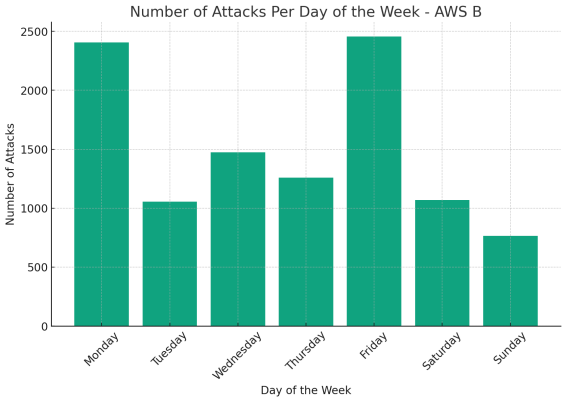

Figure 3.11: Attacks per weekday - AWS B

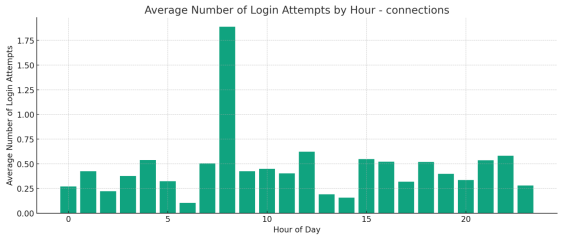

Figure 3.12: Attacks by the hour - AWS B

An analysis of AWS B shows similar trends to AWS A regarding the non-uniform distribution of login attempts throughout the day. Nevertheless, the pattern of these attempts exhibits distinct differences. A key observation is the peak in attack occurrences on Monday, followed by a gradual decrease as the week progresses. Notably, unlike AWS A, AWS B does not show a perceptible increase in attacks during the weekend.

Additionally, Fig. 3.12 shows a surge in activity around 8:00, characterized by a rise in the average number of login attempts. This suggests an interval of heightened cyber activity and potential susceptibility to breaches.

Interestingly, an analysis of the relation between attacks and the day of the week confirms that there is a noticeable fluctuation in login attempts. Fridays, for instance, endure the highest average volume of attempts.

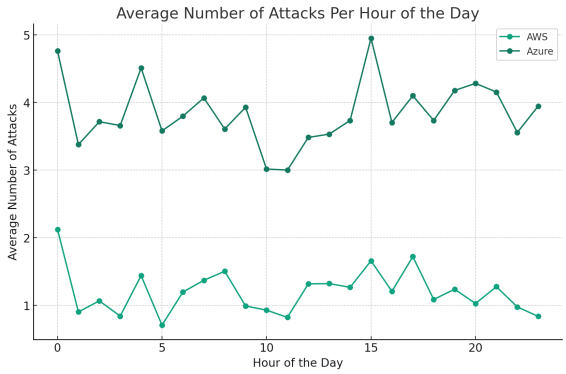

Figure 3.13: Average attacks by hour on both providers

Fig. 3.13 confirm that the two cloud providers have a similar distribution of the average number of attacks throughout the day even though AWS experienced significantly fewer attacks on SSH than Azure.

3.1.3 Findings

The examination of SSH login attempts over time revealed an uneven yet similar distribution of connections on all honeypots. A few key observations can be drawn from this analysis:

- Most attacks occur on Monday on both cloud environments.
- Attacks in the AWS environment mainly occur towards the close of the workweek, recording lower values on Sundays—a pattern that deviates from the one observed within the Azure environment.
- There is a clear variation in the volume of attacks between the servers. AWS B has a lower number of attacks than AWS A, while Azure B has a much higher number of attacks than Azure A

These findings clearly reveal a clear temporal dependency in the attack patterns, thus contradicting Hypothesis *H1* which posits an absence of such a dependency. To reinforce this claim and further investigate this temporal relationship, we will continue our analysis throughout this study.

3.2 Time zones: Attacker's Perspective

To enhance our understanding of attack timing, our analysis zeroes in on the specific times these incidents take place, especially from the attacker's perspective. To offer a clear visualization of the results, we partition the day into three periods.

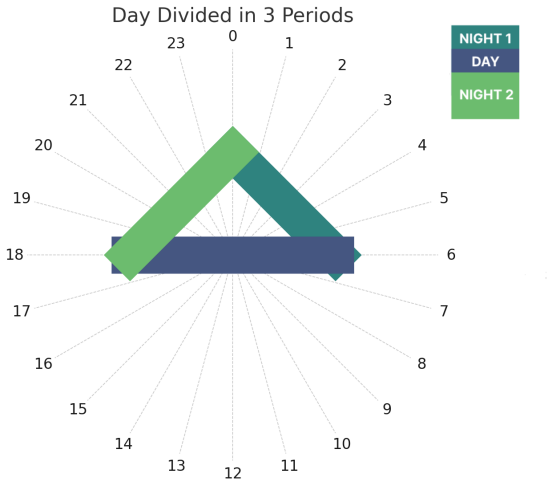

Figure 3.14: Day periods

As shown in Fig. 3.14 the term 'Night1' refers to the period from midnight to 6 AM, 'Day' encompasses the timeframe from 6 AM to 6 PM, and 'Night2' covers the hours from 6 PM to midnight Although the honeypots are situated in Amsterdam, we utilize Coordinated Universal Time (UTC) in our analysis. This + simplifies the comparison among the attackers' time zones and yields identical results, because of the negligible time difference.

Fig 3.15 and 3.16 show the distribution of the number of executed SSH commands over different periods of the day on both cloud providers from two different prospectives: One of the honeypots and the attackers and the attackers'. This is done by taking into account the timezone the command was run from. This allows us to get accurate results in this daytime, and nighttime analysis.

3.2.1 Azure

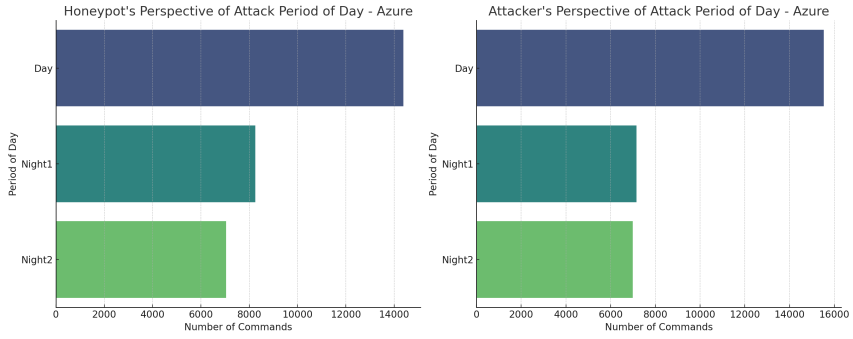

Figure 3.15: SSH commands execution over different periods of the day - Azure

3.2.2 AWS

Figure 3.16: SSH commands execution over different periods of the day - AWS

Given that the 'Day' period lasts 12 hours, there's a noticeable increase in run commands from midnight to 6 AM during the 'Night1' period. If we merge both the 'Night1' and 'Night2' periods, we find that, from the perspective of the honeypot, commands are executed slightly more during the night.

Switching to the attackers' viewpoint, the distribution of attacks across the night periods seems to be more evenly balanced. When we consider these periods together, it's observed that the quantity of run commands executed during the night aligns with those carried out during the day.

3.3 Time zones: Findings

From the honeypot's perspective, we observe a higher frequency of command executions during the night, particularly from 6 PM to midnight. When merging both night periods, it becomes clear that commands are executed more often during the night.

When switching to the attackers' viewpoint, we discover that the number of run commands during the 'Night2' period closely parallels that of the 'Day' period. This observation is in line with AWS's analysis. When considering both the 'Night1' and 'Night2 periods, it's evident that, from an attacker's perspective, there's a trend towards executing more attacks during the night.

Findings

Based on the presented data, the following conclusions can be drawn:

1. Azure: Honeypot prospective attacks occur slightly more frequently during the night, and from the attacker's perspective, the attacks are more evenly distributed throughout the day.

2. AWS: Attacks occur more frequently during the night from both perspectives. Intriguingly, from the honeypot's viewpoint, attacks concentrate between 6 AM and midnight, whereas from the attacker's perspective, most commands are executed between midnight and 6 AM. This discrepancy may result from the timezone difference between the two perspectives.

To fully understand the patterns observed on AWS, it becomes essential to shift our focus to the role of time zones.

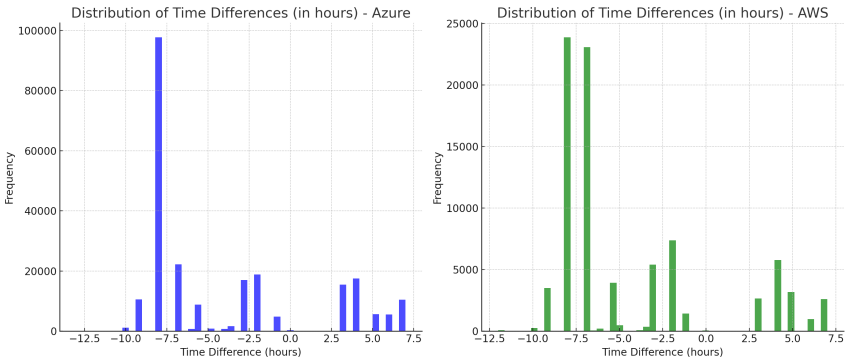

Figure 3.17: Distribution of the difference between the attacker's time zone and honeypot's time zone

Figure 3.17 illustrates the distribution of the time difference between the two perspectives. Notably, most commands are executed with a time difference of -8 hours on both cloud providers, indicating that the attacker's timezone lags by 8 hours.

An intriguing discrepancy emerges when AWS shows a significant spike around the -7.5 hours mark, a pattern not evident in Azure. This agrees with our earlier observation of increased attacks during the 'night1' period from the attacker's perspective. This disparity might be due to the geographical distribution of attacks, a topic that will be further explored in the following chapter.

3.4 Autocorrelation Analysis

To delve deeper into the temporal patterns of attacks, we will apply autocorrelation analysis on various metrics associated with each honeypot, using a specific offset, denoted as α. This process will help to focus on recurring attack patterns, to simplify the discovery of hidden behavioral trends among the attacks.

3.4.1 Determining the Optimal Offset α

The following graphs illustrate the fluctuations in the average auto-correlation value of *SSH login attempts* across different time lags. The data has been resampled at a minute level. For instance, a lag of 30 in the plots corresponds to a 30-minute time interval.

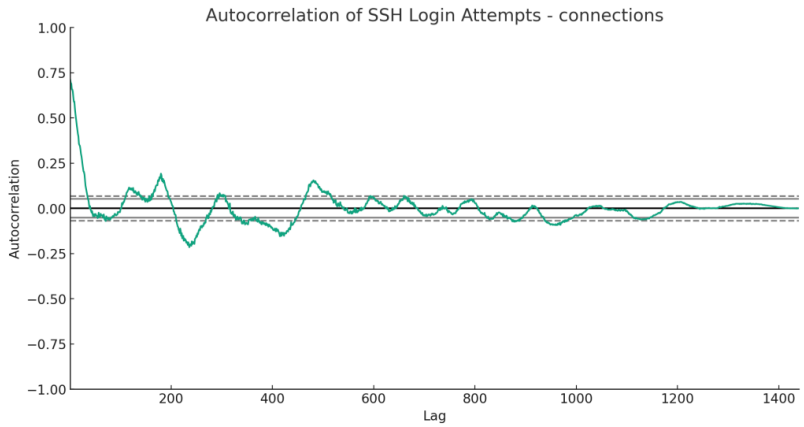

Figure 3.18: Lag α in minutes - Azure A

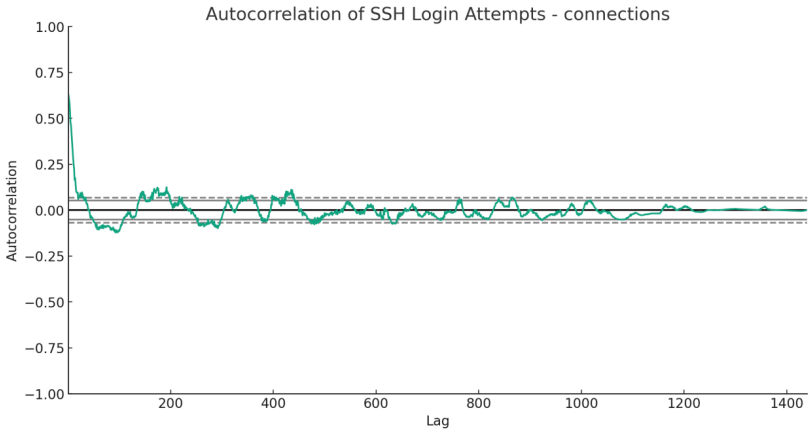

Figure 3.19: Lag α in minutes - Azure B

Fig. 3.18 and 3.19 show that the correlation values peak with lower values of α. As α increases, the correlation values tend to fluctuate, ultimately converging to zero when the lag exceeds 600 minutes (10 hours).

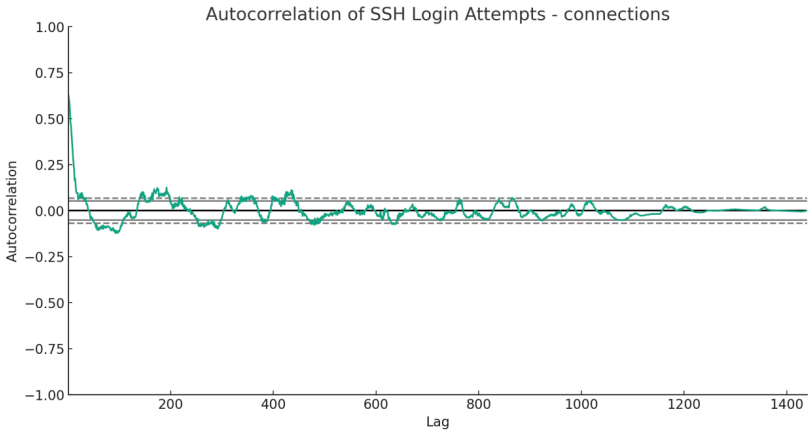

Figure 3.20: Lag α in minutes - AWS A

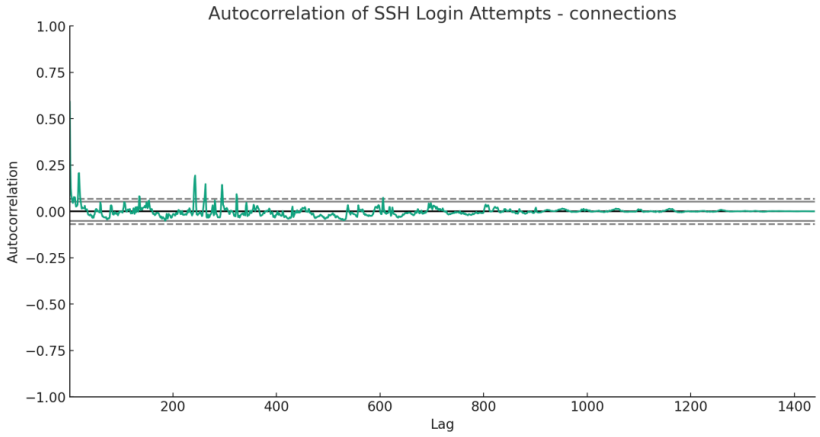

Figure 3.21: Lag α in minutes - AWS B

Concerning AWS, we found that similar to Azure, lower values of α result in higher correlations. However, this phenomenon is less present on AWS B because of the lower count of attacks observed.

Our findings from both cloud providers led us to select α as 3, which equates to a time offset of 3 minutes. We based this decision on our results, which revealed a strong for this specific offset in our study. This suggests that most attacks recur not hourly or daily, but rather minutely.

This marked interrelation formed the basis of our hypothesis that utilizing the same time delta for autocorrelations across all SSH metrics would yield valid and reliable results. Furthermore, this approach simplifies the comprehension of results across all honeypots.

3.4.2 Autocorrelating SSH Metrics: Azure A

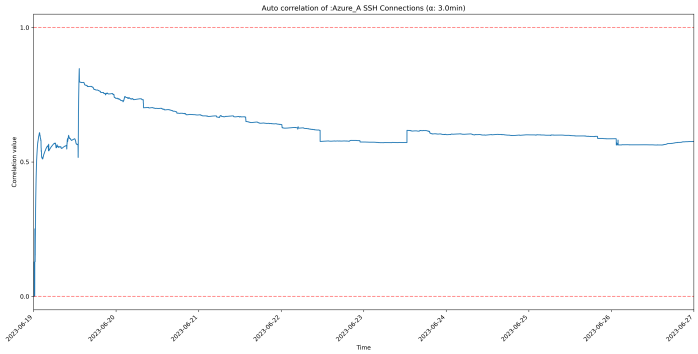

Figure 3.22: Login Attempts (SSH)

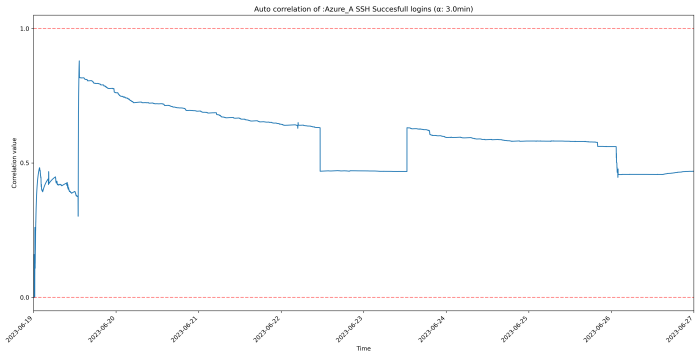

Figure 3.23: Successful logins (SSH)

The autocorrelation graph shows a significant, abrupt decrease (lasting for 24 hours from 2023-06-22 at 12:00) in the autocorrelation graphs for both login attempts and successful login in figures 3.22 and 3.23. This pattern may suggest that an automated script, which was targeting the honeypot at a regular interval (3 minutes), unexpectedly stopped operations for roughly a day before resuming its activities. The more noticeable drop in the autocorrelation graph for successful logins may be due to the typically lower occurrence of successful logins with respect to total attempts. Consequently, the temporary halt in the script's activity has a more discernible impact on the overall correlation.

3.4.3 Autocorrelating SSH Metrics: Azure B

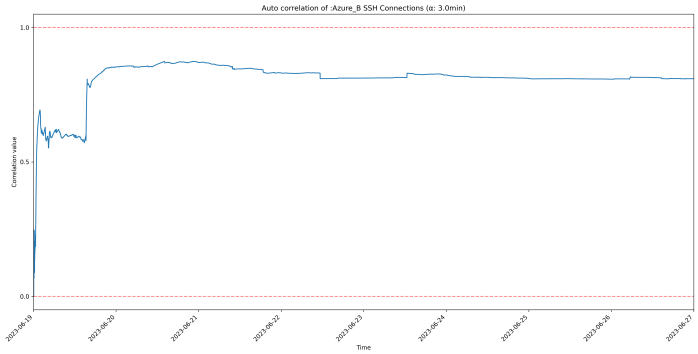

Figure 3.24: Login Attempts (SSH)

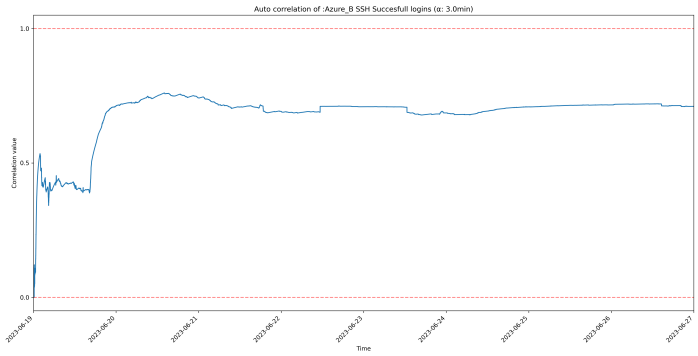

Figure 3.25: Successful logins (SSH)

Besides the heightened autocorrelation observed between total login attempts and successful ones, we identify an intriguing shift in the pattern.
The sudden drop previously observed in Azure A has unexpectedly transformed into a sharp spike, as shown in Fig. 3.24 and 3.25. This likely indicates that an automated script temporarily shifted its focus from Azure A to Azure B.

3.4.4 Autocorrelating SSH Metrics: AWS A

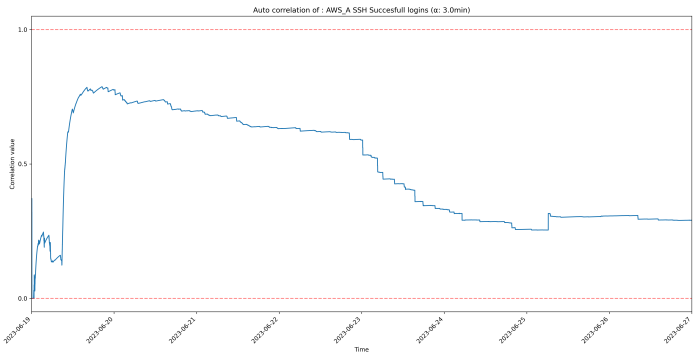

Figure 3.26: Successful logins (SSH)

Switching the cloud provider from Azure to AWS reveals some intriguing differences in autocorrelation patterns. For example, the variability in the correlation of successful logins suggests a more unpredictable pattern in the timing of attacks.

Furthermore, a significant finding emerges: there is an abrupt decline in correlation on AWS successful logins, as shown in figure 3.26, mirroring similar patterns previously observed on Azure, even if with a noticeable time lag. These findings further strengthen the hypothesis that we are dealing with a botnet[3] targeting both cloud providers.

3.4.5 Autocorrelating SSH Metrics: AWS B

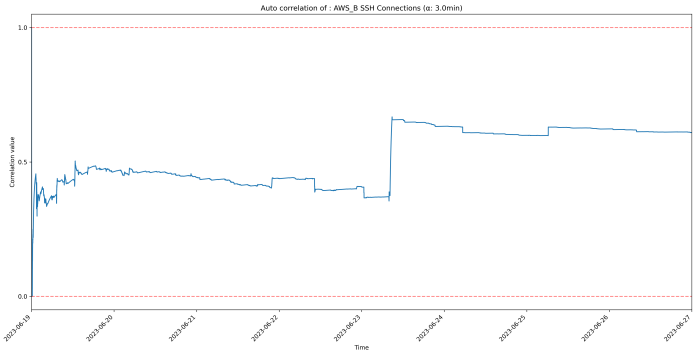

Figure 3.27: Login Attempts (SSH)

[3]A botnet refers to a network of computers infected with malicious software and controlled as a group without the owners' knowledge, often used for nefarious activities such as sending spam or conducting distributed denial-of-service attacks.

Interestingly, we notice a significant increase in both connections and commands in Fig. 3.27 and 3.28, suggesting a potential occurrence of an automated attack executed by a script working periodically. Additionally, we have noticed a repeated decrease in correlation across multiple instances on both AWS and Azure platforms. This repeated pattern lends additional support to the hypothesis that the same botnet targeted all the servers.

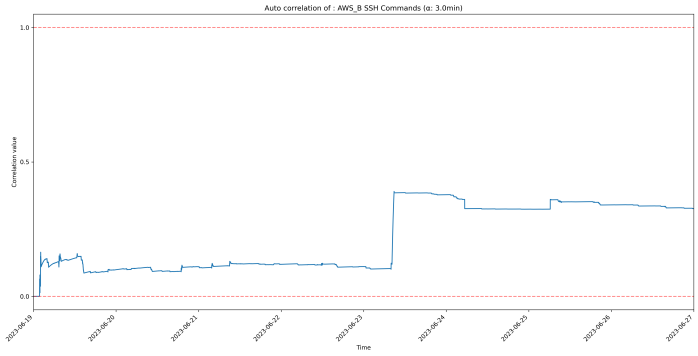

Figure 3.28: Commands (SSH)

3.4.6 Autocorrelating SMB Metrics: Azure A

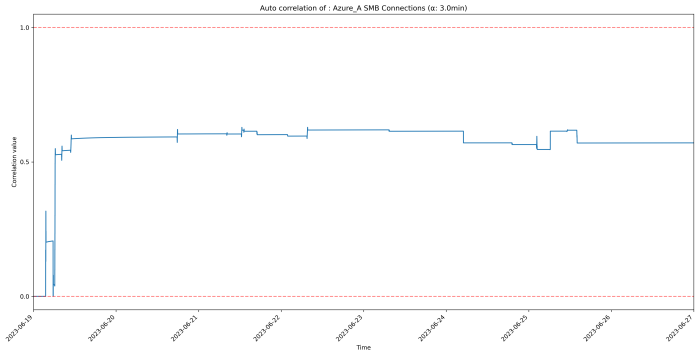

Figure 3.29: Connections (SMB)

We observe a correlation coefficient of roughly 0.6 in figure 3.29. Intriguingly, we also detect a consistent decrease in correlation, indicating a similar pattern across the data. This observation prompts us to conjecture that the aforementioned botnet may not have been confined to SSH attacks but could have also executed SMB attacks. This implies that the botnet's activities were not restricted to a single protocol but spanned multiple attack vectors.

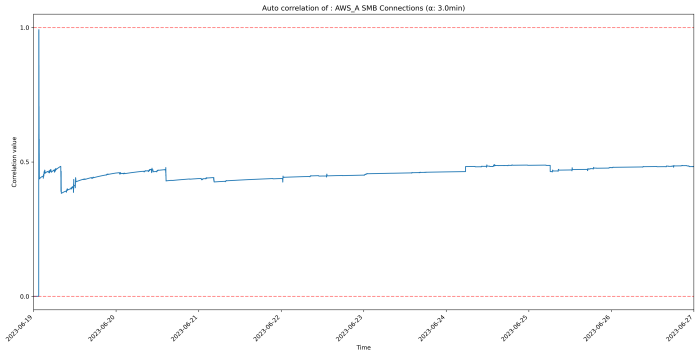

Figure 3.30: Connections (SMB)

3.4.7 Autocorrelating SMB Metrics: AWS A

Figure 3.30 shows that in AWS we have a relatively lower correlation level, hovering in the range of 0.4-0.5. However, on Azure, the previously observed drop in correlation morphs into a spike, indicating a significant shift in SMB attacks from AWS to Azure. This shift suggests that the same perpetrator or entity behind the attacks has now targeted AWS B, underscoring their adaptability in choosing and attacking different server instances.

3.4.8 Inconclusive Correlations

In our analysis, autocorrelations of FTP, SMB, and HTTP protocols did not yield any useful insights and were therefore omitted.

3.4.9 Findings

Autocorrelations were analyzed for different protocols and cloud providers. The chosen value α is the offset in minutes, and correlations were observed with a precision of 30 seconds.

1. SSH:
    - Azure A: Stronger autocorrelation between login attempts and successful logins than commands. A significant drop in autocorrelation patterns for 24 hours may be due to a script attacking the honeypot.
    - Azure B: Similar observations as Azure A, with a spike in autocorrelation after a previous drop, suggesting that a script temporarily shifted its focus.
    - AWS A: District cloud provider with fluctuations in the autocorrelation of successful logins, indicating an erratic attack pattern. A sudden drop in correlation is similar to Azure, indicating the presence of a botnet across different platforms.
    - AWS B: Increased connections and commands, suggesting an automated attack. Repeated decreases in correlation further support the hypothesis of a shared botnet across all servers.
2. SMB:
    - Azure: Moderate correlation level with a consistent decrease, indicating possible SMB attacks conducted by the same botnet targeting multiple protocols.
    - AWS: Comparatively lower correlation level, indicating SMB attacks but with less consistency.

.

3.5 Temporal Analysis: Findings

In conclusion, this chapter has provided a comprehensive overview of the deployment and analysis of honeypots in cloud infrastructures, specifically targeting AWS and Azure. The primary goal was to understand the patterns and tactics of various attack strategies. The study proposed multiple hypotheses, including Hypothesis *H1*: "No notable temporal dependency can be discerned."

However, analyzing the data collected from the honeypots revealed a clear temporal dependency in the observed attack patterns. The study found that the frequency of attacks varied based on the day of the week and the time of the day. Namely observing that most attacks occur on Mondays. Additionally, the study suggests a preference for attacks occurring during the attacker's local nighttime. However, it is essential to underscore that the automated nature of most cyberattacks implies that these activities can be executed without specific regard to day or night. This lack of temporal preference is largely due to the automated systems and scripts that can initiate attacks at any time. The intricacies of this automated behavior will be thoroughly examined in Chapter 6.

Therefore, based on the evidence gathered and analyzed in this chapter, Hypothesis *H1* is rejected. The findings clearly indicate that there is a significant temporal dependency in the attack patterns on the honeypots deployed in the cloud infrastructures.

4 Spatial Analysis: Deciphering the Geographical Origins of Attacks

4.1 Origins of Cyber Threats: A Closer Look

After understanding when most cyber-attacks occur, the source, or 'from', of the attacks becomes an evident subsequent subject of analysis to obtain a deeper understanding of the attacker's behavior.

Our research proceeds in steps, initially examining each cloud environment individually. Subsequently, we compare these findings to gain fresh perspectives on cloud-targeted attacks. This enables us to delve into the disparities that might be unique to different cloud environments.

To increase the granularity of our analysis, we consider that some attackers may engage in multiple assaults. Thus, an initial investigation based on the overall number of attacks is followed by a more detailed analysis focused on unique IP addresses in order to analyze the count of individual attackers rather than the total number of attacks.

4.1.1 Total attacks - Azure

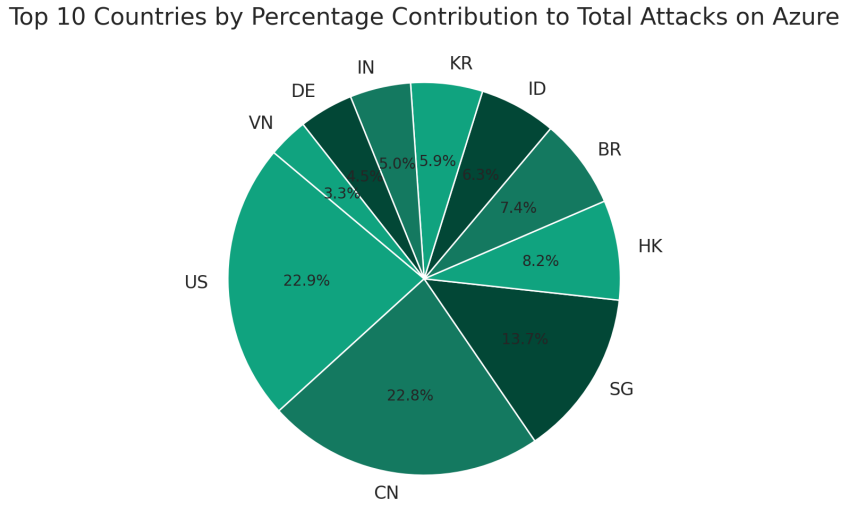

Figure 4.1: Top 10 Attackers' countries by percentage - Azure

As shown in figure 4.1 most attackers on Azure are from the United States (22.9%) followed by China (22.8%) and Singapore (13.7%), making Asia the continent from where most attacks come. Figure 4.2 shows the total attacks from each country, for a total of over 3 million connections.

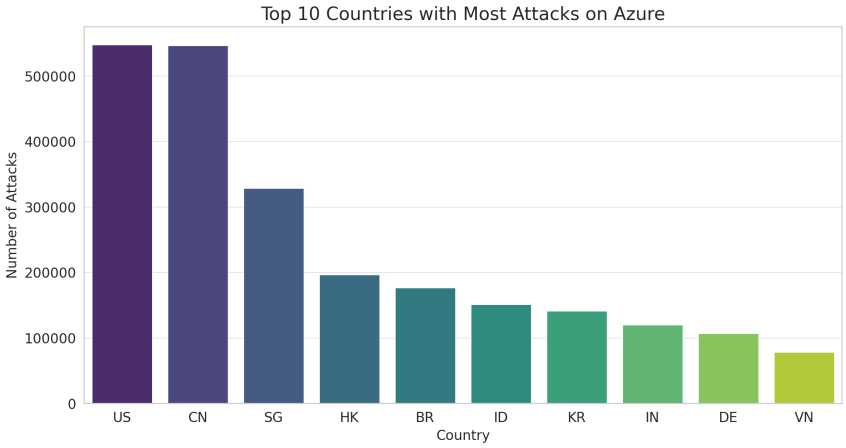

Figure 4.2: Top 10 Countries by total attacks- Azure

4.1.2 Total attacks - AWS

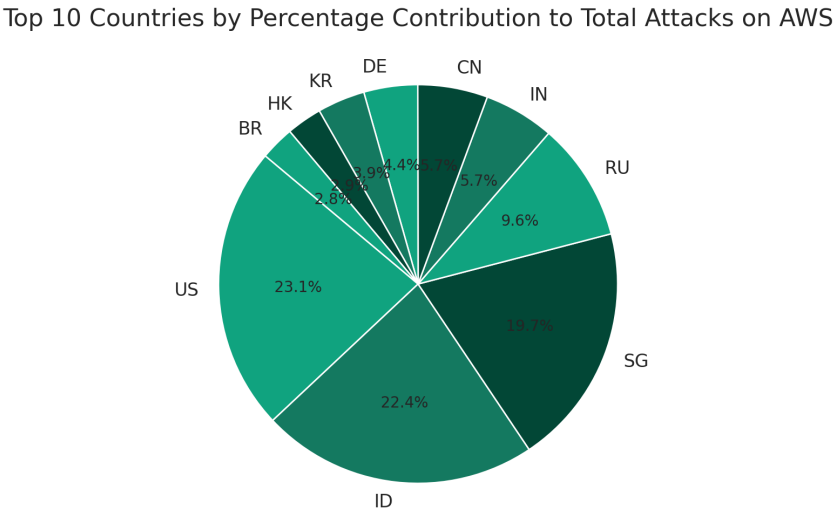

Figure 4.3: Top 10 countries by percentage - AWS

Fig. 4.3 shows that for AWS most attacks originate from the United States (23.1%), trailed by Indonesia (22.4%) and Singapore (19.7%). Asia stands out yet again as the continent responsible for most of these attacks. Fig. 4.4 presents the aggregate number of attacks from each country, accounting for almost 1 million connections, only a third of what was observed with Azure.

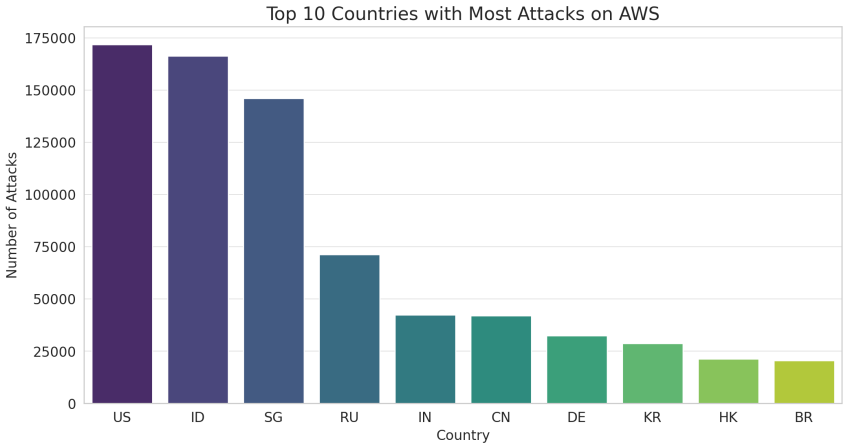

Figure 4.4: Top 10 countries by total attacks - AWS

4.1.3 Clouds Compared by IP Count

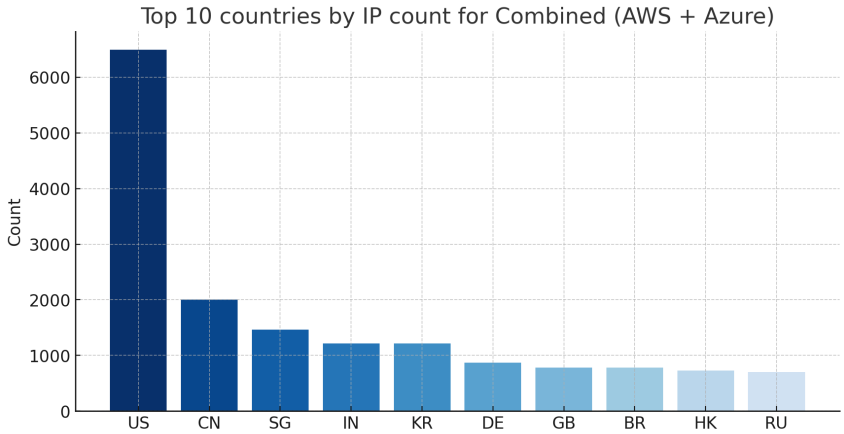

Figure 4.5: Top 10 countries by IP count - Combined

Analyzing from an IP perspective, we encountered 12,178 unique IP addresses for AWS and 11,917 for Azure, resulting in a total of 20,250 unique IP addresses (counting only once the common ones) across both platforms. By associating these IP addresses with their originating countries, we gain valuable insights into the geographical patterns and differences in attacks across the two platforms. Figure 4.5 shows the top 10 countries by IP count for each cloud provider and for the combined dataset.

Upon closer examination of the geographical distribution of attackers on each cloud platform, distinct patterns emerge, as shown in Figure 4.6. Even if the total number of attacks on Azure is larger than those on AWS, the diversity of actors targeting AWS is bigger. This indicates that numerous unique sources have attacked AWS.

For AWS, the top three countries with the most IP addresses are the US (3,245), China (1,308), and India (669). For Azure, the top three countries are the US (3,251), Singapore (844), and China (695).

Interestingly, each platform has unique countries from which it receives attacks. 15 countries, including the Bahamas, Belize, and Curacao, only appear in the AWS data. Azure has 16 unique countries, including Burundi, Cape Verde, and Cyprus.

However, there are also commonalities. A total of 126 countries appear in both the AWS and Azure data, including major countries like the US, China, India, Singapore, South Korea, Germany, Brazil, Russia, Hong Kong, and the UK.

Additionally, although Indonesia ranked second in the total number of attacks on AWS, we observed that it accounted for a relatively small proportion of unique attackers. This suggests that attacks from this country are repeated several times.

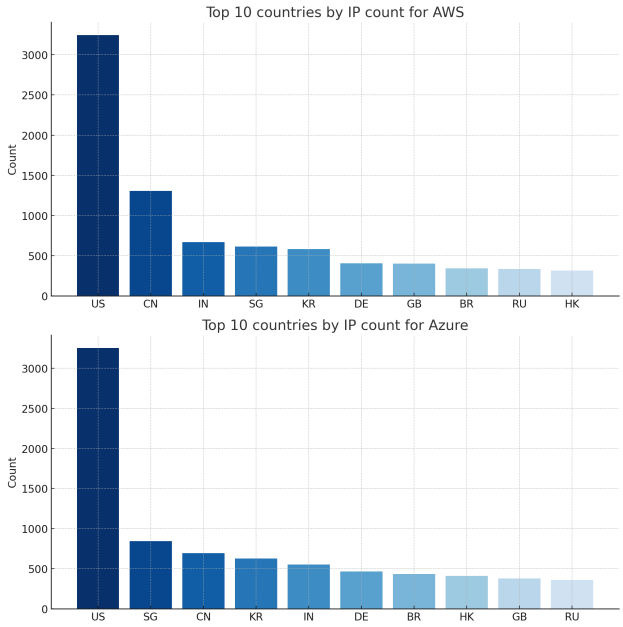

Figure 4.6: Top 10 countries by IP count

4.2 Normalization of Attack Data by Population and Geographical Area

In the forthcoming analysis, we adjust our data to account for variables such as population size and geographical area of each country as in other studies [10]. This data normalization method is crucial in gaining a clearer comprehension of the distribution pattern of cyber attacks. By incorporating these factors, we ensure that our interpretation is not skewed by sheer size or population, resulting in both more precise comparisons, and draw accurate conclusions.

For example, initial data might imply an increased volume of attacks originating from the United States, a conclusion that could simply reflect the country's substantial population and expansive geographic spread. This factor alone may skew the perceived threat level. By controlling for these aspects, we can more precisely comprehend the intensity of attacks based on per capita or per square kilometer calculations, thereby offering a more accurate and nuanced understanding.

Country	Attacks	GDP (Trillion USD)	Population (Million)	Attacks per Capita	R	R per GDP	R per Population
US	171717	21.43	331.0	518.782	0.225889	0.010541	0.000682
ID	166393	1.13	276.0	602.873	0.218885	0.193704	0.000793
SG	146011	0.361	5.7	25615.965	0.192073	0.53206	0.033697
RU	71263	1.48	145.0	491.469	0.093745	0.063341	0.000647
IN	42481	3.05	1390.0	30.562	0.055883	0.018322	4e-05
CN	42073	15.42	1410.0	29.839	0.055346	0.003589	3.9e-05
DE	32422	3.85	83.0	390.627	0.04265	0.011078	0.000514
KR	28676	1.63	51.0	562.275	0.037722	0.023143	0.00074
HK	21372	0.341	7.5	2849.6	0.028114	0.082447	0.003749
BR	20527	1.45	213.0	96.371	0.027003	0.018623	0.000127
VN	17248	0.341	98.0	176.0	0.022689	0.066537	0.000232

Figure 4.7: Attacks by Countries Normalized - AWS

In Fig. 4.7 and 4.8, the variable R denotes the ratio of attacks originating from a specific country to the total number of attacks from all countries. This ratio indicates each country's relative contribution to the overall attack volume. The values "R per GDP" and "R per Population" serve to normalize R by considering a country's Gross Domestic Product (GDP) and population size, respectively. These normalized metrics yield insights into the intensity of attacks in relation to a country's economic output and population. Specifically, a high "R per GDP" value implies a substantial volume of attacks compared to the nation's economic scale, while a high "R per Population" value points to a large number of attacks in proportion to its populace.

Country	Attacks	GDP (Trillion USD)	Population (Million)	Attacks per Capita	R	R per GDP	R per Population
US	547915	21.43	331.0	1655.33	0.223601	0.010434	0.000676
CN	546684	15.42	1410.0	387.719	0.223098	0.014468	0.000158
SG	329004	0.361	5.7	57720.0	0.134264	0.371924	0.023555
HK	196761	0.341	7.5	26234.8	0.080297	0.235475	0.010706
BR	177319	1.45	213.0	832.484	0.072363	0.049905	0.00034
ID	151706	1.13	276.0	549.659	0.06191	0.054788	0.000224
KR	141890	1.63	51.0	2782.16	0.057904	0.035524	0.001135
IN	120263	3.05	1390.0	86.52	0.049079	0.016091	3.5e-05
DE	107275	3.85	83.0	1292.47	0.043778	0.011371	0.000527
VN	78938	0.341	98.0	805.49	0.032214	0.094469	0.000329
RU	52663	1.48	145.0	363.193	0.021491	0.014521	0.000148

Figure 4.8: Attacks by Countries Normalized - Azure

Our data indicate that countries like the United States and China bear the brunt of the highest number of attacks across AWS and Azure platforms. However, upon normalizing GDP and population, smaller nations such as Singapore and Hong Kong exhibit a heightened intensity of attacks. Specifically, Indonesia and Singapore emerge prominently within the AWS environment when considering the number of attacks relative to GDP. Simultaneously, Singapore and Hong Kong are noteworthy when we evaluate attacks in relation to population size. Meanwhile, within the Azure environment, Singapore and Hong Kong present the highest frequency of attacks when analyzed in terms of GDP and population size.

4.3 Autonomous Systems: Exploring Attacks' Genesis

An Autonomous System (AS) can be described as a network of interconnected IP routing prefixes under the authority of one or more network operators, all representing a single administrative entity or domain. It presents a cohesive and precisely defined routing policy for the Internet.

To deepen our comprehension of the spatial distribution of attacks, we implement an analytical evaluation of the AS associated with each aggressor for both cloud service providers leveraging MaxMind's AS Database [20]. Our aim is to compare the emergent data, thereby revealing potential correlations or deviations.

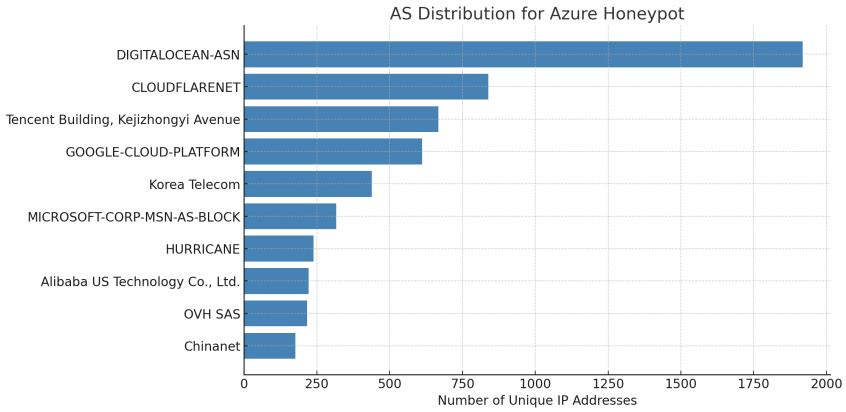

Figure 4.9: AS distribution - Azure

On Azure, DigitalOcean's prominence is observed. ASNs corresponding to Google Cloud Platform and Tencent also feature among the top ASNs, indicating that most attackers use cloud services to carry out attacks. ASNs associated with regional ISPs such as Korea Telecom and international cloud service providers like Alibaba are also common, emphasizing the widespread usage of these services. Interestingly, some ASNs like CLOUDFLARENET and MICROSOFT-CORP-MSN-AS-BLOCK are not prominent in the AWS environment in fig. 4.10, appear within the top ASNs for Azure, suggesting unique interaction patterns within the Azure environment.

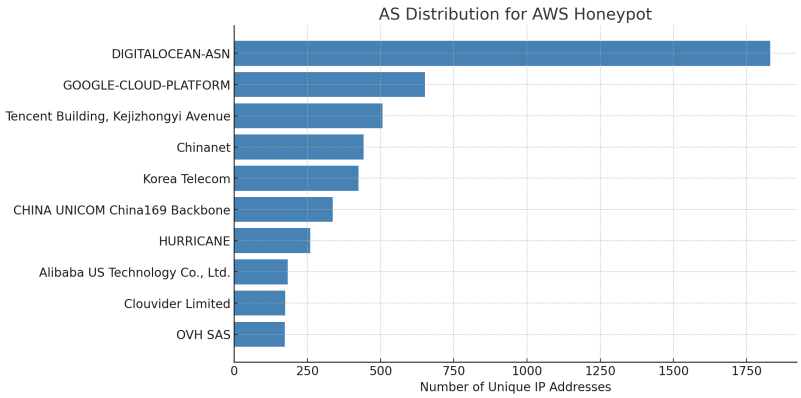

Figure 4.10: AS distribution - AWS

On AWS, the DigitalOcean AS emerges as the most common, potentially indicating that traffic in this environment frequently originates from, or is routed through DigitalOcean-hosted infrastructure. Furthermore, technology giants' presence, such as Google Cloud Platform and Tencent, is conspicuous, suggesting their infrastructures are recurrently used for AWS honeypot interactions.
Regional ISPs like Korea Telecom and international cloud providers like Alibaba also feature within the top ASNs. However, certain ASNs like CHINA UNICOM China169 Backbone and Clouvider Limited, which are not prominent in the Azure environment, appear within the top ASNs for AWS. This suggests that there are some patterns unique to the AWS environment.

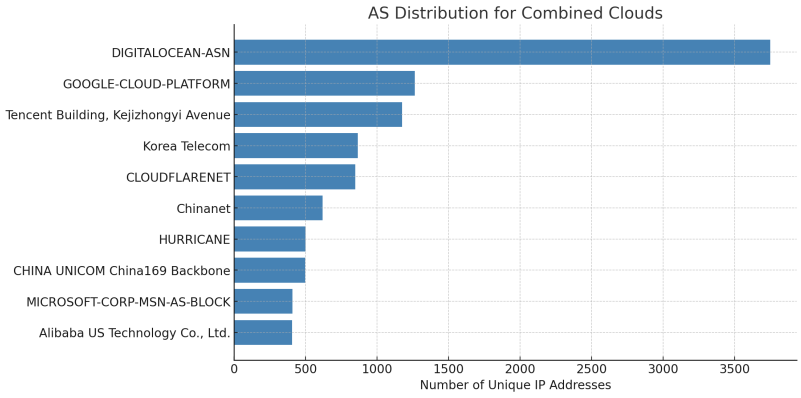

Figure 4.11: AS distribution for both clouds

When considering the combined dataset from both the AWS and Azure environments, DigitalOcean remains the most common ASN, implying its substantial role in the traffic observed across both environments. The frequent appearance of ASNs corresponding to Google Cloud Platform and Tencent highlights their widespread use in both environments. The presence of ASNs linked to regional ISPs like Korea Telecom and international cloud service providers like Alibaba underscores the global and distributed nature of the analyzed traffic.

4.4 Spatial Analysis: Findings

Our analysis of cyber threats originating from various geographical regions and their correlation with different cloud environments offers critical insights. We observed that the United States and China are the primary sources of attacks on both AWS and Azure platforms. When normalized for GDP and population, smaller nations like Singapore and Hong Kong displayed an unexpectedly high intensity of attacks, indicating a disproportionate cyber threat considering their size and economic scale.

Furthermore, examining the Autonomous System (AS) associated with each attacker revealed distinct patterns across the two cloud platforms. While DigitalOcean was the most common AS for both AWS and Azure, certain AS showed a pronounced association with specific cloud environments. For instance, CHINA UNICOM, China169, and Clouvider Limited were mostly linked to AWS, whereas CLOUDFLARENET and MICROSOFT were more commonly associated with Azure. According to our findings, the distribution of Autonomous Systems is not uniform across different cloud providers. This clearly contradicts Hypothesis *H3.1*, which assumes an even distribution of Autonomous Systems across cloud platforms. Therefore, we reject Hypothesis *H3.1*.

Our results confirm that it's clear that geographical origin does significantly influence the distribution of threats. This evidence contradicts the assertion made in Hypothesis *H2*, which assumes that there is no dependency on spatial factors for the occurrence of attacks. Consequently, we reject Hypothesis *H2*.

5 Protocol-Specific Analysis

5.1 Most attacked services: Cloud Providers Compared

Now, let's delve into the distribution of attacks across different protocols on each cloud. Gaining insights into which protocol faces the highest number of attacks can be crucial in identifying the security measures implemented by these service providers.

As previously noted, two honeypots were deployed on each cloud provider, one of which supported protocols other than SSH. While it might initially seem insignificant to compare SSH results with those of other protocols, because SSH was supported by all honeypots, it is still meaningful to compare the proportions of attacks on each protocol between the different cloud providers. This comparison can yield several insights into each provider's security landscape, as we will explore shortly.

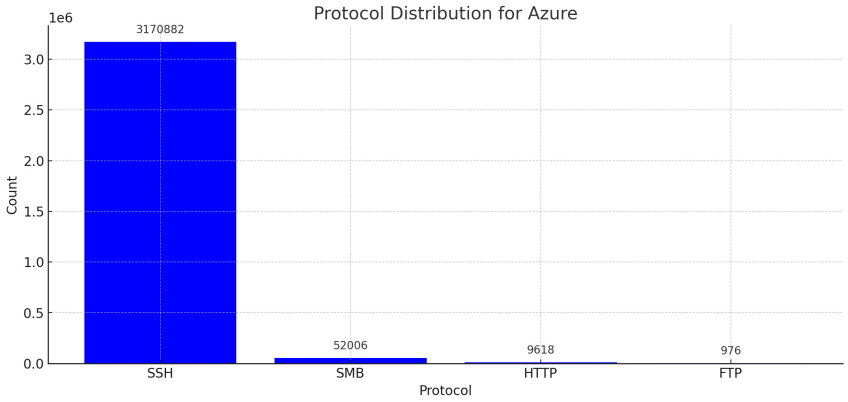

Figure 5.1: Azure's most attacked protocols distribution

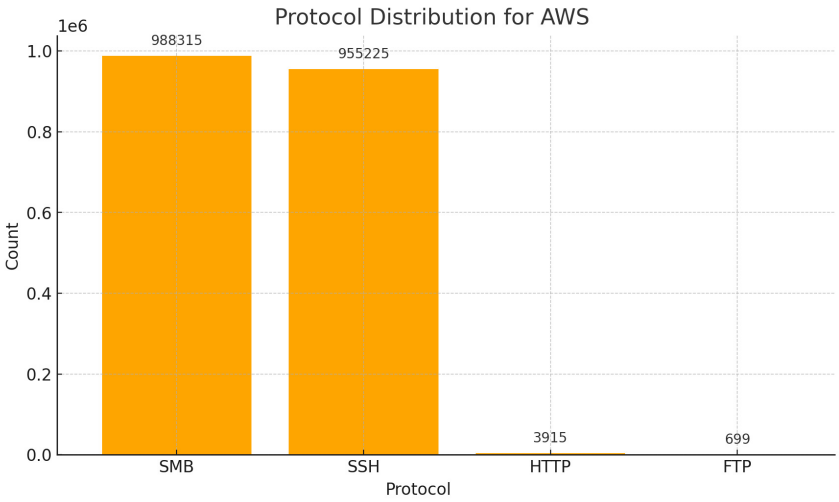

Figure 5.2: AWS's most attacked protocols distribution

The data in Fig. 5.1 and 5.2 reveal distinct patterns in attack targets for Azure and AWS. Most attacks on Azure are primarily focused on SSH, while AWS experiences a marginally higher frequency of SMB attacks than SSH attacks.

This notable difference suggests the presence of robust security measures in Azure, which effectively deter a significant proportion of adversarial traffic. Interestingly, this trend contradicts the expectation that Azure, a Microsoft product, would endure more SMB attacks because SMB is mostly associated with Windows systems.

This leads us to assume that Azure's firewalls may be particularly calibrated to block SMB connections due to their close association with Windows, while AWS's firewalls may be more proficient at filtering out SSH connections than SMB ones.

Findings

These observations shows a significant dependency on the type of attacked protocol. Therefore, we reject hypothesis *H3*, which initially suggested a lack of such dependency.

Chapter 8 will comprehensively analyze the varying attack patterns observed for each protocol on individual providers.

6 Attacker Profiling

To better understand the diverse nature of attackers, we have established various categories that capture distinct attack patterns. This classification extends beyond the scope of Agrawal's 2022 study [21], which focused on the number of targeted honeypots and the involved services. Our methodology also incorporates the count of cloud providers targeted by each attacker, thus offering a more comprehensive view of their behavior.

6.1 Defining Attackers' Profiles

1. **Singular Cloud Strike** - Attacks only one honeypot and only one service within one cloud provider (CP = 1, H = 1, S = 1)

2. **Solo Cloud Intruder** - Attacks only one honeypot and more than one service within one cloud provider (CP = 1, H = 1, S > 1)

3. **Single-Provider Cloud Blitz** - Attacks only one cloud provider and more than one honeypot and service (CP = 1, H > 1, S > 1)

4. **Multi-Target Cloud Assault** - Attacks more than one honeypot and only one service per honeypot within one cloud provider (CP = 1, H > 1, S = 1)

5. **Rogue Cloud Invasion** - Attacks more than one cloud provider and more than one service per cloud provider (CP > 1, H = 1, S > 1)

6. **Multi-Provider Cloud Raid** - Attacks more than one cloud provider and more than one honeypot per provider, and only one service per honeypot (CP > 1, H > 1, S = 1)

7. **Total Cloud Onslaught** - Attacks more than one cloud provider and more than one honeypot per provider, and more than one service per honeypot (CP > 1, H > 1, S > 1)

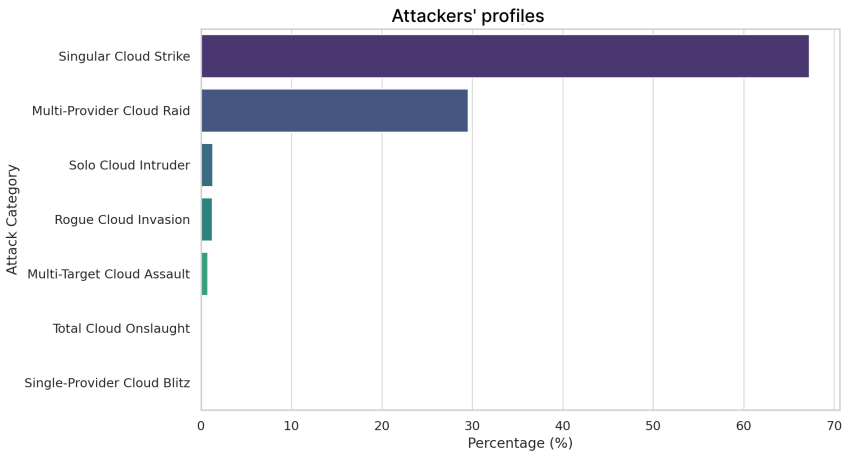

Figure 6.1: Number of attackers by category

Table 2: Distribution of Total Attacks

Profile	**Percentage**
"Singular Cloud Strike"	67.2%
"Multi-Provider Cloud Raid"	29.8%
"Solo Cloud Intruder"	1.2%
"Rogue Cloud Invasion"	1.1%
"Multi-Target Cloud Assault"	0.7%
"Total Cloud Onslaught"	0.01%
"Single-Provider Cloud Blitz"	0.004%

We initiate our analysis by examining the distribution of attackers' categories across both cloud providers. in Figure 6.1 and Table 2. Notably, 67.2% of attackers are directed toward a single cloud and service, while a significant 29.8% of attackers extend their activities to both clouds, targeting more than one honeypot within each provider. While the other categories are only a small fraction of the total attackers, their importance is foundational, as they represent the most powerful and dangerous attackers.

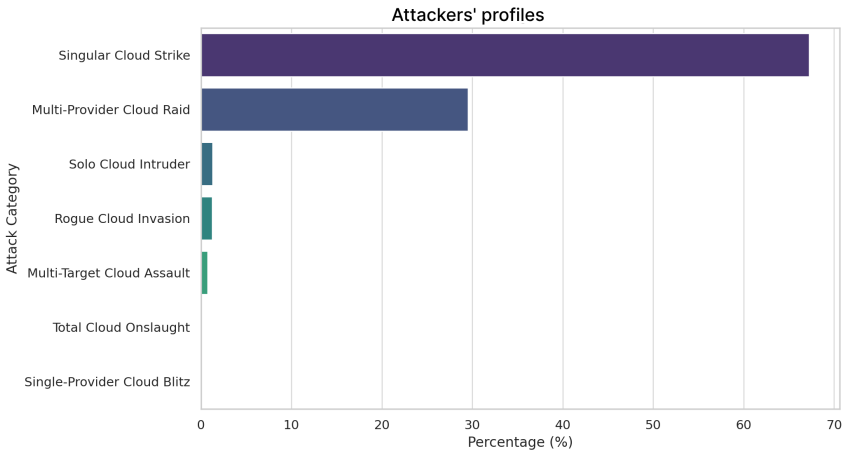

Figure 6.2: Number of attackers by category - Azure

Table 3: Distribution of Azure Attacks

Profile	**Percentage**
"Singular Cloud Strike"	66.7%
"Multi-Provider Cloud Raid"	30.6%
"Solo Cloud Intruder"	0.9%
"Rogue Cloud Invasion"	1.1%
"Multi-Target Cloud Assault"	1.1%
"Total Cloud Onslaught"	0.02%
"Single-Provider Cloud Blitz"	0.02%

The distribution of attackers on Azure, as in Figure 6.2 and detailed in the corresponding table, shows a significant majority (66.7%) of the attackers are classified as "Singular Cloud Strike". This category includes attackers who have targeted a single cloud provider and have restricted their activities to a single honeypot and service.

Moreover, a notable proportion (30.6%) of attackers fall under the "Multi-Provider Cloud Raid" category, showing that these attackers have launched assaults on multiple cloud providers.

Only a small fraction of attackers are classified as Solo Cloud Intruder" (0.9%), Rogue Cloud Invasion" (1.1%), Multi-Target Cloud Assault" (1.1%), Total Cloud Onslaught" (0.02%), and "Single-Provider Cloud Blitz" (0.02%).

Categories Distribution - AWS

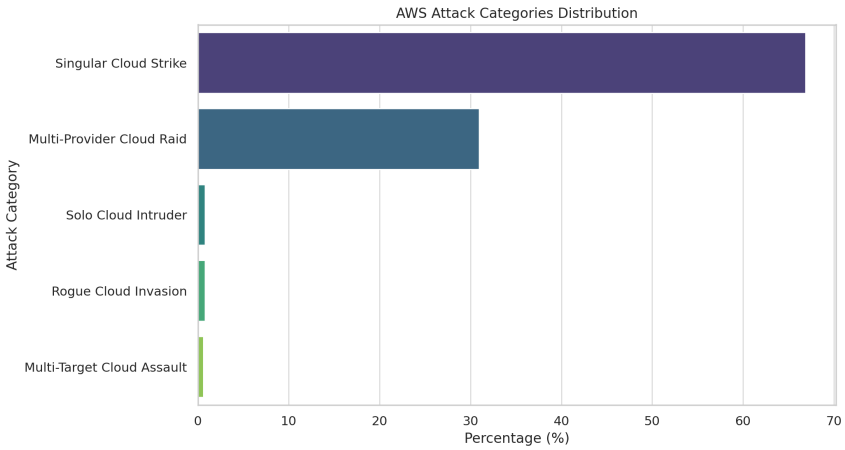

Figure 6.3: Number of attackers by category -AWS

Table 4: Distribution of AWS Attacks

Attack Type	**Percentage of Attacks**
"Singular Cloud Strike"	67.8%
"Multi-Provider Cloud Raid"	30.0%
"Solo Cloud Intruder"	1.1%
"Rogue Cloud Invasion"	0.9%
"Multi-Target Cloud Assault"	0.2%
"Total Cloud Onslaught"	0.01%
"Single-Provider Cloud Blitz"	0%

Similar to Azure, Fig. 6.3 shows the distribution of attacks on AWS, with a significant majority (67.8%) classified as "Singular Cloud Strike".

The "Multi-Provider Cloud Raid" profile also represents a significant proportion (30.0%) of AWS attackers. This shows that several attackers target more than one cloud provider.

The remaining categories, including "Solo Cloud Intruder" (1.1%), "Rogue Cloud Invasion" (0.9%), and "Multi-Target Cloud Assault" (0.2%), make up only a small fraction of AWS attacks. Interestingly, the "Total Cloud Onslaught" profile is even smaller (0.01%), and the "Single-Provider Cloud Blitz" profile was not found in AWS.

Findings

On both cloud providers, 67% of attackers, target a single honeypot and restrict their focus to one service within a singular cloud provider. However, a significant minority, around 30%, shows more diverse strategies, attacking multiple cloud providers and honeypots within each provider.

The 0.1% of the attacks show an even more extensive reach, attacking multiple cloud providers, multiple honeypots within each provider, and multiple services. This level of multi-vector attack was used by only two identified attackers - one originating from Hong Kong and the other from China.

As a result of this analysis, hypothesis H5, which assumes that attackers can be classified by their behavior, has been confirmed.

6.2 Skill and Strategy: Run Commands

In this section, we aim to analyze the attackers' tactics by examining the most frequent commands for both cloud platforms and then comparing the results.

6.2.1 Commands Timing and Origins

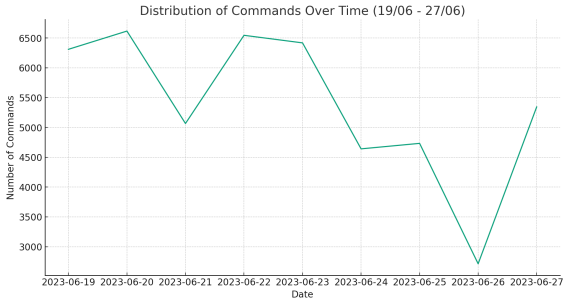

Figure 6.4: Distribution of commands over time

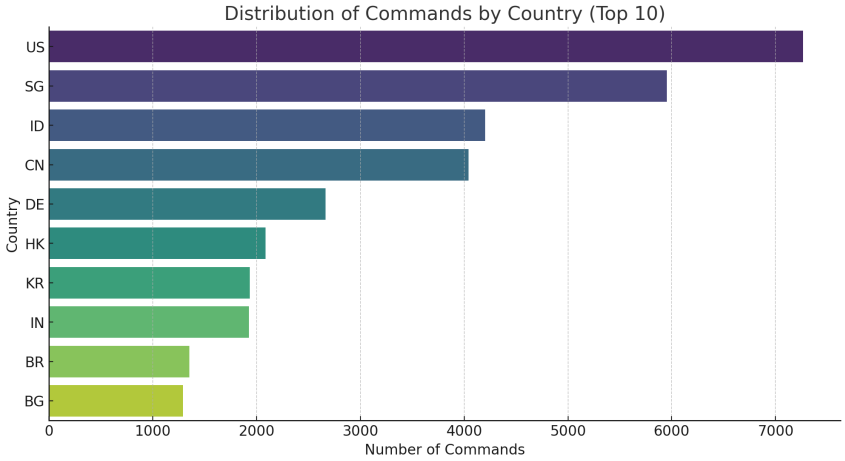

Figure 6.5: Distribution of commands by country

As expected, the United States and Singapore lead in executing the most commands and carrying out the highest number of attacks. Fig.6.4 shows a noticeable decline in overall commands towards the end of the week.

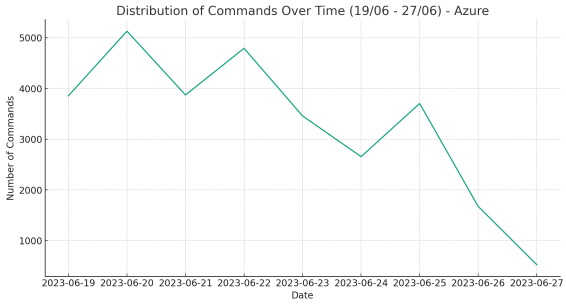

Figure 6.6: Distribution of commands over time - Azure

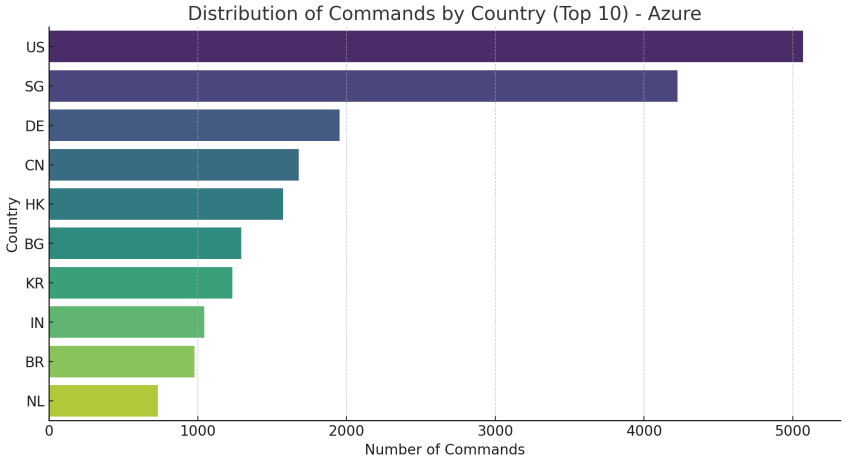

Figure 6.7: Distribution of commands by country - Azure

The Azure patterns align with the overall results observed previously, which is to be expected due to the higher number of attacks targeted toward this provider.

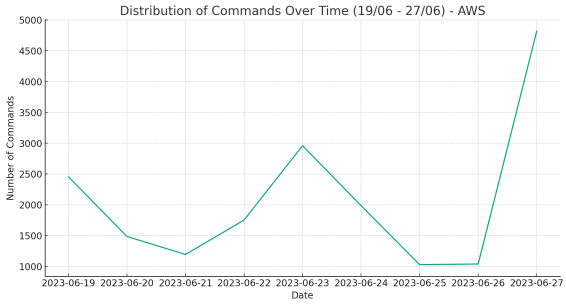

Figure 6.8: Distribution of commands over time - AWS

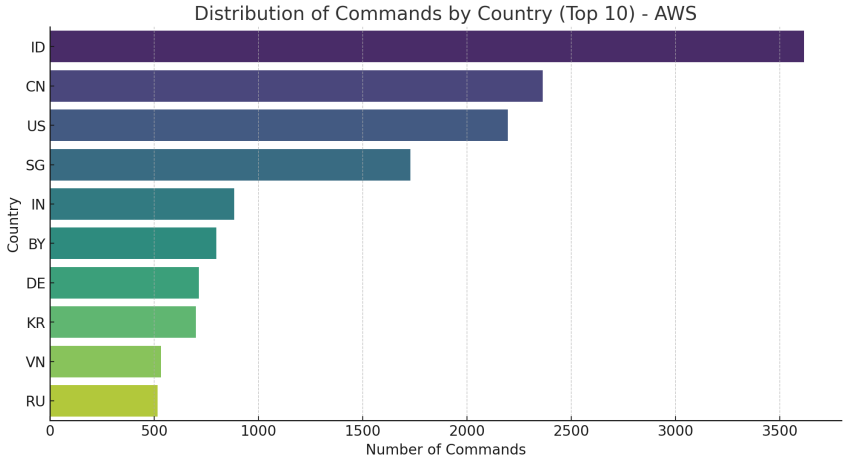

Figure 6.9: Distribution of commands by country - AWS

Nevertheless, we observe a distinct trend on AWS. The number of executed commands dramatically increases, marking a stark contrast to the plummeting numbers on Azure. Furthermore, Indonesia now emerges as the predominant source of most commands. This is consistent with the findings of our previous analysis that concluded Indonesia has repeatedly been targeting AWS, most likely with a series of commands.

6.2.2 Commands Categories

Similar to the methodology used for attackers' categorization, we have systematically sorted a total of 395,753 commands into various categories. This comprehensive classification not only presents an organized overview of the commands but also provides an insightful understanding of the attacker's intentions.

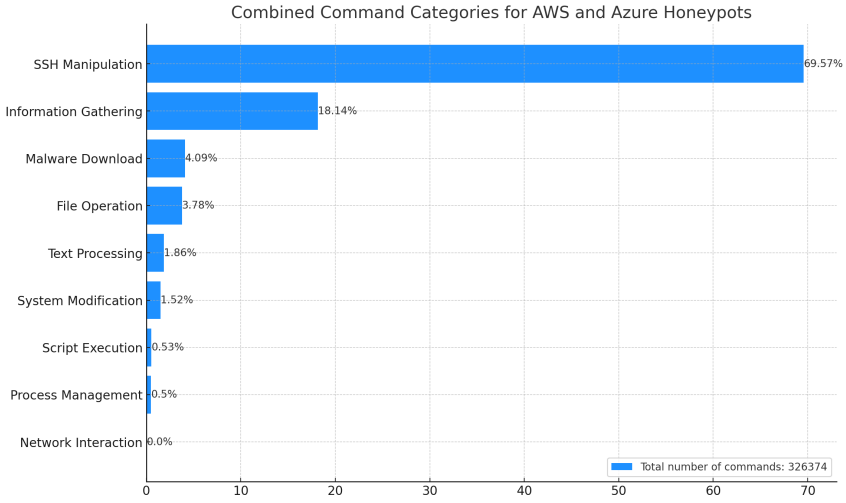

Figure 6.10: Commands Categories

As shown in Fig. 6.10, most of the commands fall within the SSH Manipulation category, indicating that compromising the SSH service is a primary objective for attacks.

We shall now examine each cloud provider individually to understand the differences

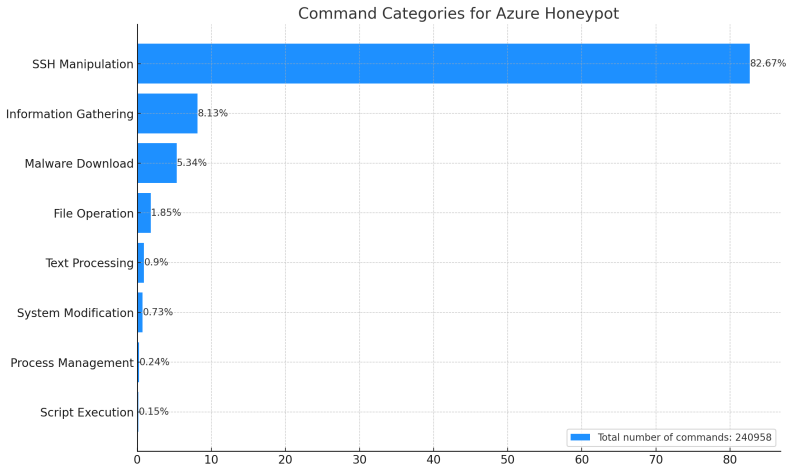

Figure 6.11: Commands Categories - Azure

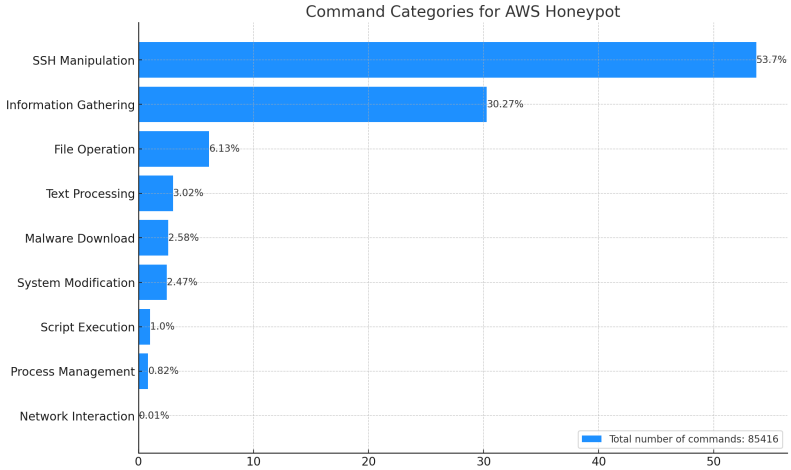

Figure 6.12: Command Categories - AWS

Fig. 6.11 and 6.12 presents the distribution of command categories on Azure and AWS platforms respectively. A notable 82.67% of commands on Azure fall within the SSH manipulation category, in contrast to AWS, where this category only accounts for 53.7%. Interestingly, the proportion of commands classified as information gathering is substantially higher on AWS (30.27%), than in Azure (8.13%). This pattern suggests that attackers on AWS are more prone to seek information about the machine than on Azure.. Additionally, it is worth mentioning that other categories like Malware Download, File Operation, Text Processing, System Modification, and Process Management make up a minor percentage of the total commands on both platforms.

Category	Description	Example Commands
Malware Download	Commands that appear to download files, that may reveal an attempt to install malware or other unwanted software.	wget, curl, scp
Information Gathering	Commands that retrieve system information or check system status. These might be used for a better understanding of the system.	uname, lscpu, cat /etc/*-release, ifconfig, ip addr, top, crontab -l, cat /proc/cpuinfo
SSH Manipulation	Commands that involve SSH or .ssh, indicating an attempt to manipulate SSH keys or settings.	Any command containing .ssh
System Modification	Commands that change system settings or files, potentially indicating an attempt to gain further control over the system or prepare for additional actions.	chattr, chmod, chown, useradd, adduser, passwd
Process Management	Commands that manage processes, potentially indicating an attempt to control what is running on the system.	ps, kill, pkill, bg, fg, nohup
File Operation	Commands that manipulate files or directories, potentially indicating an attempt to change the system's file structure or access specific files.	ls, cd, rm, mv, cp, cat, touch, mkdir
Network Interaction	Commands that interact with the network, potentially indicating an attempt to communicate with other systems, access network resources, or perform network reconnaissance.	ping, netstat, nmap, ssh, telnet
Package Management	Commands that involve a package manager, potentially indicating an attempt to install or remove software packages.	apt-get, yum, dnf, pacman
Script Execution	Commands that execute scripts, potentially indicating an attempt to perform complex actions or automate tasks.	sh, bash, python, perl
Text Processing	Commands that process text, potentially indicating an attempt to manipulate text files or parse command output.	grep, awk, sed, cut, sort, wc

6.3 Unveiling Linux's Role in Cyber Attacks

In order to shed light on Linux's function in cyber incidents, it is necessary to first position Linux within the larger device landscape by comparing its adoption rate with that of Windows and macOS. Examining relevant data provides a fundamental perspective on Linux's role in this context.

Linux's footprint in the desktop/laptop operating system market is smaller compared to Windows and macOS. According to global statistics provided Windows is leading this segment, covering approximately 68.15% of the market share in 2023. MacOS followed next with about 21.38%, while Linux constituted a mere 3.08% [22].

However, interpreting these figures as indicative of Linux's minimal presence would be a gross oversimplification. Indeed, Linux's sphere of influence extends significantly beyond personal computing. It is the underlying architecture for most server environments and a notable percentage of all Internet of Things (IoT) devices. As a matter of fact, a report from Eclipse Foundation in 2020 has shown that 43% of IoT developers prefer Linux for their IoT devices [23].

This predominance is largely attributed to its open-source nature, flexibility, and robust security features. Yet, these qualities have also made Linux a significant player in the cyber threat landscape. This is particularly true given that many exploitation tools have been exclusively developed for this platform.

6.4 OS Detection: SSH Clients

In order to detect the attacker's OS, the SSH client version was used. As a matter of fact, the SSH client versions can provide valuable insights into the OS platforms used to attack the honeypot. To achieve this, a classification approach was implemented to categorize SSH client versions into different operating systems as shown in 6.13. The approach involved examining specific keywords within the client versions' strings and associating them with either Windows or Linux OS. If the keywords matched, the SSH clients were classified accordingly. Specifically, SSH client strings that included OpenSSH, libSSH, or Go were classified as Linux, which included PuttY, and Windows. The clients were classified as 'Unknown' if no matches were found.

Figure 6.13 presents the analysis previously mentioned. The SSH clients are sorted in descending order based on the frequency of usage, making *libssh* the most frequently used. This substantial usage of libssh clients could be due to automated scripts operating as part of a larger botnet structure.

SSH Version	OS
SSH-2.0-libssh_0.9.6	Linux (libssh)
SSH-2.0-Go	Linux
SSH-2.0-libssh2_1.4.3	Linux (libssh2)
SSH-2.0-PUTTY	Windows
SSH-2.0-libssh-0.2	Linux
SSH-2.0-libssh-0.6.3	Linux
SSH-2.0-OpenSSH_7.9p1	Linux
SSH-2.0-OpenSSH_7.4	Linux
SSH-2.0-libssh_0.9.5	Linux
SSH-2.0-ZGrab ZGrab SSH Survey	Linux
GET / HTTP/1.1	Unknown
SSH-2.0-OpenSSH_7.9p1 Raspbian-10+deb10u2+rpt1	Linux
SSH-2.0-OpenSSH_7.9p1 Raspbian-10+deb10u2	Linux
SSH-2.0-OpenSSH-keyscan	Linux
SSH-2.0-PuTTY_Release_0.63	Windows
SSH-2.0-libssh_0.7.4	Linux
SSH-2.0-PuTTY_Release_0.62	Windows
SSH-2.0-OpenSSH_5.3	Linux
SSH-2.0-paramiko_2.1.3	Linux
SSH-2.0-paramiko_1.8.1	Linux

Figure 6.13: OS associated with each SSH Client

SSH Client Name
\x16\x03\x01\x00\xf2\x01\x00\x00\xee\x03\x03z-n\x1B\xb3\x1F\x89<L>\x1Fa^\x94\xe7\xx8e\xf6 \xb4R]\xbe\x91\x13\x95\x1B\xbd;T\x00\x00\x82\xc00\xc0,\xc0(\xc0$\xc0\x14\xc0
\x16\x03\x01\x01\t\x01\x00\x01\x05\x03\x03\xd7`\xbc\x82\xd9\xfb\xc5\xc5t\xe8\xa3C\xb1\xaeJ\ \xd8\x14\xfc\x8d\xca\xf7\x0FB\xd6\xdfm\x1D\xd07\xc75kuL]wMV][5\xc7\xf5P\x07wN\x00&\xc0/^
GET /freepbx/recordings/index.php HTTP/1.1
\x16\x03\x01\x00\xac\x01\x00\x00\xa8\x03\x03Ĵhe\xe4h\xdbI"\x8b3\x88\x18\xb2\xba3\xd2Z'\xfd'
\x16\x03\x01\x01\t\x01\x00\x01\x05\x03\x03\xb2h\xe7jw\x89\xb7\xf1OM:\x03I\x8ao\xc4\xea\xd(%\x98\xea$I\x88\x89\xbb\x97\x00\t\x87\xc2s\xe2Q\xc0\xb3\xf8
\xe0Cookie: mstshash=Administ
\xe0Cookie: mstshash=hello
\xe0Cookie: mstshash=Test
\xe0Cookie: mstshash=Domain
fox a 1 -1 fox hello
POST /ipp HTTP/1.1

Figure 6.14: SSH client strings injections

Throughout this analysis, we noted intriguing SSH client names, including unusual instances like *GET/ HTTP/1.1*. We uncovered additional instances exhibiting similar patterns, as illustrated in Fig. 6.14. These unconventional client names most likely are injection attempts forged by attackers. Since these strings do not correspond to any SSH client they do not convey information about the attacker's OS. Thus, they have been categorized as unknown.

Particularly noteworthy is the instance *GET /freepbx/recordings/index.php HTTP/1.1*. This instance refers to a vulnerability in FreePBX, a communication software, which bypasses authentication [24]. Some other instances within the table seem to signify encrypted network traffic, possibly TLS. The rationale for using HTTP requests as SSH clients remains uncertain.

Our attention was also drawn to strings like *xe0Cookie: mstshash=Administ* [25],

and others where 'Administ' is supplanted by different strings. These attacks were recognized as RDP BlueKeep Denial of Service [26] attacks and were only detected on Azure.

6.5 OS Detection: Clouds compared

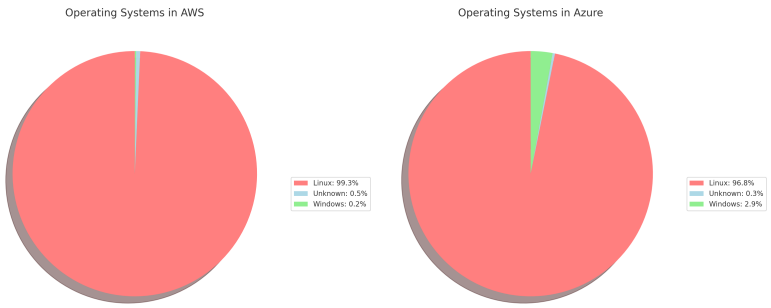

Figure 6.15: OS Fingerprint distribution

Fig. 6.15 shows that AWS mostly targets Linux nodes, while Azure shows a minor involvement with Windows nodes (2.9%). Furthermore, we observe a notable presence of unidentified clients, corresponding to the injection of strings we previously encountered.

Findings

Our analysis clearly shows Linux's significant role in the attack landscape; thus, hypothesis *H4* is accepted. Furthermore, we found that Azure has been targeted by more Windows machines than AWS, suggesting that AWS may have additional security layers against SSH client injection strings.

6.6 Investigating the Role of TOR Nodes

The Onion Router (Tor) enables anonymous communication by directing internet traffic through a worldwide network of servers. While it is a vital tool for preserving privacy and avoiding censorship, it can also be exploited by malicious actors to hide their identities and location. Fig. 6.16 shows that TOR nodes constitute a minimal

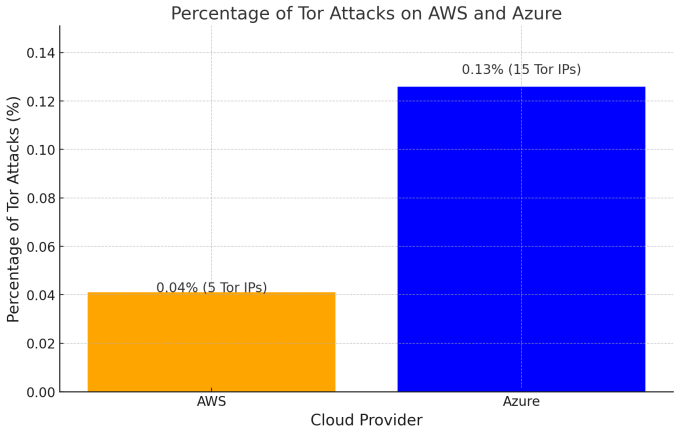

Figure 6.16: TOR nodes per Cloud

number of attackers. Out of a total of 20,250 attackers, only 20 were identified as using the TOR network. Interestingly, Azure had twice the number of TOR attackers compared to AWS. Several factors could contribute to this discrepancy, including varying security measures implemented on AWS and Azure platforms.

Findings

Based on the data, we can conclude that hypothesis $H5.3$ is proven as attacks originating from TOR nodes are not statistically significant, accounting for only 0.17% of attackers.

6.7 Skill and Strategy: Attack Duration and Attacker Expertise

One useful approach to classify attacks considers whether they are carried out manually or using automated tools. In order to distinguish between these two categories the duration of the SSH connection can be used. In this study, we will say that an attack is automatic if it lasts less than 2 minutes. If it lasts longer, we will call it a manual attack.

Automated Attacks

These attacks utilize automated scripts and botnets. These tools attack a number of targets simultaneously. These attacks are typically quick and tend to result in a shorter connection.

Manual Attacks

On the other hand, manual attacks are usually more sophisticated than automated ones. They manually connect to the system and explore the environment trying different attack vectors and adapting their strategies based on the responses from the system. This process is slower and results in a longer connection.

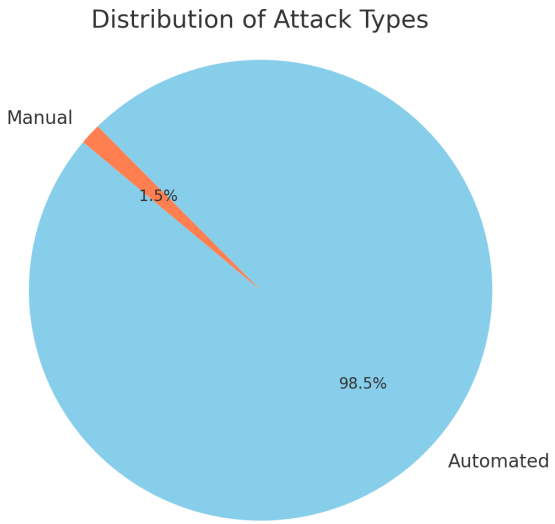

Figure 6.17: Attack categorization (using SSH Session Duration as discriminator)

Fig. 6.17 shows the distribution of the two types of attacks: automated and manual. The overwhelming majority (98.5%) of attacks are automated, while manual attacks constitute a small fraction (1.5%). This suggests that most attackers rely

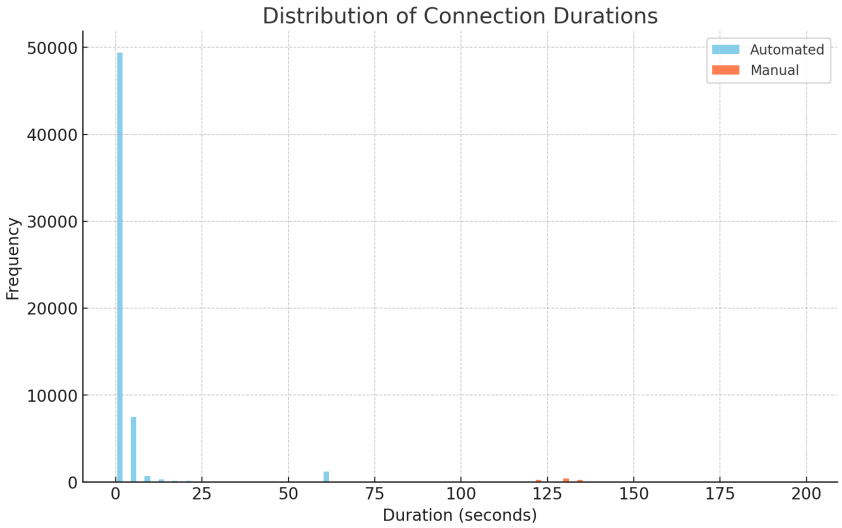

Figure 6.18: SSH Session duration distribution

on automated tools or scripts to carry out their attacks, which could be due to the operation of botnets.

As can be seen in Fig. 6.18, most attacks are characterized by very short session times. This strongly indicates that these attacks are likely carried out automatically, by scripts. However, there are also instances of longer session times, as confirmed by the findings presented in the pie chart above.

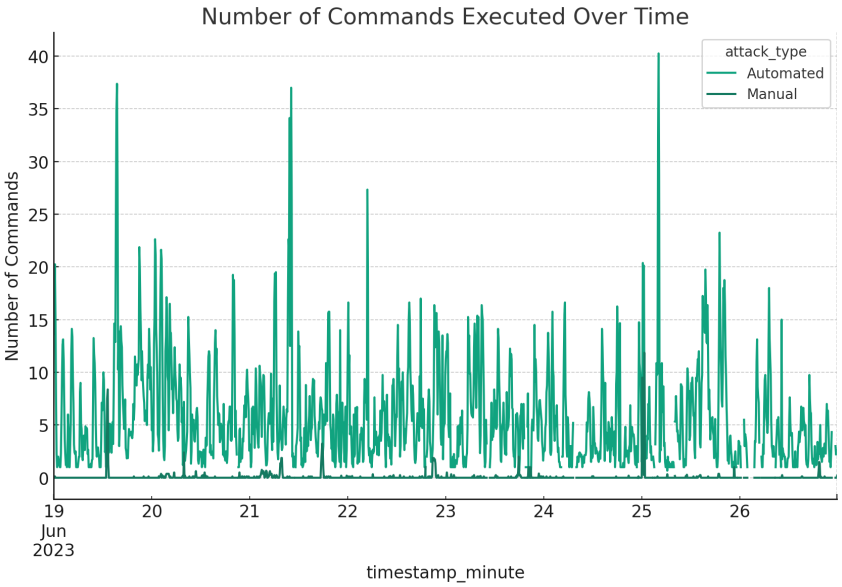

Figure 6.19: Run commands over time based on Attacker's category

In Figure 6.19, we can see that the number of commands executed in automated ones is higher than in manual attacks. There are some noticeable spikes in the number of automated commands, which could represent periods of increased attack activity.

Interestingly, as shown in Figure 6.19, the frequency of manual attacks—the dark-green segment at the bottom—appears relatively stable over time, with no fluctuations apart from an isolated spike on June 25th. This pattern could indicate that despite their smaller numbers, manual attackers have kept levels of activity throughout the analyzed period.

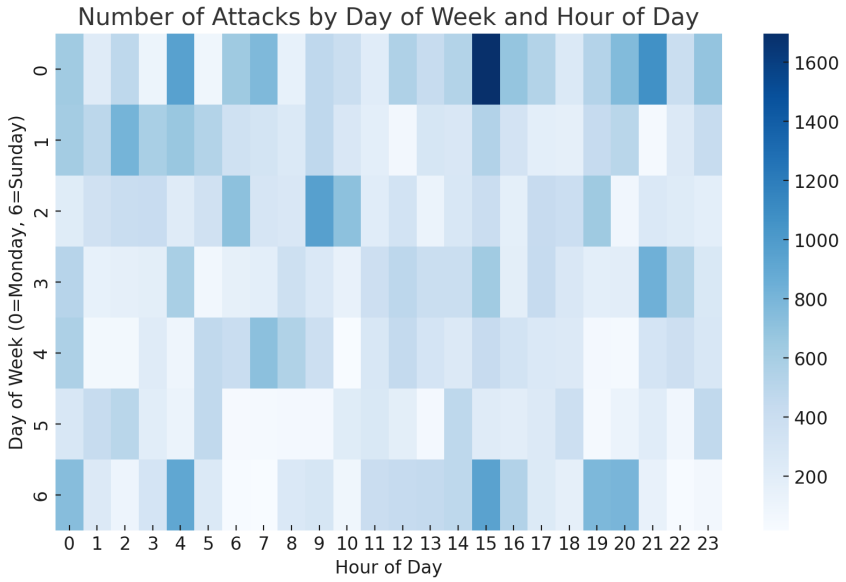

Figure 6.20: Heat-map of attacks over time

This heat map shows the number of attacks by day of the week and hour of the day and it reveals some interesting patterns. For instance, there appears to be a higher number of attacks at the beginning of each hour, as indicated by the darker vertical bands. These patterns could reflect the behavior of the attackers and the automated tools they use. For example, the concentration of attacks at the beginning of each hour could be due to automated scripts that are scheduled to run on the hour.

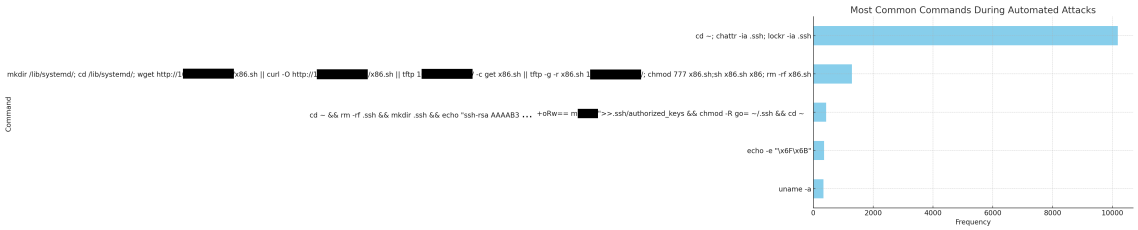

Figure 6.21: Top 5 automated attacks commands

Figure 6.21 shows the most common commands executed during automated attacks. The command *cd ; chattr -ia .ssh; lockr -ia .ssh* is the most frequently used, appearing more than 3000 times. It removes the immutable and append-only attributes from the .ssh directory, allowing the attacker to modify or delete SSH keys. The longest one involves downloading malware from an external server.

The prevalence of these commands in automated attacks suggests that many attackers are using similar tools or scripts.

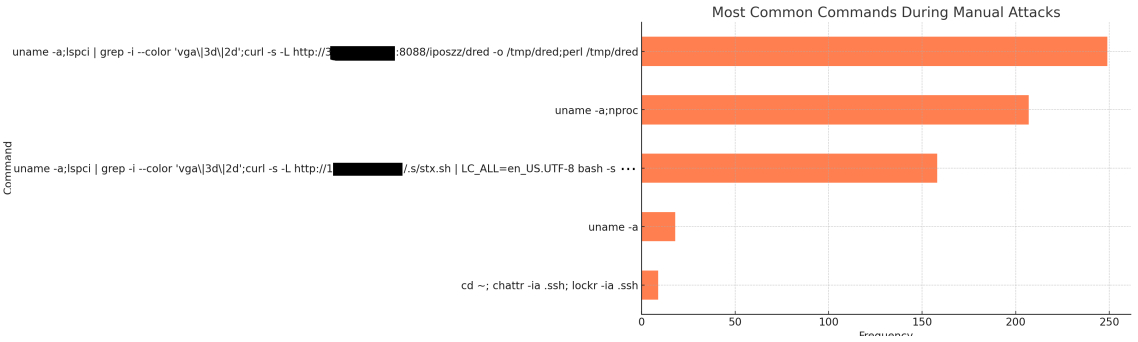

Figure 6.22: Top 5 manual attacks commands

Figure 6.22 shows the most common commands executed during manual attacks. The most frequent commands are those that retrieve system information (*uname -a;nproc, uname -a*) or download and execute scripts from external servers.

Findings

Our analysis found a significant difference in the connection durations between automated and manual attacks. Most automated attacks lasted less than a few seconds, while manual attacks could last several minutes or more. This finding aligns with our understanding of the behaviors of these two types of attackers.

In light of our findings, it's evident that the skill levels and resources needed for automated and manual attacks are notably different. Automated attacks mostly rely on a set of shared commands and may not need advanced skills aiming to attack as many systems as possible.

On the other hand, manual attackers, who have extended connections and a diverse repertoire of commands, appear more adaptable and tactical. These attackers may potentially modify their strategies based on the unique characteristics of the system they target, exhibiting a higher degree of skill and strategic planning.

These findings support hypotheses H5.1 and H5.2, showing that their skill level and available resources significantly influence the attackers' patterns. For example, the availability of a botnet could simplify the execution of widespread automated attacks.

7 Unraveling Discovered Malware: Origins and Types

7.1 Malware Origins Analysis (SSH)

In this section, we delve into the geographical origins of malware Transfers (File uploads and Downloads) observed on our SSH honeypots by analyzing the source IPs of attackers. This investigation can provide us with insights into the most active regions from where malware attacks originate. We utilized VirusTotal to analyze the malware samples [27].

SSH Malware Uploads vs Malware Downloads

In this context, the term ***Malware Upload*** refers to the execution of SSH commands that upload malware, generally with the intent of hijacking the SSH-authorized keys. Possessing these keys provides the attacker with a backdoor to the system facilitating their reentry into the compromised system.
In contrast, ***Malware Downloads*** directly downloads malware onto the machine through web requests using '*wget*' commands and 'curls' from the attackers. The malevolent files are downloaded from an external server, usually under the attacker's control.

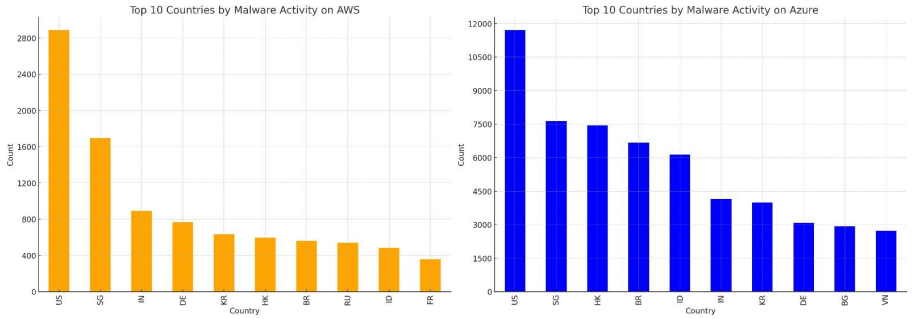

Figure 7.1: Number of Malware Transfers (Uploads & Downloads) by Country and honeypots' Data Center

The above bar chart shows the number of malware transfers organized on the geographical origins of the attackers, inferred by their IP addresses. Most of these transfers are initiated by United States-based IPs, followed by those originating from Singapore and Germany.

The relatively high frequency of malware transfers attributed to German-originating attacks may be explained by their geographical proximity to the Netherlands, where the honeypots are located. This geographical proximity may enhance the interaction between German attackers and these honeypots, therefore leading to an above-average rate of malware downloads from this region.

7.2 Malware Uploads Analysis

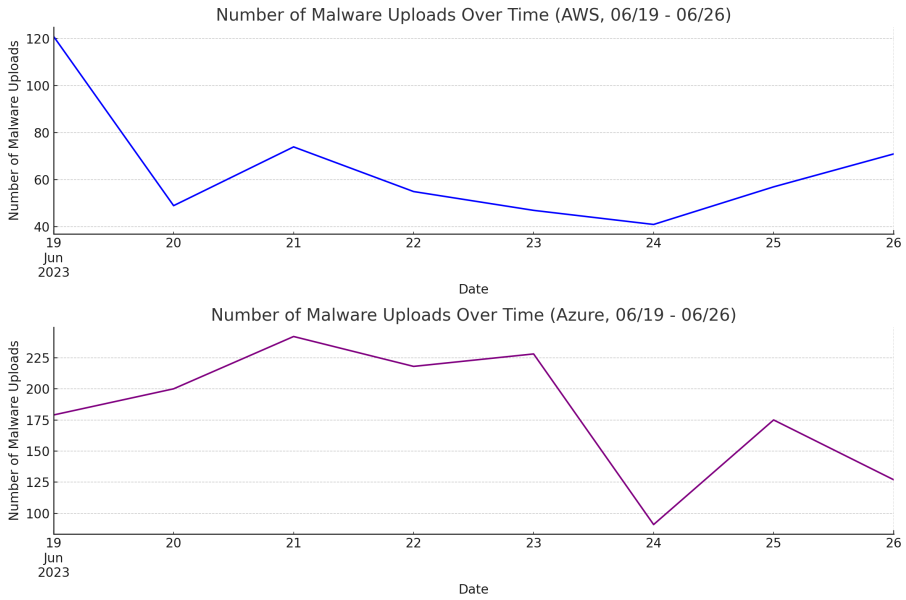

Figure 7.2: Number of Malware uploads over time

As we can observe from Fig. 7.2, the number of malware uploaded is higher in the case of the Azure honeypots. We can observe the following:

- 19/06 was the day with the most malware uploads on the AWS honeypots.
- On 21/06, there was an increase in malware uploads detected in the AWS honeypots, indicating a bounce-back trend. This day also had the most malware uploads on Azure's honeypots. This similar trend in both data centers probably suggests the existence of a common attack.
- On 24/06 there is a steep descent in malware uploads on both the datacenter suggesting that on this Saturday, automated attackers ceased activity

Let's now briefly focus on the principal aim of these uploads which is to hijack the SSH-authorized keys:

Interestingly, most (over 95%) of these malware uploads across both cloud providers upload the same SSH key. The global pattern observed in Fig. 7.3 suggests the existence of a botnet or multiple attack vectors that scan the network. Upon identifying a vulnerable (or brute-force susceptible) SSH client, the attacker attempts to incorporate the machine into the existing botnet. The recurrent use of the same SSH keys supports this hypothesis.

Figure 7.3: Top 5 Countries of Common Backdoor Key Uploads

Our previous analysis observed a pervasive pattern of the SSH backdoor key being uploaded to both AWS and Azure data centers. This key was not confined to any specific geographic location, but rather, was detected across numerous countries worldwide.

This global distribution, coupled with the consistent uploads to two distinct data centers, may imply the existence of a coordinated operation on a global scale. A plausible interpretation of this phenomenon could be the activities of a widespread botnet. Alternatively, this might indicate a common attack vector being exploited by multiple attackers worldwide.

While these interpretations are just our speculations, the data suggests that a coordinated effort is being made to exploit this particular SSH backdoor key across various geographic locations and infrastructure providers.

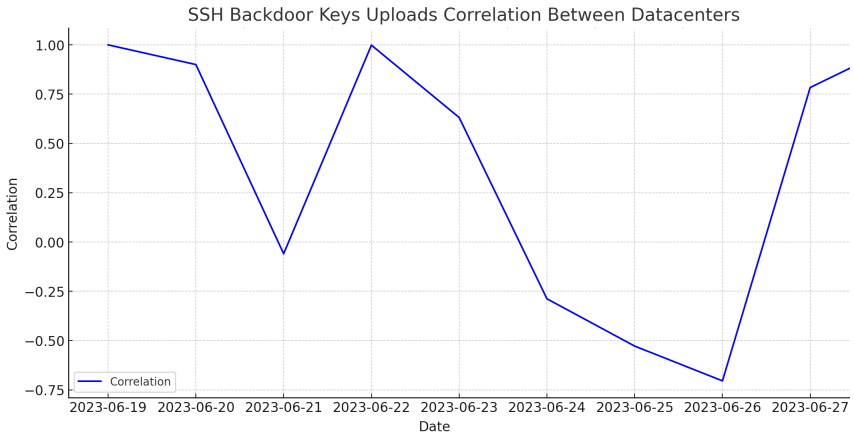

Figure 7.4: Common SSH backdoor key Correlation between the two data centers

In extending our analysis of the SSH backdoor key's uploads across AWS and Azure honeypots, we focus on the correlation between these uploads over our week-time data. This strong correlation, especially considering the global distribution of these uploads, further confirms the hypothesis of a coordinated operation. The upload patterns hint at a network of actors, such as a globally distributed botnet, systematically uploading the same SSH backdoor key to these data centers.

7.3 Malware Downloads Analysis

Azure

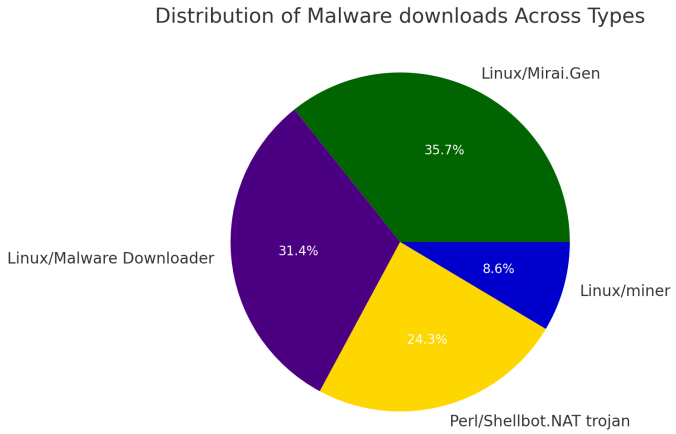

Figure 7.5: Percentage of Malware Types

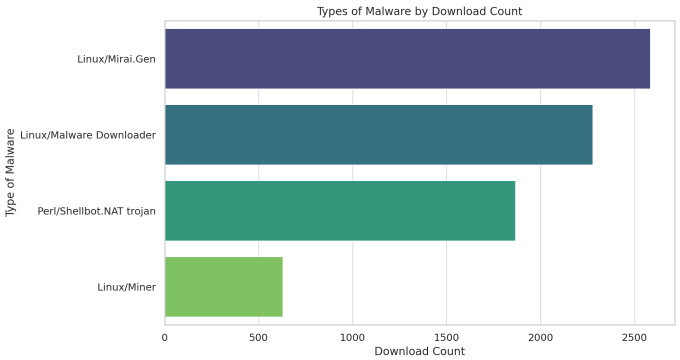

Figure 7.6: Distribution of Malware Types

These charts provide an overview of the types of malware Downloaded in the SSH honeypots by the attackers (from their servers), sorted by the download count. The Linux/Mirai.Gen (with 2500 copies), Linux/Malware Downloader (with 2000 copies), and Perl/Shellbot.NAT trojan (with 1500 copies) types, indicating that they are the most commonly downloaded malware types. In particular, Linux/Mirai.Gen shows the highest values (in copies downloaded), suggesting that this type of malware is a significant threat.

In conclusion, examining the data from our SSH Azure honeypots has shed light on several key findings. This study has suggested a possible link between geographical location and the cyber threats experienced. We will be conducting an analysis of this finding in Chapter 9. Additionally, our previous sections have identified a significant presence of libssh fingerprints in our dataset. This observation suggests

the existence of several custom attack scripts, with a substantial fraction possibly originating from Mirai or similar botnets.

AWS

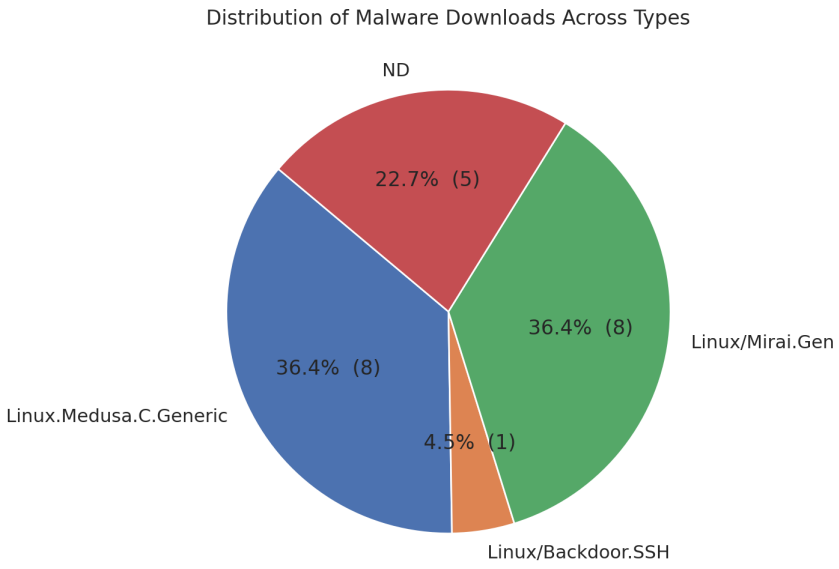

Figure 7.7: Distribution of Malware Types in Percentage

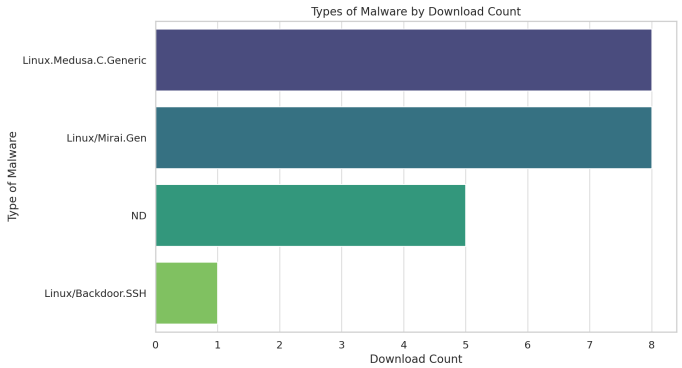

Figure 7.8: Distribution of Malware Types by Download Count

Fig. 7.8 confirms the existence of a higher volume of Malware Downloads on Azure than AWS. This suggests that attacks targeting AWS over SSH are not only fewer in number, as indicated by previous analyses but also download malicious software with a much lower frequency than those targeting Azure. Note: ND in the figure refers to malware instances whose types were not determined.

Interestingly, despite the lower number of malware downloads on AWS, there is a strong correlation between SSH key uploads in both data centers. This could suggest

that AWS may adopt more robust firewall protections against malicious downloads, leading to fewer successful malware downloads.
An alternative explanation could be that attackers using malware downloads prefer Azure servers.

7.4 Exploring Malware Hosting

Malware hosting refers to the servers or infrastructure attackers use to store and distribute malicious software. These servers are accessed via command-line interfaces such as SSH, where specific commands are used to download the malware onto target systems.

Due to the high volume of malware downloads observed on Azure's honeypots and the comparatively insignificant data obtained from AWS honeypots, as observed in the previous sub-chapter, we have narrowed our focus exclusively to Azure's malware downloads dataset. This approach ensures that our analysis remains robust, drawing from a richer data set, and provides more meaningful insights into the malware-hosting landscape.

Our analysis focuses on the origins of the malware servers which have been used to download the malware in the honeypots. The United States and China will emerge as significant contributors, hosting most of the malware. The distribution of malware types across these countries provides a unique insight into the prevalent threats in these regions.

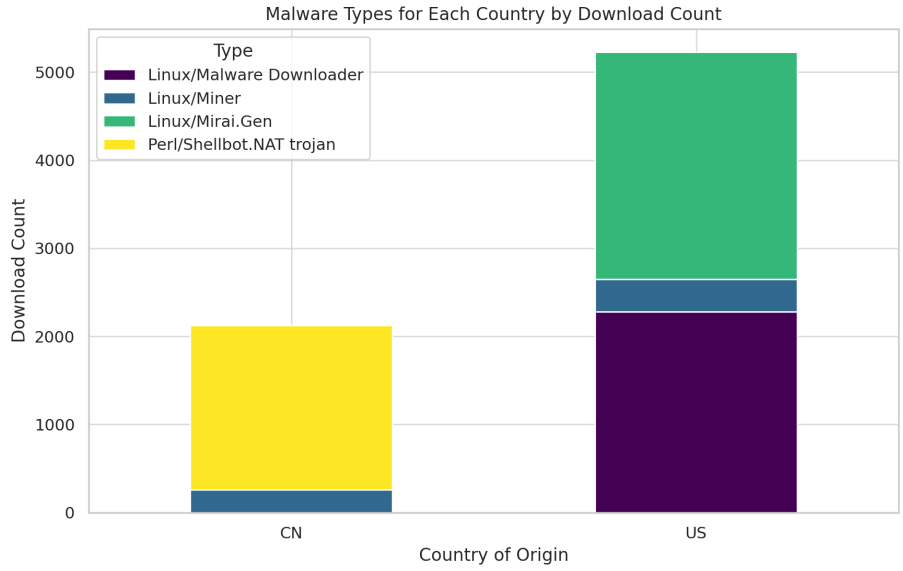

Figure 7.9: Malware Types for Each Hosting Country by Download Count (Azure)

Fig. 7.9 offers a breakdown of malware types for each originating server's country. Each color signifies a different type of malware.
From United States (US) Servers, we observe an important prevalence of Linux/Mirai.Gen and Linux/Malware Downloader malware types, where Linux/Mirai.Gen is dominant. These two types are the majority of malware downloads from the US, showing the country's role in the propagation of these specific malware types.

In contrast, the malware downloads from China's (CN) servers are mostly Perl/Shellbot.NAT

trojan and Linux/Miner, with Perl/Shellbot.NAT trojan is the predominant type. This suggests that this type of malware is particularly prevalent in China.

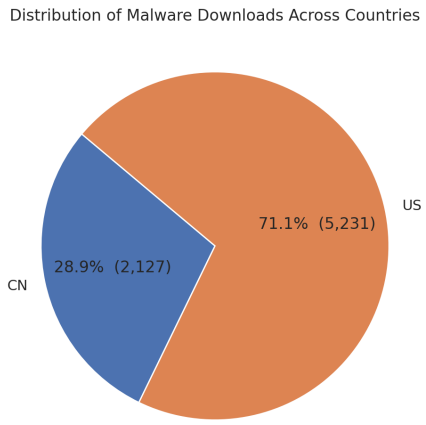

Figure 7.10: Malware Downloads Across Attackers' Database locations

The United States (US) and China (CN) stand out as the primary sources for malware hosting. The United States, in particular, accounts for most file downloads, again showing its dominant position in the malware landscape.

Findings

Our investigation, focused on Azure's honeypot malware downloads due to its high volume, provides significant insights.
The analysis of Azure's honeypot malware downloads highlights the fundamental role of the United States and China as significant sources of such malware. In particular, the United States is the leading contributor, accounting for most malware downloads. It is notable that specific types of malware, namely Linux/Mirai. Gen and Linux/Malware Downloader are mostly associated with the United States. In contrast, China shows a higher prevalence of Perl/Shellbot.NAT trojan and Linux/Miner.
The distribution of malware downloads across these countries provides a perspective into the regional threat landscape. It underscores the varying threat profiles of different regions, emphasizing the need for region-specific malware detection and prevention strategies.

7.5 Malware Origins Analysis (SMB)

It was apparent that SMB honeypots presented an interesting opportunity for potential attackers. From their perspective, they encountered an SMB file server accommodating not only the latest versions of Samba, namely SMBv2, and SMBv3, but also the more vulnerable SMBv1. This older version has been a frequent target of attacks, particularly against outdated Microsoft Windows and Windows Server systems. Given this context, one would expect to find many Windows-specific malware.

As stated earlier, half of the deployed honeypots were designed to support the SMB protocol. While the number of honeypots might seem significant at first glance, we argue that its relevance is minimal in this context. Instead of relying on raw numbers, we opt for a more proportion-focused approach, utilizing graphical representations for better clarity.

Specifically, we aim to carry out an analysis focusing on the geographical distribution of originating IP addresses to establish a meaningful comparison between SSH and SMB attackers. This method allows us to visualize and better understand the proportional differences and similarities between these two types of attacks.

Azure

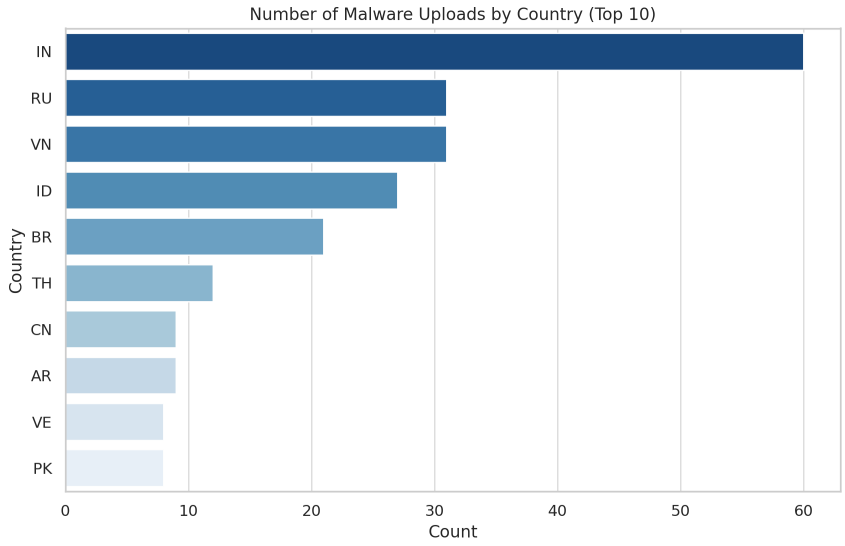

Figure 7.11: Number of Malware Uploads by Country - Azure (SMB)

The chart in Fig. 7.11 provides insight into the top 10 countries accounting for the most malware uploads through the SMB protocol. A comparison of this data with the SSH Malware Transfers data reveals a fundamental difference in the primary sources of these attacks.

Although SSH malware transfers displayed the United States, Singapore, and China as the main origins, SMB data, on the other hand, introduces an entirely different set of countries into the equation. It is worth noting that countries like India and Russia emerge as prominent contributors to SMB malware uploads.

This variation in the data supports the idea that the geographical origins of attacks can significantly differ based on the protocol that is being exploited.

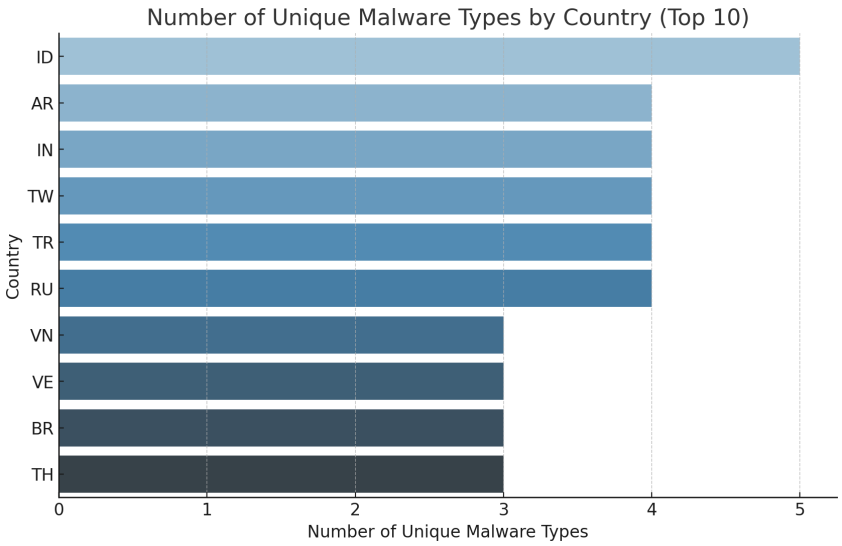

Figure 7.12: Number of unique Malware Types by Top 10 Countries (SMB - Azure)

Fig. 7.12 shows the distribution of unique malware types, divided by the top 10 countries of origin (from attackers who uploaded the malware). Notably, each country shows at least three distinct malware types, with Indonesia presenting the highest variety. This pattern mirrors the distribution observed in the total SMB malware upload count, with approximately 60 instances originating from India, implying an average ratio of roughly 12 iterations per malware type. This trend, even if less pronounced, is also present in the same data for the other countries.

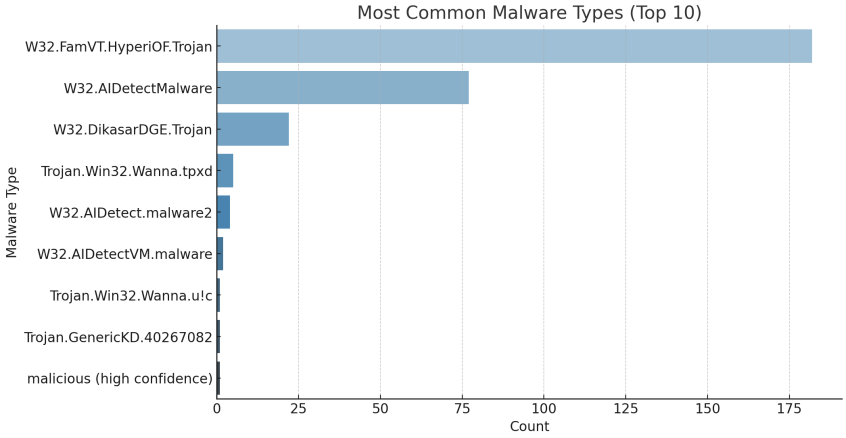

Figure 7.13: Top 10 Malware Types distributed by uploaded variants - Azure (SMB)

The above bar chart shows the 10 most prevalent types of SMB-uploaded malware. Each type can be associated with numerous uploads, but is rather interesting that the first nine malware types are either variants or slight modifications of the WannaCry malware. This is coherent with our anticipatory assumption concerning the potential extensive exploitation of SMBv1, which we will explore in a subsequent discussion.

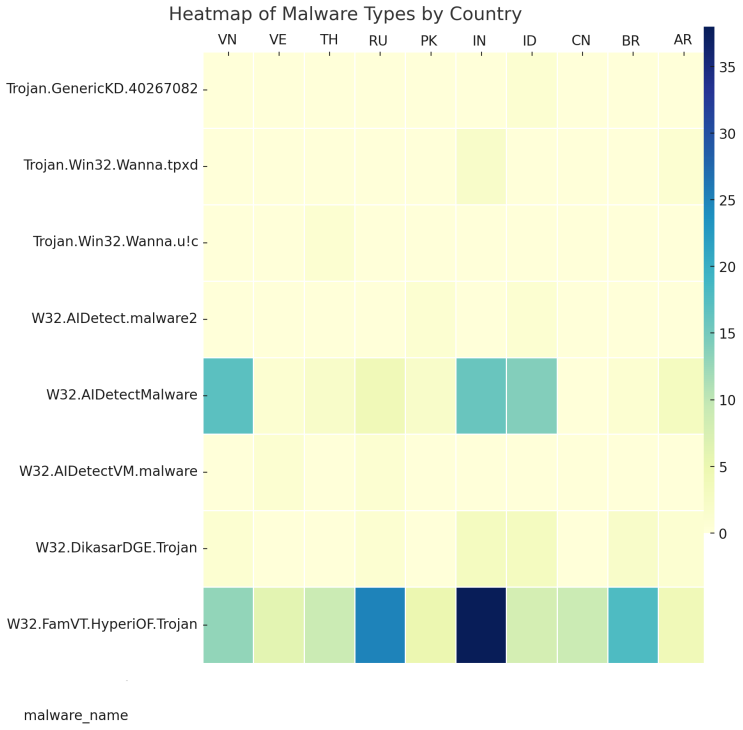

Figure 7.14: Heatmap of Malware Types by country - Azure (SMB)

The preceding heatmap shows the correlation between the top 10 malware types and their countries of origin. This visual representation simplifies an understanding of

the geographical distribution of each malware type. India, in particular, exhibits a darker hue, indicative of a higher count in malware uploads. Specifically, the WannaCry variant *W32.FarmVT.HyperOf.trojan* appears to have a significant presence in India, highlighting the country's prominent role in this context.

AWS

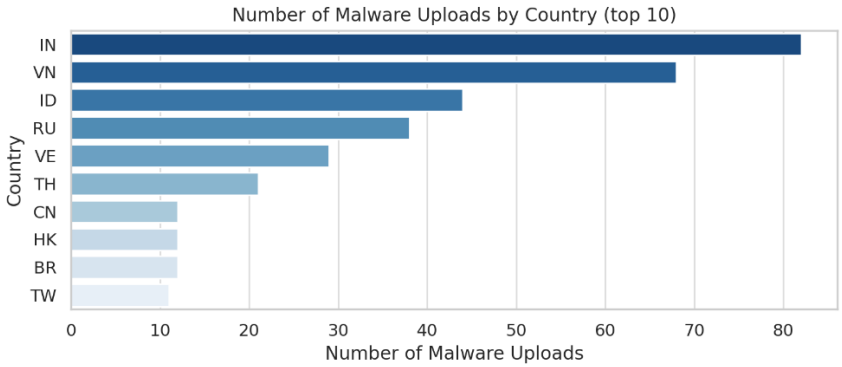

Figure 7.15: Number of Malware Uploads by country - AWS (SMB)

The bar chart in Fig. 7.15 indicates that India is the leading country in terms of malware uploads, with the highest figure exceeding 80 copies of malware. This pattern is also replicated in the Azure datacenter's honeypot that records SMB activity, which registered the most malware downloads over SMB originating from India, tallying up to 60 instances.

The following countries, ranked by the number of malware downloads over SMB, present a varied picture compared to the data observed in the Azure data center. While the second most active country in Azure's case was Russia, the AWS honeypot places Vietnam in this position, with more SMB malware uploads than previous observations.

Russia, nonetheless, remains a prominent player in terms of the number of malware uploaded over SMB, but, with a smaller proportion compared to the leading countries. Consequently, the analysis generates similar data to that obtained from the Azure honeypot, implying a certain level of independence in the country of origin for SMB attackers across the two data centers.

The bar chart in Fig. 7.16 presented above shows the distribution of unique uploaded malware variants, divided according to their countries of origin. Notably, each country has been found to host at least three distinct malware types, with India showcasing the greatest diversity. This distribution reflects the overall count of malware uploads, with roughly 82 instances from India, yielding an average of

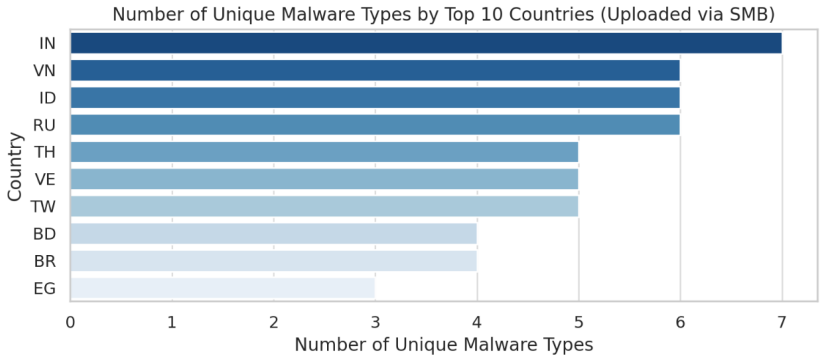

Figure 7.16: Number of unique Malware Types by Top 10 Countries - AWS (SMB)

approximately 11.7 instances per malware type.

Upon comparing this data with the statistics from the Azure honeypot, we notice a shift in the rankings. Vietnam emerges as the second most active country, displacing Russia from its earlier position. This alteration confirms the observation made in the previous bar chart, where many SMB malware uploads from Vietnam were witnessed. On the other hand, there is the same observed average number of instances per malware type (12) originating from India, suggesting common attackers in both datacenters honeypots.

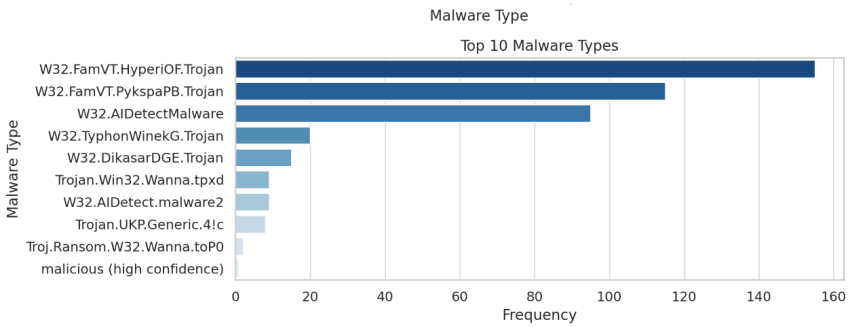

Figure 7.17: Top 10 Malware Types distributed by uploaded variants - SMB (AWS)

Looking at the bar chart in Fig. 7.17, we see that a few malware types clearly stand out. One type is much more common than the others, followed by nine, gradually becoming less frequent.

Delving further into the specifics, we discern that the most common malware types are variants or minor modifications of the *WannaCry* malware. This observation aligns with our previous supposition concerning the potential exploitation of the SMBv1 protocol, an aspect that will be examined in further depth in further analyses.

A comparative analysis of the SMB-uploaded malware types prevalent in AWS and Azure presents intriguing facts. Some malware types appear across both platforms, suggesting shared threats between the two cloud providers. However, the frequency of these malware types presents a marked variation, thereby showing distinctive threat landscapes for AWS and Azure.

Intriguingly, a new malware, *W32.FamVT.Pykspa.Trojan*, has emerged with multiple versions. This MS-Windows worm, known to propagate via Skype messaging, mapped drives, and network shares (that uses SMB), carries a backdoor that permits the execution of arbitrary commands by a remote attacker [28].

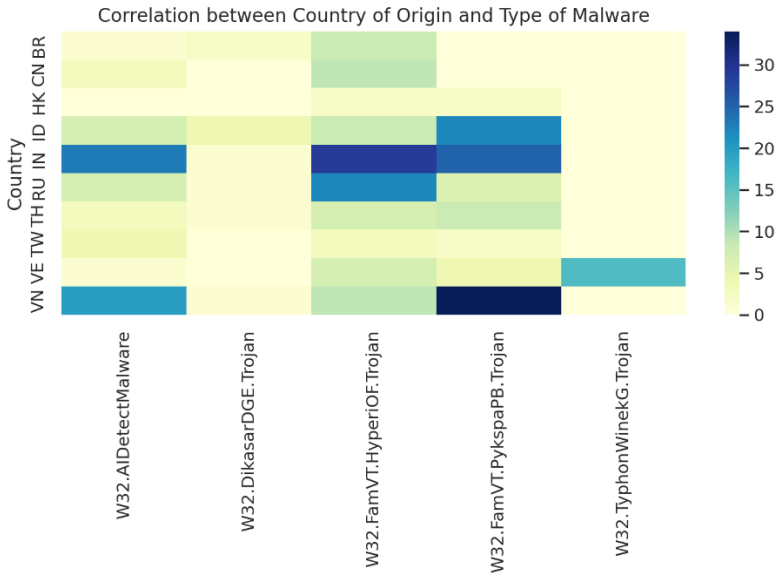

Figure 7.18: Heatmap of Malware Types by Country - AWS (SMB)

The heatmap is crucial in linking the emergence of the new malware variant *Pykspa* to the increase in attacker activities from Vietnam. The results from the heatmap show a strong correlation between the appearance of this worm variant and the surge in attacks from Vietnam, with over 30 copies being downloaded.

Interestingly, the heatmap also reveals that attackers from Indonesia and India have shown engagement with this particular malware family. Moreover, the presence of the *WannaCry* variant, *W32.FamVT.HyperiOF.Trojan*, which was commonly observed in the Azure data, is also reflected in the AWS honeypot data. This observation suggests a substantial overlap in the threat landscapes of the two major cloud platforms, with the addition of the new *Pykspa* malware in the AWS environment. While not robust, this observed trend could indicate stronger security measures being employed against SMB attacks within the Azure environment. Since both Azure and SMB are Microsoft products, there may be a natural focus on protecting against these types of intrusions. Alternatively, this trend might simply reflect a greater predilection among attackers for targeting AWS machines.

7.6 Unraveling Discovered Malware: Findings

The study reveals significant insights into malware attacks' origins, types, and objectives.

The geographical analysis of the malware transfers indicates a global reach, with significant hotspots including the United States, Singapore, and Hong Kong. Understanding these origins is an important step in comprehending the extent and scope of the attackers.

SSH

Regarding attack types, a key finding is the high volume of malware uploads with the primary objective of hijacking the SSH-authorized keys. This tactic provides the attacker a backdoor for reentry into the compromised system. The recurrence of the same SSH keys across several attacks suggests the potential existence of a botnet or multiple attack vectors scanning for vulnerable SSH clients.

The malware download analysis reveals specific *malware types, notably Linux/Mirai.Gen, Linux/Malware Downloader, and Perl/Shellbot.NAT trojan* types are the most common.

The analysis also points out the prevalence of a common SSH backdoor key across different countries and infrastructure providers, implying a coordinated operation possibly involving a globally distributed botnet or multiple actors exploiting a common attack vector.

Finally, the malware-hosting analysis highlights the United States and China as significant contributors, hosting most of the malware downloads. The type of malware hosted in these countries offers intuitions into the prevalent threats in these regions.

To answer the Hypothesis of type *H6*:

- *H6: All geographical regions contribute equally to the origins of malware activities.*

  The data collected from honeypots deployed on Azure and AWS contradicts this hypothesis. The geographical origins of malware attacks show significant disparities, indicating that all regions do not contribute equally.

- *H6.1: The contribution of geographical regions to the origins of SSH malware attacks is not equal.*

  The collected data aligns with this hypothesis. Notably, the origins of SSH malware attacks show significant geographical variations, with the United States, Singapore, and Germany being particularly active sources.

- *H6.2: All geographical regions contribute equally to the origins of malware uploads.*

The hypothesis is rejected based on the evidence. The data indicates a stark disparity in the volume of malware uploads from different geographical regions, with the United States leading in terms of contributions.

- *H6.3: The contribution of geographical regions to malware-hosting is not equal.*

  The data supports this hypothesis. There is a clear difference in the contributions of different regions to malware-hosting, with the United States and China standing out as major contributors.

- *H6.4: The use of a common SSH key across multiple attacks does not suggest a coordinated operation.*

  The data contradicts this hypothesis. The recurrent use of the same SSH key across multiple attacks suggests the possibility of a coordinated operation or botnet activity.

- *H6.5: All types of malware are downloaded with equal frequency.*

  The hypothesis is rejected based on the evidence. The data clearly demonstrates that different malware types are downloaded at varying frequencies with Linux/Mirai.Gen, Linux/Malware Downloader, and Perl/Shellbot.NAT Trojan is the most prevalent.

SMB

After examining the data from Azure and AWS honeypots, we discern several interesting differences and correlations that explain the SMB threats in these two cloud platforms.

Prominently, a different malware variant, *W32.FamVT.Pykspa.Trojan*, makes its appearance in the AWS data. This worm, known to propagate via various channels such as mapped drives and network shares (both of them use the SMB protocol), carries a backdoor that allows the execution of arbitrary commands by a remote attacker. The appearance of this new malware variant in the AWS environment, which is absent in the Azure data, presents an intriguing fact to our findings.

Simultaneously, we observe an increasing trend in the number of attacks originating from Vietnam, particularly involving this new *Pykspa* malware, as indicated by the heatmap's strong correlation. The heatmap also reveals that attackers from Indonesia and India are engaging with this particular malware family, further emphasizing its prevalence.

In contrast, the presence of the *WannaCry* variant *W32.FamVT.HyperiOF.Trojan*, commonly observed in the Azure data, is reflected in the AWS environment. This overlap suggests a certain level of commonality in the threat landscapes of the two cloud platforms.

However, compared to AWS, Azure exhibits a slightly distinct set of countries as the primary sources of SMB protocol attacks, but countries like India prevail in both the data centers' data. A lower count of connections and malware uploads could indicate stronger security measures on Azure against SMB attacks or, alternatively, a higher interest among attackers in infiltrating AWS systems via SMB.

Furthermore, the higher frequency of SMB attacks on AWS and the emergence of the new Pykspa malware could suggest a heightened interest among attackers in exploiting AWS systems. Yet, it may also be the case that this observed pattern might simply be a product of chance or a reflection of the larger population of AWS users.

8 Attack patterns: Cloud providers compared

In our research, we thoroughly examine correlations between different honeypots across various cloud service providers and protocols. This approach is aimed at showing shared attack patterns.

In this particular segment of our study, we employ data points that have been gathered over the defined one-week observation period. The observational period under consideration extends from midnight on the 19th of June, 2023, stretching right through the week to the final second of the 26th of June, 2023, at 23:59:59 hours.

This week-long period has been chosen as representative and offers a balanced view of the regular operations and incidents. It is long enough to capture daily and weekly patterns yet short enough to avoid the inclusion of potential anomalies that might occur over longer periods. This timeframe will provide a comprehensive snapshot of the attack patterns, allowing us to uncover insights and correlations in a controlled and manageable context.

Note that the initial surge noticed in each graph is attributed to the limited data points at the onset, not indicative of an actual pattern. As data accumulates, correlations evolve to be more exact. Thus, our focus lies on the under data-points, aiming to reveal facts about synchronized attacks or common vulnerabilities.

In this context, the correlation coefficient computed at each data point measures the degree of association or relationship between different honeypots across the various honeypot metrics, varying from 0 (no relationship between data points) to 1 (the data points have equal values).

In conclusion, these correlations have substantial practical implications. They shape our understanding of threats and attackers.

8.1 Correlation: Different Honeypots, Same Cloud Provider, Same Metrics

In the context of this research, conducting an exploration of correlations within the same cloud service providers proves to be useful. By drawing comparisons across diverse cloud service providers, we stand to discover possibly orchestrated intrusion patterns. Using a comparative methodology can help reveal recurring vulnerabilities in these data centers or expose the tactics used by attackers who launch simultaneous attacks on multiple honeypots.

8.1.1 Azure

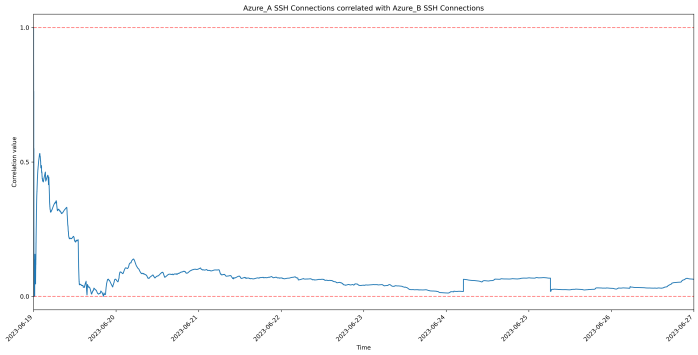

Figure 8.1: Login Attempts (SSH)

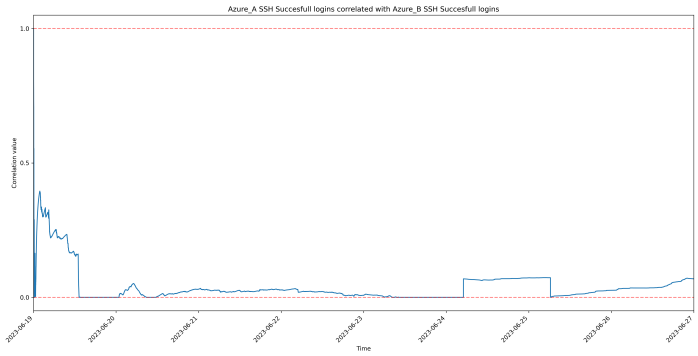

Figure 8.2: Successful Logins (SSH)

Interestingly, a distinct step-shaped pattern can be observed in a section of the correlation graphs (the 24h window starting 2023-06-24 at 12:00), suggesting the possibility of a coordinated attack between the two honeypots. This observation raises the hypothesis that a deliberate and synchronized effort might occur in both systems simultaneously.

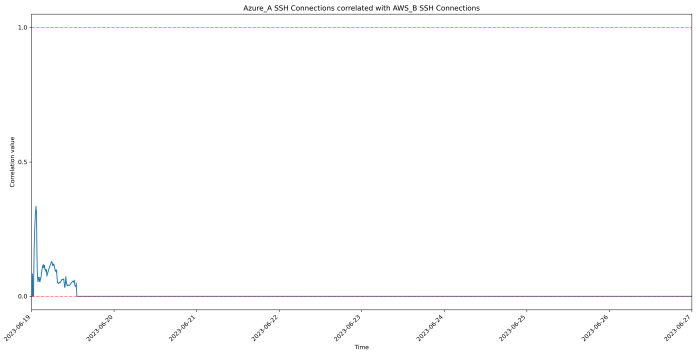

Figure 8.3: File Captures (SSH)

When examining the above correlation graph, a little and not easily identifiable step-like pattern emerges in a specific area. This pattern might be due to a coordinated attack shared between the two honeypots.

Findings

The analysis of SSH login attempts within two honeypots in the same Azure data center shows important facts. A step-shaped pattern is discernible in the correlation graph within a specific 24-hour window. This pattern strongly indicates a coordinated attack on both honeypots, suggesting an attempt to exploit system vulnerabilities simultaneously.

However, no significant correlation can be established when we focus on SSH commands.

In contrast, the correlation graph of the SSH file captures shows a not very discernible step-like pattern. Although vague, this could suggest a shared coordinated attack strategy between both honeypots.

8.1.2 AWS

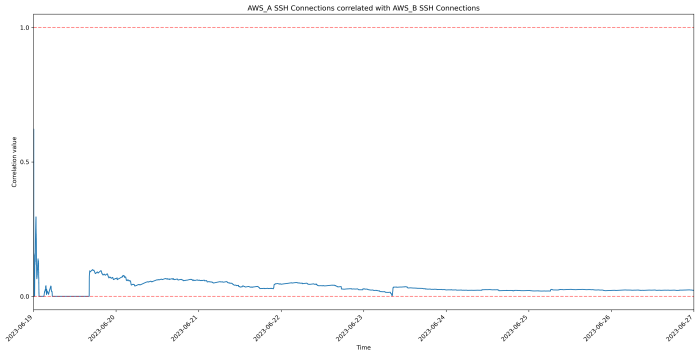

Figure 8.4: Login Attempts (SSH)

The correlation graph shows a certain level of correlation, although no distinct pattern can be discerned. The sudden spike in the graph potentially points out the beginning of collaborative attacks of attackers facing the two honeypots.

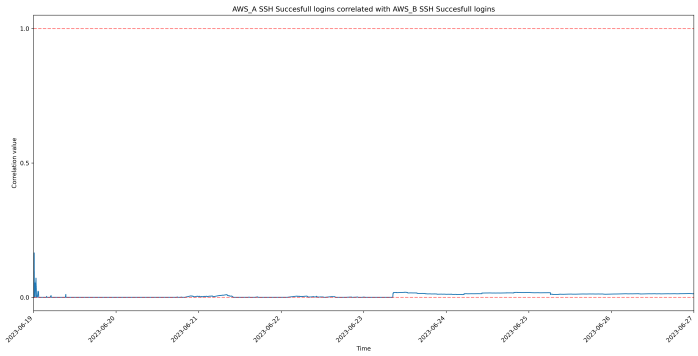

Figure 8.5: Successful logins (SSH)

The correlation graph shows a subtle step-like pattern in a specific area (from 12:00 of the 23-06), introducing uncertainty on a coordinated attack between the two honeypots.

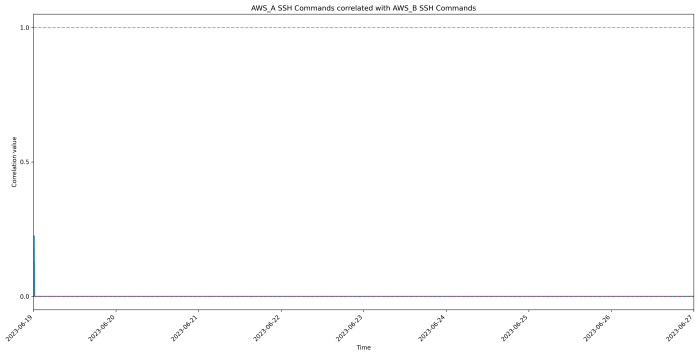

Figure 8.6: SSH Commands

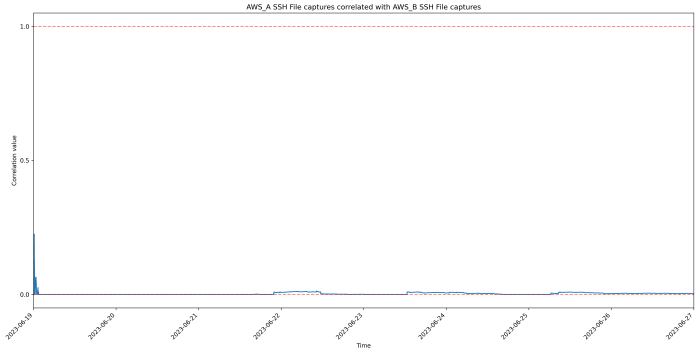

Figure 8.7: File Captures (SSH)

It is noteworthy that the correlation graph displays three faint step-like patterns, suggesting the possibility of three synchronized attacks targeting both honeypots. it is fundamental to clarify that the low correlation level does not necessarily suggest that the same attacker targeted the two honeypots. The correlation is computed on the number of attacks at a specific instance, and there may have been variations in the total number of attacks between the two honeypots, resulting in a higher correlation value.

8.1.3 Findings

During the analysis of different honeypots within the same cloud provider and the same protocol, the following findings were observed:

Azure (SSH):

- A distinct step-shaped pattern was observed in the correlation graphs for Login Attempts and *Successful Logins*, suggesting the possibility of a coordinated attack between the two honeypots.
- No significant correlation was found in the *Commands* graph, indicating a lack of coordinated activity in this aspect.
- In the *File Captures* graph, a subtle step-like pattern emerged, albeit not easily identifiable, raising uncertainty abou a planned attack between the honeypots.

AWS (SSH):

- The correlation graphs for *Login Attempts* showed a certain level of correlation, although no distinct pattern could be discerned.
- Similar to the *Login Attempts*, the correlation graph for *Successful Logins* displayed a subtle step-like pattern in a specific area, introducing uncertainty regarding a coordinated attack.
- No discernible significant correlation was observed in the *Commands* and *File Captures* graphs.

Overall, the findings suggest the possibility of coordinated attacks between honeypots within the same cloud provider and protocol. The step-shaped patterns and correlations indicate synchronized efforts.

8.2 Correlation: Different Honeypots, Different Cloud Providers, Same Metrics

Exploring Correlations between providers is a valuable aim of our study. We may find coordinated attack patterns by comparing patterns across different cloud service providers. This comparative approach could illuminate systemic weaknesses across platforms or reveal the tactics of attackers who simultaneously target multiple systems.

8.2.1 SSH Metrics

Azure A Correlated with AWS A

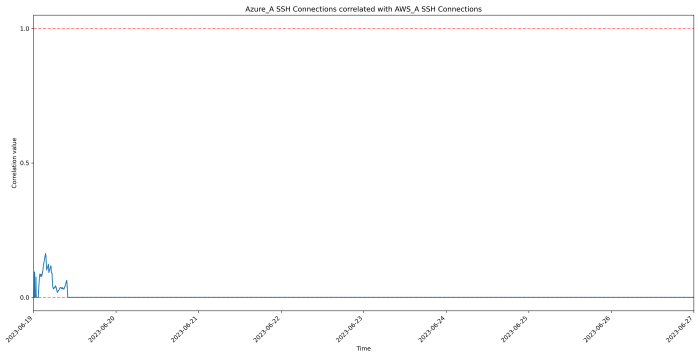

Figure 8.8: Login Attempts (SSH)

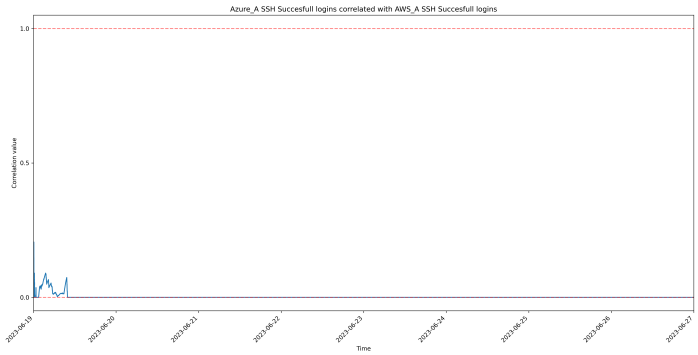

Figure 8.9: Successful Logins (SSH)

The correlation of (SSH) login attempts on two distinct honeypots, each situated in different data centers is minimal. This indicates that the relationship between the SSH login attempts on these two honeypots and the successful logins achieved is notably weak. Despite the strong similarity in the data, the correlation is surprisingly low.

Azure A Correlated with AWS B

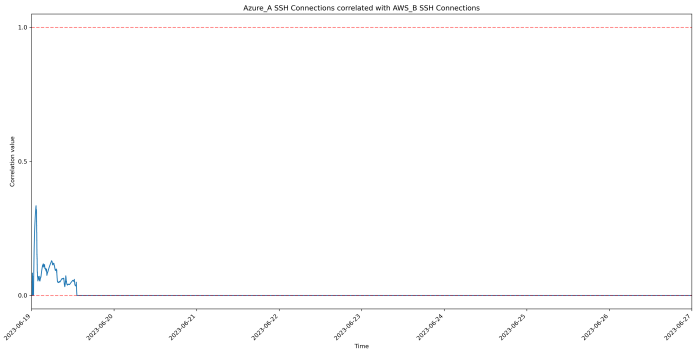

Figure 8.10: Login Attempts (SSH)

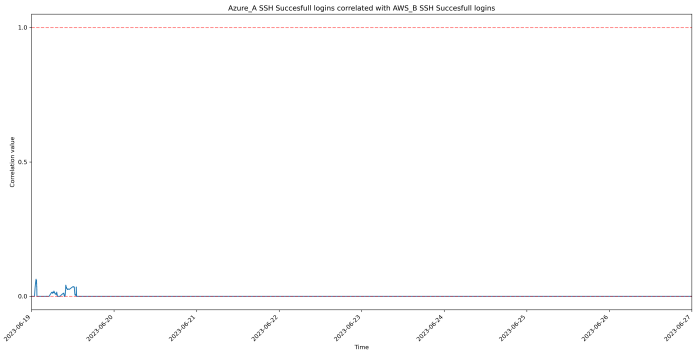

Figure 8.11: Successful Logins (SSH)

The correlation of (SSH) login attempts on these two distinct honeypots is minimal. This indicates that the relationship between the SSH login attempts on these two honeypots and the successful logins achieved is notably weak. Despite the strong similarity in the data, the correlation is surprisingly low. This observation is further corroborated by the high correlation between SSH login attempts and successful logins on these honeypots.

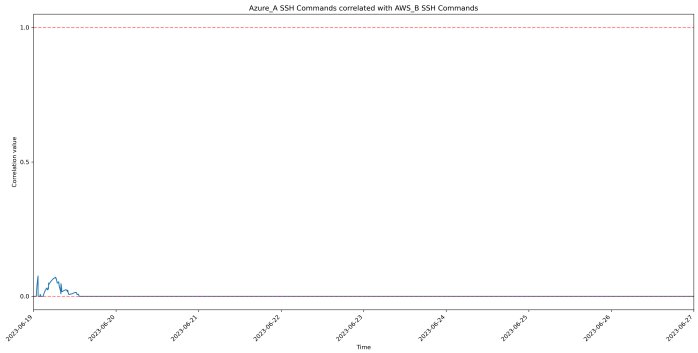

Figure 8.12: SSH Commands

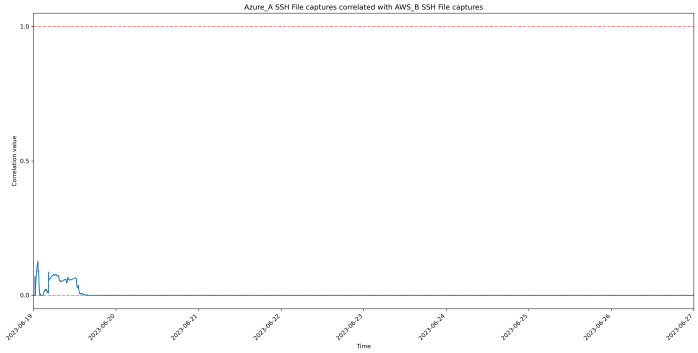

Figure 8.13: File Captures (SSH)

The correlation of (SSH) file captures from the two distinct honeypots is negligible. This suggests that the relationship between the SSH file captures of the two honeypots and the SSH commands executed on these systems is markedly weak. Despite the strong similarity in the data, the correlation is surprisingly low. This assertion is further substantiated by the high correlation between the executed SSH commands and the corresponding SSH file captures.

Findings

The analysis of SSH data from these two honeypots, Azure A and AWS B, even if situated in geographically close data centers, reveals notably weak correlations. This applies to SSH login attempts, successful logins, executed commands, and file captures. Despite their similarities, the data from these honeypots behaves independently. These findings highlight the need for tailored security measures in these diverse cloud environments.

Azure B Correlated with AWS B

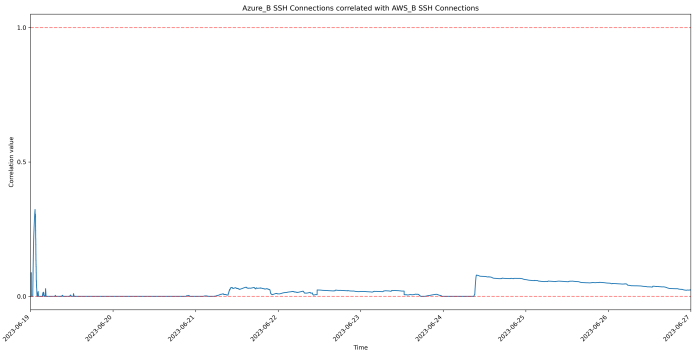

Figure 8.14: Login Attempts (SSH)

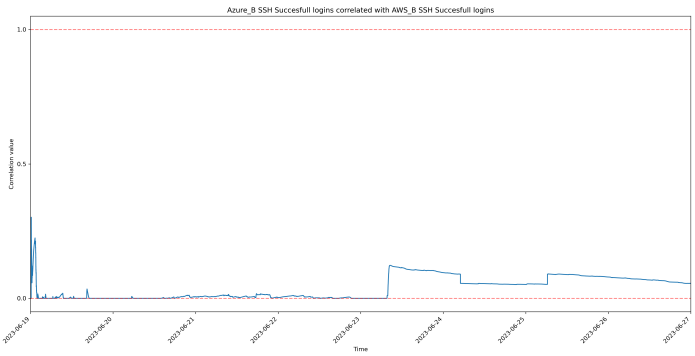

Figure 8.15: Successful Logins (SSH)

The analysis of (SSH) login attempts on the two distinct honeypots presents an interesting dynamic. Initially, the correlation between the login attempts on these two systems was low; this points out a lack of synchronicity in the attack patterns. However, as days pass, the correlation coefficient exhibits a fluctuating pattern, slightly rising, reverting to zero, and then escalating to the previous value. This suggests a non-linear relationship between the login attempts on the two honeypots, possibly introducing a new attacker common to the two honeypots.

Simultaneously, a similar trend is observed in the correlation between successful logins on these honeypots. This parallel rise could point out a shared set of successful strategies used by attackers across different data centers, despite the initial lack of correlation.

Furthermore, the marginally higher correlation value between successful logins within a 24-hour period starting from 23 at 12:00, as compared to SSH login attempts, suggests the possibility of a targeted attack with a higher rate of successful logins. This

hypothesis is partially supported by the significant correlation observed between SSH login attempts and successful logins. Another hypothesis emerges from the lower correlation of login attempts, which may have led to a higher correlation of successful logins due to the low number of SSH successful logins on both honeypots.

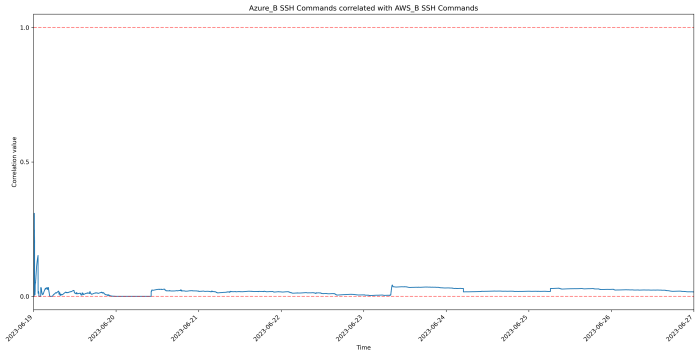

Figure 8.16: SSH Commands

Interestingly, we can observe step-shaped patterns in the correlation graph, which suggests the possibility of an attacker targeting both honeypots at the same time. The patterns observed in this graph are similar to the ones observed in the correlation of SSH successful logins; this can be attributed to the high correlation between SSH Successful Logins and SSH Executed Commands.

Findings

Our analysis of SSH login attempts on Azure B and AWS B shows interesting findings. The data indicates the functions of the correlation coefficient between login attempts on both systems over time, suggesting the presence of shared attackers employing a non-linear attack pattern.

Interestingly, we observed a parallel increase in the correlation between successful logins on both honeypots. This indicates the potential of attackers leveraging a common set of successful strategies across different data centers, despite the initial discordance in their attack patterns.

An intriguing observation was the marginally elevated correlation between successful logins within a 24-hour period, starting from the 23rd hour. This is indicative of a potential targeted attack with a higher success rate.

Our exploration also uncovered step-shaped patterns in the correlation graph of SSH

commands, suggesting a synchronized targeting strategy on both honeypots. The observed patterns are similar to the correlation trends in successful logins, likely due to the high correlation between successful logins and SSH commands.

8.2.2 SMB

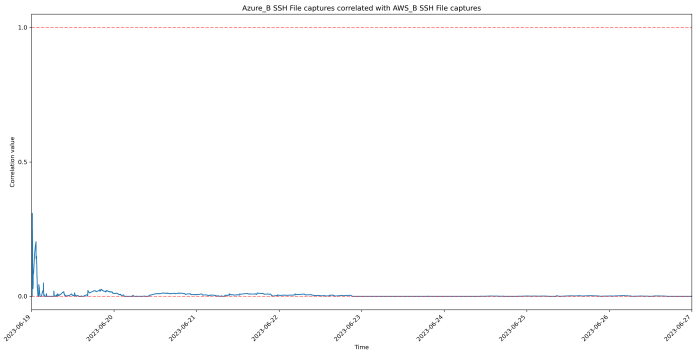

Figure 8.17: Connections (SMB)

Notably, the correlation graph reveals some step-like patterns, indicating a compelling correlation between SSH connections and SMB activity. This intriguing observation suggests the possibility of three synchronized attacks, targeting both honeypots with a coordinated approach between SSH and SMB protocols. Such findings present intriguing possibilities for coordinated efforts in the realm of SSH and SMB-based exploits.

8.2.3 HTTP

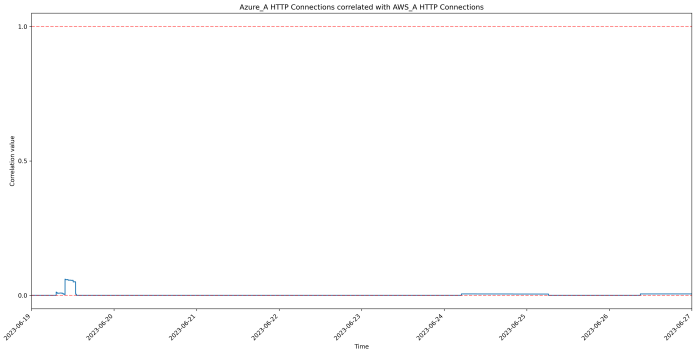

Figure 8.18: HTTP Connections

Findings

we can observe distinct patterns in the data, suggesting a potential shared activity between the two honeypots, particularly concerning the HTTP protocol.

8.2.4 Findings

The analysis of different honeypots deployed on different cloud providers, focusing on the same metrics, has produced several results:

SSH: Intriguing patterns emerge when comparing honeypots within the same cloud provider, indicating the possibility of coordinated attacks. However, when comparing honeypots across different cloud providers, no significant correlations were observed in most cases, suggesting a lack of coordinated attacks.

FTP: No significant correlation was observed between honeypots, indicating a lack of coordinated FTP-based attacks.

SMB: The correlation analysis between SMB honeypots revealed small but interesting step-like patterns, indicating a potential correlation between SSH connections and SMB activity. This suggests the possibility of coordinated attacks targeting both SSH and SMB protocols.

HTTP: Similarly, when comparing honeypots, non-distinct patterns were observed in the data, remotely indicating potential collaboration or shared activity, particularly in relation to the HTTP protocol.

8.3 Correlation: Same Honeypot, Same Cloud Provider, Different Metrics

Investigating correlations between different metrics within the same honeypot and cloud provider can yield valuable insights into attackers' behavior and modus operandi.

1. A correlation between SSH login attempts and successful logins can offer insights into the effectiveness of intrusion attempts.
2. The correlation between malware downloads and SSH commands can help identify potential post-intrusion activities.
3. The analysis of the relations between login attempts and SSH file captures can shed light on the data exfiltration practices of attackers.
4. The analysis of the correlation between SSH malware downloads and SMB or FTP uploads can reveal cross-protocol attack patterns.
5. The correlation between SMB connections and SSH login attempts may point to multi-vector intrusion strategies.

8.3.1 Correlation between Login attempts and successful logins (SSH):

The correlation between SSH login attempts and successful attempts is valuable for detecting brute-force attacks and identifying credential stuffing.

Azure A

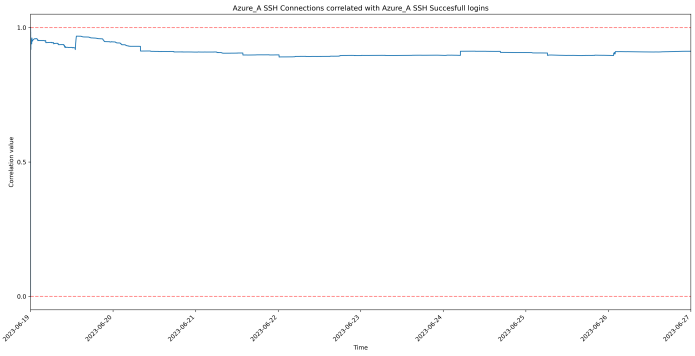

Figure 8.19: Total attempts (SSH) - Successful logins (SSH)

On Azure A, the high correlation (0.8 to 0.9) between SSH login attempts and successful logins in the honeypot suggests a strong relationship between these two variables. This indicates a direct influence of login attempts on the likelihood of successful SSH logins. The correlation coefficient is close to 1, suggesting a highly positive linear relationship, implying that the probability of successful logins also increases with the number of login attempts.

Azure B

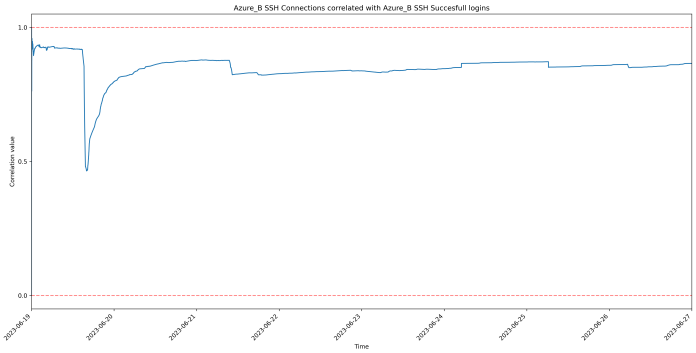

Figure 8.20: Total attempts (SSH) - Successful logins (SSH)

On Azure B, the high correlation between SSH login attempts and SSH successful logins indicates a strong positive relationship between these two variables. This suggests that as the number of login attempts increases, there is a corresponding increase in successful logins. However, the sudden plummet down near 0.5 for a few minutes may indicate a possible SSH brute force attack. It could may be due to a repeated effort to gain unauthorized access to the system during that specific timeframe.

AWS A

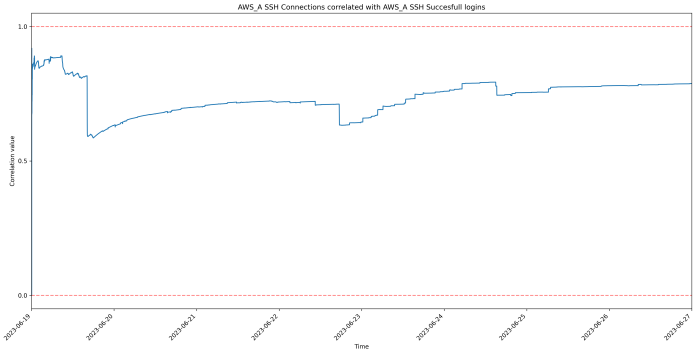

Figure 8.21: Total attempts (SSH) - Successful logins (SSH)

Here on AWS A, we observe another plummet at approximately the same time as the previous one. This could suggest that the DDoS attack was perpetrated on multiple cloud providers.

AWS B

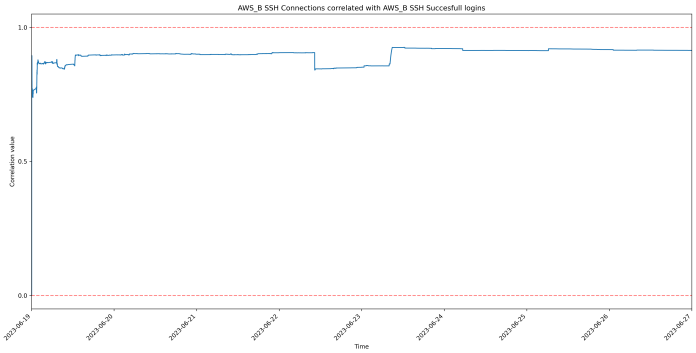

Figure 8.22: Total attempts (SSH) - Successful logins (SSH)

In this case, on AWS B, there is no sudden decrease in activity, then it cannot be attributed to any visible decrease, indicating that the alleged DDoS attack did not specifically target this server. Furthermore, we can observe a distinct step-like pattern, which suggests a well-defined discrepancy between the total number of attempts and the successful ones.

One potential approach to analysis involves investigating the relationship between Login Attempts (SSH) and SSH commands within the honeypot environment. By examining the timestamps and correlating the two-time series, it is possible to determine whether specific types of Login Attempts (SSH) are associated with particular SSH commands. For example, a sudden spike in Login Attempts (SSH) followed by a surge in SSH commands could indicate a coordinated attack.

8.3.2 Correlation between Malware Downloads and SSH Commands

Correlating SSH commands with malware downloads reveals valuable insights into attack patterns and aids in classifying different types of attackers. This correlation enables the identification of unauthorized access and the subsequent download of malicious files or corruption of the SSH access mechanism through the observation of SSH Commands.

Azure Honeypots Correlations

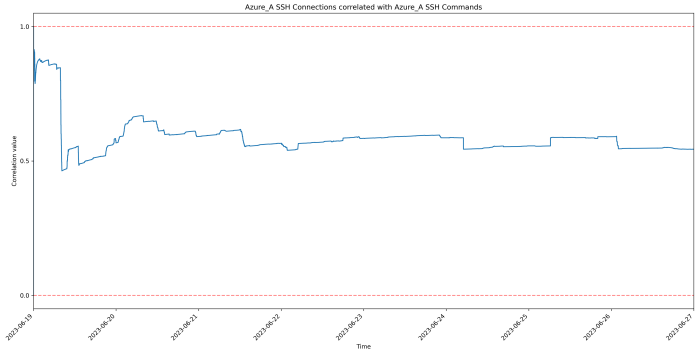

Figure 8.23: Total attempts (SSH) - Commands (SSH)

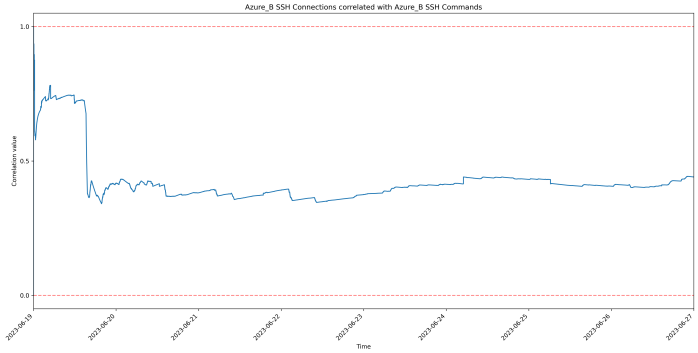

Figure 8.24: Total attempts (SSH) - Commands (SSH)

The already-encountered plummet is present here as well; the reason may be that commands usually follow after a connection. However, the plummet actually becomes a peak since the overall correlation is much lower, suggesting weaker connections between connections and commands when compared to the correlation between total and successful attempts.

AWS Honeypots Correlations

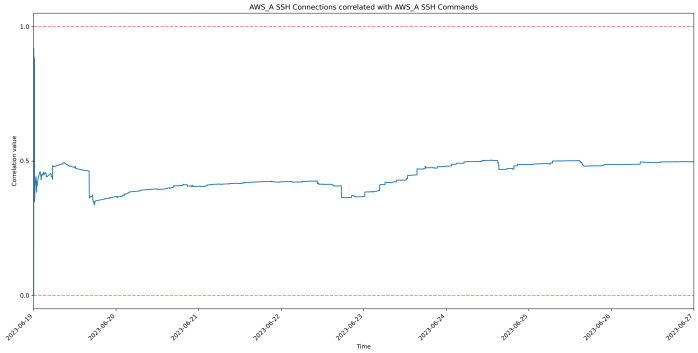

Figure 8.25: Total attempts (SSH) - Commands (SSH)

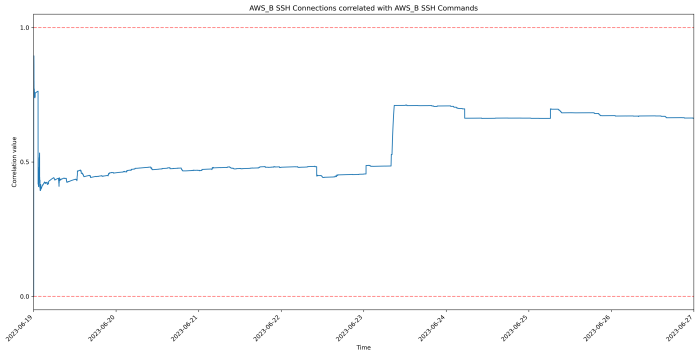

Figure 8.26: Total attempts (SSH) - Commands (SSH)

Here we don't see the spike anymore. We observe a similar correlation between the two honeypots. A distinct rise in correlation can be observed from 23-06; the same correlation is nearly reached due to the AWS A honeypot but with a constant increase.

8.3.3 Correlation between Login Attempts and SSH File Captures

The correlation between Malware Downloads and SSH Login Attempts offers a valuable tool for distinguishing between automated and manual attacks. Automated attacks often involve repeated login attempts, followed by malware downloads once access is achieved. In contrast, manual attacks might result in a more random pattern of SSH login attempts and malware downloads.

In this analysis, we aim to analyze the correlation between these two metrics to help identify the nature of the attacks—whether they are automated or manual. The hypothesis is that a strong positive correlation might indicate automated attacks, while a weaker correlation might suggest manual attacks.

Azure Honeypots Correlations

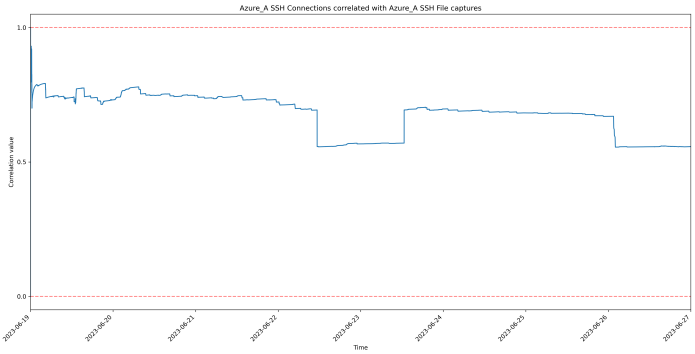

Figure 8.27: Total Login attempts (SSH) - File Captures (SSH)

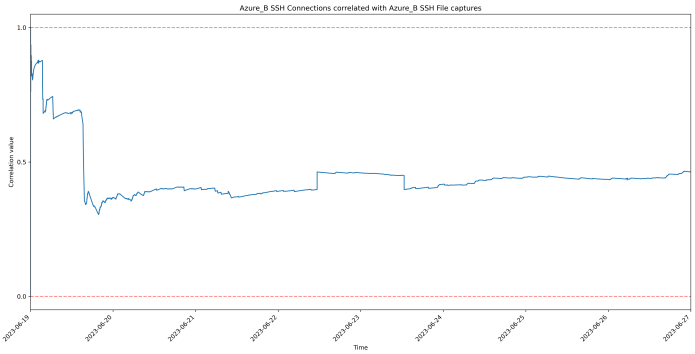

Figure 8.28: Total Login attempts (SSH) - File Captures (SSH)

We note a higher degree of correlation in Azure A compared to Azure B. Interestingly, a temporary decrease in correlation observed in Azure A appears to correspond with a surge in correlation in Azure B.

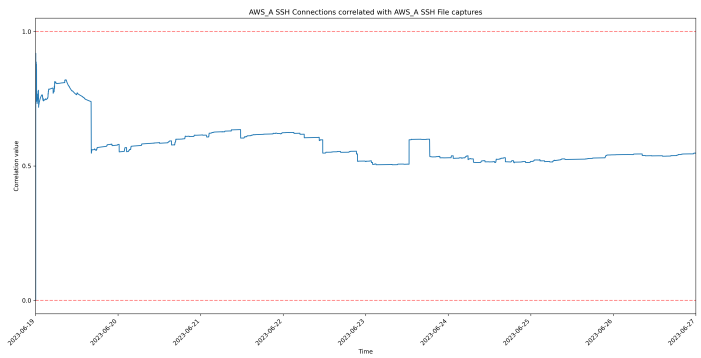

Figure 8.29: Total Loign attempts (SSH) - File Captures (SSH)

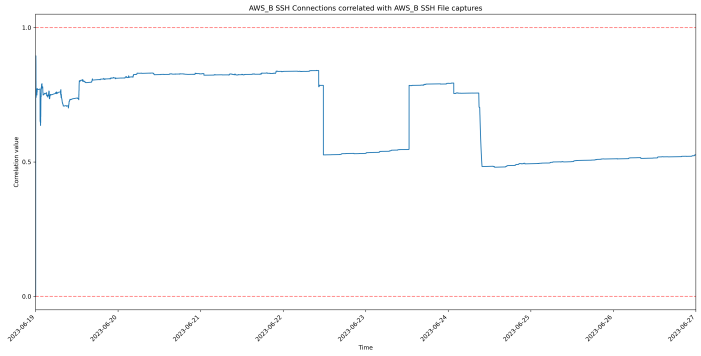

Figure 8.30: Total Login attempts (SSH) - File Captures (SSH)

As shown in Fig. 8.29 and Fig. 8.30 there is a high degree of correlation marked by various fluctuations. Notably, we can identify a common decrease in correlation over a 24-hour period starting at 12:00 on the 22. This pattern is congruent with what we have previously noted in the Azure honeypots.

The observed shift could be linked to the phenomenon observed in the Azure datacenter's honeypot, where multiple attackers ceased their activities on the Azure A honeypot and redirected their focus to Azure B. Consequently, this trend of reducing attacks is similarly noticeable in the AWS B honeypot, although it is absent in AWS A.

8.3.4 Correlation between SSH Malware Downloads and SMB and FTP Uploads:

Azure Honeypots Correlations

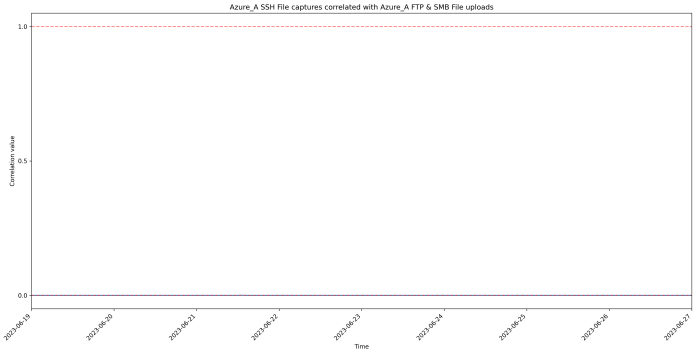

Figure 8.31: File Captures (SSH) - FTP & SMB Uploads

AWS Honeypots Correlations

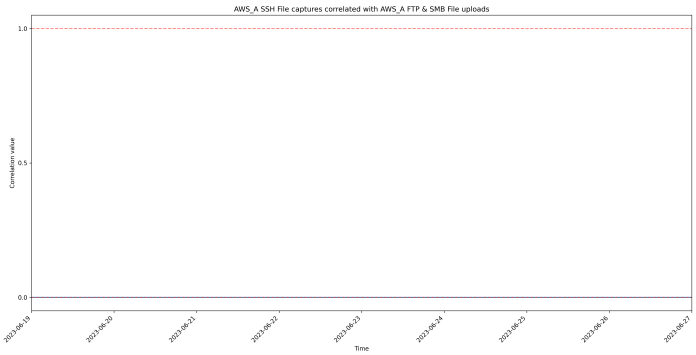

Figure 8.32: File Captures (SSH) - FTP & SMB Uploads

Here we can observe that both on AWS and on Azure there is no correlation between SSH file captures and FTP file captures. This indicates that these two types of attacks are not by the same kind of attacker.

8.3.5 Correlation between SMB connections and Login Attempts (SSH)

Azure Honeypots Correlations

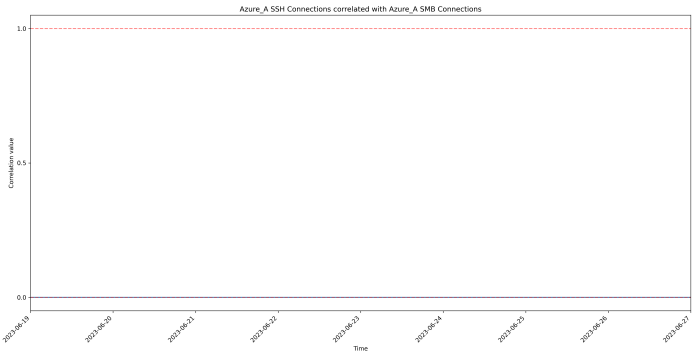

Figure 8.33: Total Attempts (SSH) - FTP & SMB Connections

AWS Honeypots Correlations

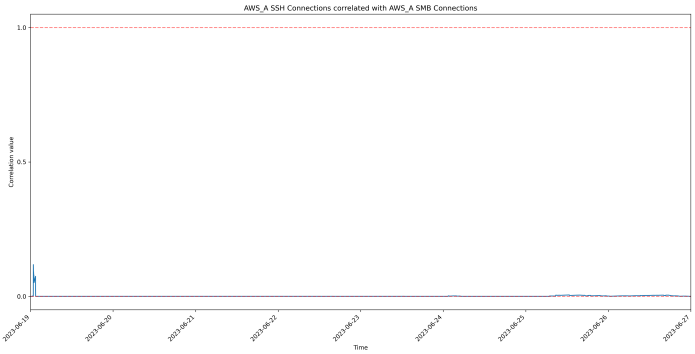

Figure 8.34: Total Attempts (SSH) - FTP & SMB Connections

Here we can observe that both on AWS and on Azure there is no correlation between SSH login attempts and SMB connections. This shows that the same kind of attackers do not perform these two types of attacks and therefore follow different patterns.

8.3.6 Findings

From the correlation analysis of diverse honeypot metrics, we derive the following findings:

1. **Strong Correlation between SSH Login Attempts and Successful Logins:** The analysis of honeypots deployed across Azure and AWS reveals a robust correlation between SSH login attempts and successful logins. This indicates that more login attempts correspond to an increased probability of successful logins. The discernible drops observed in the correlation graphs may indicate SSH brute force attacks, denoting periods of increased intrusion attempts.

2. **Varied Correlation between Login Attempts (SSH) and SSH Commands:** The correlation between SSH Login Attempts and SSH commands shows considerable heterogeneity across distinct servers on both Azure and AWS. This suggests that the relationship between these two metrics might be weaker than the correlation between total SSH Login attempts and SSH successful logins.

3. **Shared Attack Pattern:** The finding of a common attack pattern could imply a redirection of the attacker's focus towards a different honeypot, potentially showing a degree of adaptability in attackers' tactics.

4. **Limited Correlation between SSH File Captures and FTP File Captures:** The low correlation between these two metrics on both Azure and AWS suggests that distinct types of attackers likely orchestrate these attacks; this points out a lack of direct connection between SSH and FTP-based attacks.

5. **Absence of Correlation between SSH Login Attempts and SMB Connections:** The lack of correlation between SSH login attempts and SMB connections on both AWS and Azure supports the idea that these distinct types of attacks are executed by distinct attackers, reinforcing the notion of distinct threat actors behind SSH and SMB attacks.

8.4 Findings: Correlations Attack patterns of Cloud providers compared

Based on our correlation analysis of our honeypots data, **focusing on the same metrics**, the findings reveal patterns and potential coordinated attacks. Notably, the data points towards synchronized attackers' efforts within the same cloud provider, suggesting coordinated attacks exist.

In the case of SSH, a noticeable pattern was detected within the same cloud provider, indicating potential coordinated attacks. However, such patterns were not prominent across different cloud providers, implying a lack of coordinated attacks.

Regarding FTP, the data shows no significant correlations either within or between cloud providers. This lack of correlation is due to the near absence of FTP attacks in the data set rather than a lack of coordinated activity.

With respect to SMB protocols, visible step-like patterns emerged within the same cloud provider. This pattern indicates a potential correlation between SSH connections and SMB activity, suggesting the possibility of coordinated attacks targeting SSH and SMB protocols.

For HTTP, the data shows distinct patterns within the same cloud provider, indicating potential coordinated activity.

When **focusing on the same cloud providers**, Azure and AWS, the findings show the possibility of coordinated attacks between honeypots within the same cloud provider, mainly for the SSH Protocol. The distinct step-like patterns indicate synchronized efforts. However, the correlation analysis also reveals lower but still significant coordination in some aspects, such as commands and file captures.

The correlation analysis of the same honeypot metrics deployed across Azure and AWS reveals a strong correlation between SSH login attempts and successful logins. This suggests that increased login attempts correspond to an increased probability of successful logins. Notably, the correlation between login attempts and SSH commands exhibits considerable variability across different servers, suggesting a weaker relationship between these two metrics compared to the correlation between total attempts and successful logins. Identifying a shared attack pattern implies the potential adaptability of attacker tactics. this trend can also be attributed to the presence in the data of attackers who discover exploitable systems but do not execute any commands. The relatively high correlation observed between SSH Login Attempts and Malware download suggests that we observe a high number of automatic attackers that gain access to the system and then download malware with a high success rate; this confirms the deduction of the previous chapters. Furthermore, the absence of correlation between SSH login attempts and SMB connections reinforces the notion of separate threat actors for distinct attack vectors.

In conclusion, our findings provide valuable insights into potential coordinated attacks across different cloud providers and protocols. The previous work aims to unravel these complex patterns further, providing a more specific understanding of attacker behaviors and strategies, thereby informing more effective and dynamic defenses.

Therefore we can now answer the following hypotheses:

H7*: There is a significant correlation between the cloud provider's hosting choice and the observed activity.*

Our analysis and prior chapter findings show the significant influence of a cloud provider's choice on the observed metrics. This includes the attack patterns, protocols employed, and the total number of attacks. Consequently, we accept Hypothesis 7 (*H7*) due to the substantial evidence of the correlation.

H7.1: The data observed on machines within the same cloud provider do not show a significant dependency.

The patterns and correlations observed within the same cloud provider can indicate somewhat of a dependency. We cannot determine with certainty if this suggests that attack vectors, rather than the choice of cloud provider, influence the observed data.

H7.2: There is a high chance that SSH login attempt leads to successful ones.

Our analysis robustly confirms a high correlation between SSH login and successful attempts. This supports the hypothesis and implies that an Increase In the number of login attempts often increases successful intrusions.

H7.3: There is a high chance that an attacker will download or upload malware.

The high correlation between SSH login attempts and malware downloads confirms the hypothesis. This suggests successful logins often lead to potential malware downloads or file captures.

8.5 Correlating Trends: a Quick evaluation

Besides the more standard approach of correlating the number of attacks over time, we can also analyze the overall trend of attack metrics between two honeypots. To achieve this, we can modify the two-time series by subtracting each value at time t from the value at time t-1. We can identify and correlate similar attack trends among multiple honeypots by doing so.
Within the scope of our study, it's plausible that a low degree of correlation might emerge from our trend analysis. This assumption is based on the intrinsic characteristics of the dataset, which can often be sporadic, multifaceted, and influenced by too many external variables governed by attackers. Furthermore, the threat landscape constantly evolves with new patterns, making it difficult to establish stable trends over extended periods. Consequently, while our analysis method of computing trend differences between two time points could discover correlations among multiple honeypots, the degree of correlation may be modest due to these inherent complexities.

8.5.1 Correlating Trends: Different honeypots, Same cloud provider, Same metrics

Azure Honeypots (Azure A - Azure B)

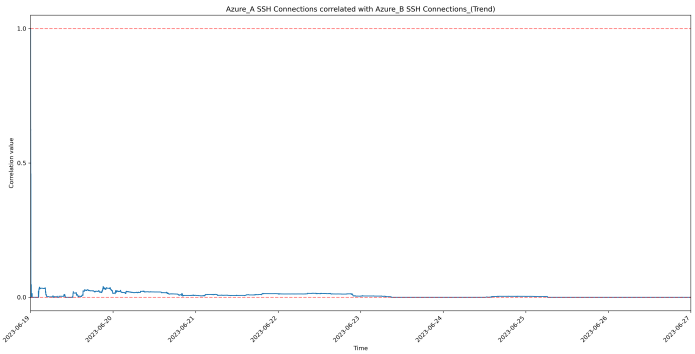

Figure 8.35: SSH Login Attempts

Here we notice a slight increased correlation between the login attempt trends of the two Azure honeypots. This observation could suggest a simultaneous increase in SSH Login Attempts targeting these two honeypots. However, caution should be used in interpretation, as the level of correlation observed, while higher than 0, is not robust enough to be classified as statistically significant.

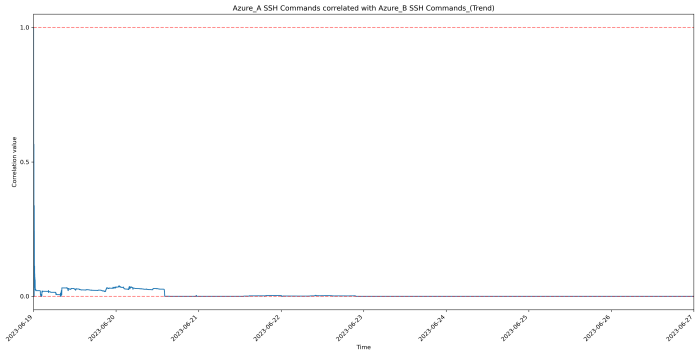

Figure 8.36: SSH Commands

From the above correlation of trends in SSH commands we can see that the slight increase in the correlation coefficient in the previous graphs is somewhat mirrored at the beginning of Fig. 8.36, an explanation can be the relatively high correlation observed between SSH Login Attempts and SSH Commands.

8.5.2 Correlating Trends: Different honeypots, Different cloud provider, Same metrics

Azure Honeypots (Azure A - AWS B)

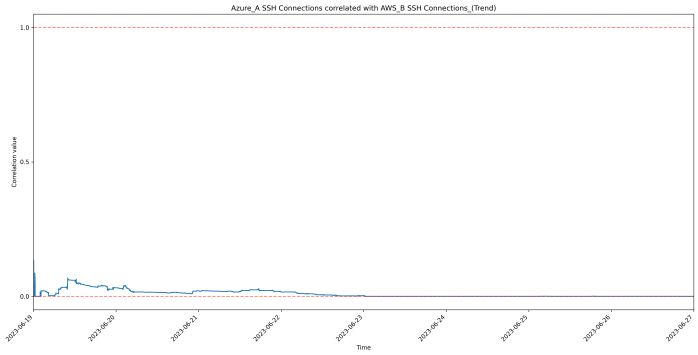

Figure 8.37: SSH Login Attempts

Here, we note a minimal rise in the correlated trends, potentially hinting at a parallel increase in SSH Login Attempts directed toward both honeypots. However, this increment does not reach a value to be regarded as a significant insight.

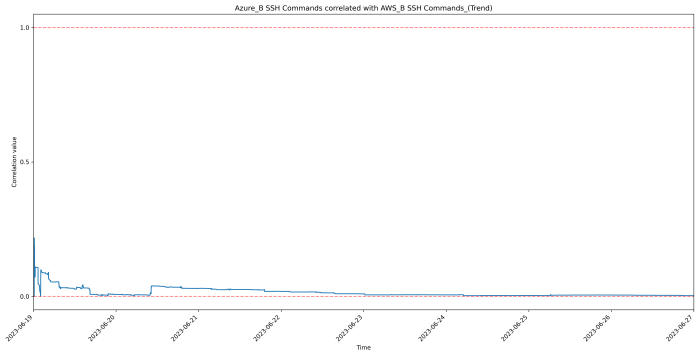

Figure 8.38: SSH Commands

We observe a slight increase in correlated trends, indicating possible concurrent SSH login attempts on both honeypots. However, this subtle rise does not provide significant insights.

8.5.3 Correlating Trends: Findings

Building on the initial explanation offered in the introduction, it is crucial to state again why we might expect a lower correlation in our analysis. The nature of intrusion data, specifically its built-in complexity and volatility, is a significant factor in these lower correlations. The data we are dealing with is influenced by a large array of external variables, all controlled by threat actors who follow unpredictable and dynamic attack patterns.

Intrusion activities are not uniform or constant; they are frequently changing in response to many factors, such as adjustments in the threat landscape, the evolution of attacker techniques, the introduction of new vulnerabilities, and shifts in attacker focus. These variables are incredibly difficult to control or account for, adding layers of complexity to the analysis.

This further contributes to the low correlation levels we might observe in our trend correlation analysis.

A marginal amplification in correlation is detected between the SSH Login attempts on instances Azure B and AWS B. This discovery may suggest possible synchronized SSH attacks aimed at both Azure B and AWS B honeypots. The parallelism in the patterns of these attempts reinforces the idea generated in the previous correlations of a coordinated SSH attack, possibly from the same botnet. It is important to recognize that the adopted methodology for trend analysis may have limitations in capturing all relevant variables and patterns.

8.6 Inconclusive Correlations

Within this chapter, we focused on correlation graphs involving various honeypots and diverse metrics. Throughout this analysis, we encountered instances where the relationships between certain factors remained enigmatic or failed to yield significant insights. Consequently, we made the deliberate choice to omit these correlation graphs.

For instance, the comparative examination of SSH commands across distinct honeypots within a shared provider did not provide any useful information. On a similar note, SSH metrics and HTTP connection correlations across different providers did not yield any noteworthy insight. A parallel situation emerged when analyzing the trend of login attempts and commands across distinct Honeypots on different providers. The following table summarizes most of the correlation analyses that didn't provide useful data:

Table 5: Summary of Correlations that did provide insignificant data

Correlation
FTP & SMB Connections on AWS
SSH & SMB Connections on AZURE
SSH Login attempts between AWS A & Azure B
SSH Successful Logins between AWS A & Azure B
SSH Commands between AWS A & Azure B
SSH File captures between AWS A & Azure B
SSH Connections Azure A - AWS A (Trend Correlation)
SSH Connections Azure B - AWS A (Trend Correlation)
SSH Connections Azure B - AWS B (Trend Correlation)
SSH Commands Azure B - AWS A (Trend Correlation)
SSH Commands Azure A - AWS B (Trend Correlation)

9 IoT-Focused Analysis: Evaluating IoT-Targeted Attacks

As we continue our in-depth exploration of IoT-targeted attacks in this chapter, we must revisit our findings from the previous analysis on SSH Malware uploads. The prominence of specific malware types and their geographical distribution, as revealed in Chapter 6 it's crucial for the current analysis. It is the predominance of *Linux/Mirai.Gen* that guides our current focus, given their evident threat to IoT environments, a quick evaluation of which devices are most vulnerable is indeed necessary.

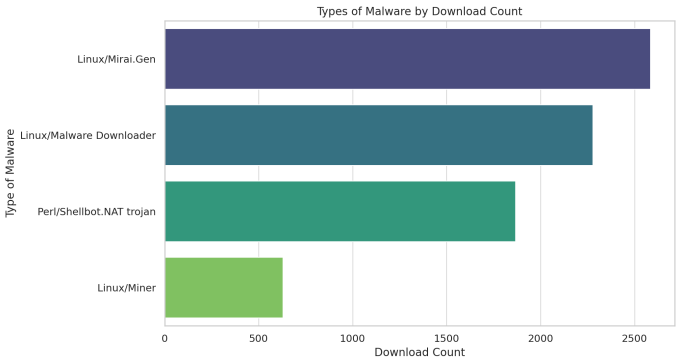

Figure 9.1: Distribution of Malware Types by Download Count

The previous findings underscored the pervasive nature of *Linux/Mirai.Gen* types of malware, particularly in the United States, have a global download count of over 2500 copies. This predominance is not merely an indication of their popularity, but it also signifies the potential vulnerabilities they exploit in our IoT landscape. As we explore this further, we seek to understand the implications of these malware types in IoT-targeted attacks.

9.1 Unveiling Mirai: A Deep Dive into its IoT Focus:

Mirai is a type of malware that generally targets low-end devices, once infected these devices become a bot, a remotely controlled device, which joins a large group of other bots, creating a "bot-net".
Once the network node becomes a bot, it tries to spread the malware in the network, this behavior explains the predominance of Mirai in what is observed in our honeypots.
Attackers that "own" a botnet can use all these infected devices to target a specified victim (such as a web service) with a DDoS attack.
The attack vector of this malware mostly focuses on SSH clients, employing brute force techniques to gain unauthorized access. After gaining access, the malware quickly spreads and enlarges the botnet's control through the newly downloaded script. This seamless integration further amplifies the botnet's capacity, enhancing its power to execute Distributed Denial of Service (DDoS) attacks.

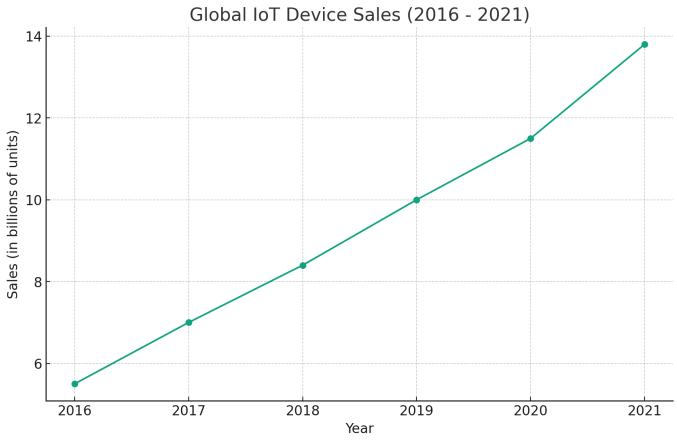

Figure 9.2: Global IoT Device Sales *(data source: IoT Analytics)*

IoT-Devices are often a receptacle for the Mirai malware, given their general lack of maintenance and usage of default passwords and usernames. Also, the always-increasing number of these devices gives cyber-criminals the possibility of a theoretically enormous botnet, that can successfully target even the most resilient of services.

In 2017, an investigation showed that 15% of IoT devices operate on default usernames and passwords [29]. If this trend is stationary, by 2023, a staggering 2 billion weakly protected devices will be interconnected. These devices can be compromised by attackers employing standard password lists and adopting attack patterns documented in this research.

9.2 Botnet Assessment: Understanding the Impact of Mirai Attacks

In recent years, Mirai has been accused of being the cause behind massive Distributed Denial of Service attacks. For example, in October 2016, significant disruption was observed affecting famous web services including Twitter, Reddit, and Netflix. This perturbation, known as the *Dyn DNS* attack, was attributed to the Mirai botnet and achieved a staggering traffic volume of approximately 1.2 Tbps, marking it as one of the most substantial in recorded history [30].

9.3 Finding Mirai in our Data:

9.3.1 Credentials Analysis

e consider in some detail the risk of attackers manipulating devices through the use of basic password lists, along with analogous attack patterns discerned from our honeypot data. Our focus shifts towards understanding common password and username combinations used by these malicious scripts. The goal is to ascertain whether our data supports the hypothesis of a significant presence of IoT-targeted attackers.

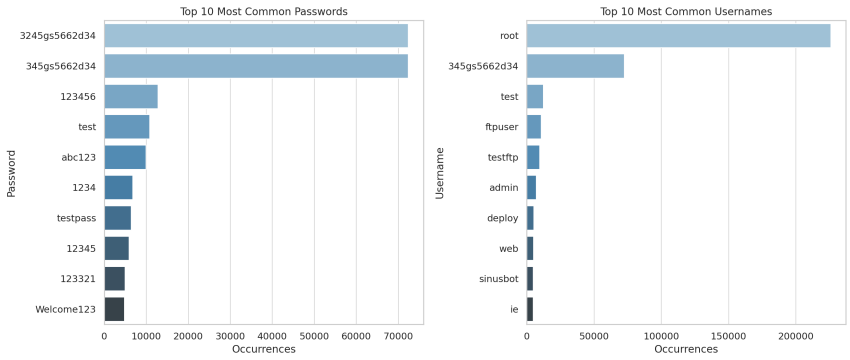

Figure 9.3: Top 10 usernames & passwords used on our SSH honeypots

The recurring username and password combination *3245gs5662d34* and its variant *345GS5662d34* have been noted across multiple instances in our data marking a surprising pattern because these passwords do not correspond to any widely used device, service or known attack vectors.

Interestingly, this phenomenon is not unique to our data. Similar patterns have been observed in other honeypots located around the globe [31] [32], suggesting a wider trend that extends beyond our study. Other researchers have been unable to attribute these specific password patterns to a known source or cause, despite their widespread appearance [33].

One theory is that these credentials could be part of a botnet's operations, yet the evidence to support this remains elusive. Furthermore, these username and password combinations do not appear in any known compilations of leaked passwords, making their widespread use even more intriguing. another explanation of the usage of these passwords could be a way for attackers' scripts to detect honeypots, their prevalence across different honeypots underscores the need for further investigation into their potential implications for threats.

To continue the research of Mirai malware in our data:

We will base our reference point on the hardcoded credentials [34] employed within the attack vector of the *Mirai.gen* malware. This comprises a set of over 50 distinct username and password combinations, which, as imagined, are predicated on frequently used combinations prevalent in the Internet of Things devices. To this end, we shall conduct a comparative analysis of our own usernames and passwords against this set.

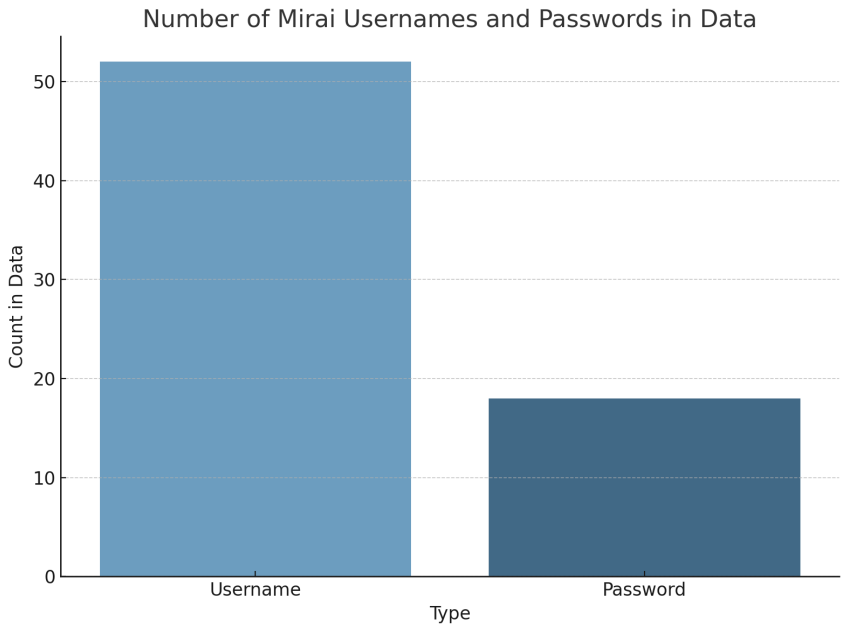

Figure 9.4: Number of Distinct Mirai.gen Usernames and Passwords in our honeypot data

This bar chart suggests that a considerable portion of our dataset consists of usernames and passwords that are hardcoded in the *Mirai.gen* malware. It is particularly concerning that most of the Mirai usernames appear in our dataset, indicating that these usernames are common targets in the attacks we observed.

However, fewer than one-third of the Mirai passwords are present in our dataset. This discrepancy could be attributed to the diverse password preferences among different attackers, as well as the relative age of our password list.

The list is derived from an early version of the Mirai malware, which frequently contains customized code that is continuously updated by the attackers, as referenced in [35]. Another plausible reason might be the use of different types of malware that come with a distinct set of hardcoded passwords.

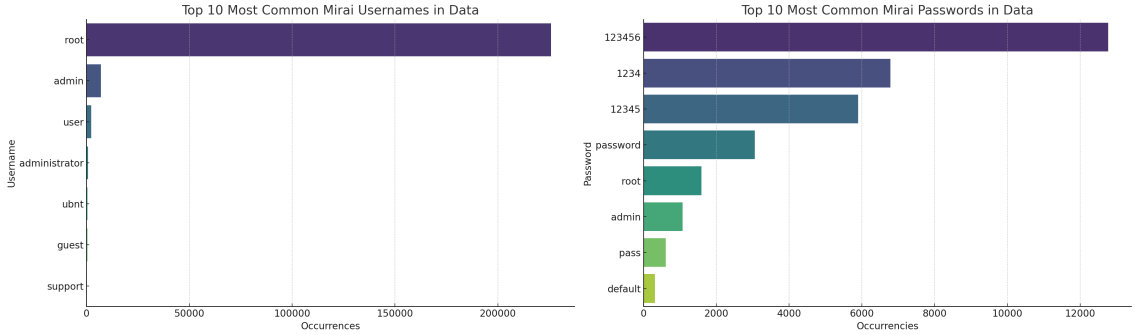

Figure 9.5: Top 10 Mirai.gen Usernames and Passwords in our honeypot data

Usernames: Among the Mirai hardcoded usernames present in our dataset, "root" is by far the most common, appearing over 225,000 times. Other frequently occurring usernames include *"admin", "user", "administrator", and "ubnt"*.

Passwords: The most frequently observed Mirai hardcoded password in our dataset is *"123456", followed by "1234", "12345"*, and *"password"*. Other commonly used passwords include "root", and "admin".
These observations reveal that most of the attacks we've witnessed have employed usernames and passwords known to be hardcoded in the Mirai malware. This suggests that our network is subject to attempted intrusions by Mirai or versions of it.

9.3.2 Bot Fingerprints

In line with our previous analyses, particularly those concerning the fingerprinting of attackers, we have discovered that the fingerprint most frequently associated with SSH attackers is generated by libssh [4], accounting for over 65% of total attacks. This prevalence hints at a significant presence of custom attack scripts, potentially originating not just from botnets in general, but specifically from those associated with the Mirai malware. This correlation indicates a prevalence of these types of attack vectors.

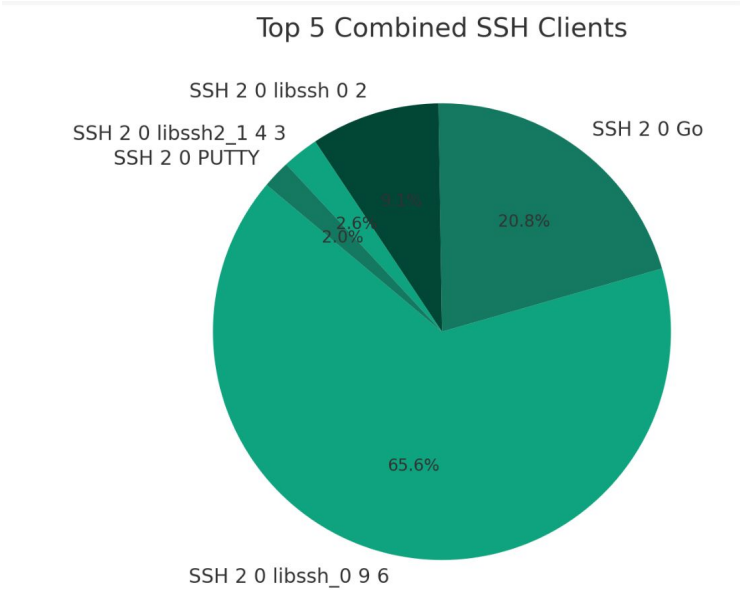

Figure 9.6: Top 5 SSH Clients

The popularity of this fingerprint, combined with the observed usage of common Mirai usernames and passwords, confirms our hypothesis of a significant Mirai malware presence in the attackers.
This interpretation, however, could be too biased by our previous findings. A high frequency of libssh fingerprints could also be due to broader use of custom attack scripts beyond just Mirai. Given the open nature of libssh, other threat actors could utilize it in their attack scripts.

[4]Libssh is an open-source library that enables applications to provide or use SSH. It is widely used due to its flexibility, enabling developers to incorporate SSH capabilities directly into their applications. However, its very accessibility and popularity make it a tool of interest for threat actors who aim to exploit SSH vulnerabilities.

9.4 IoT-Focused Analysis: Findings

This chapter has focused on IoT-targeted attacks, emphasizing the Mirai malware. Our analysis of honeypot data revealed common username and password combinations employed in SSH attacks, many of which matched those hardcoded in the Mirai malware. Notably, the combination *3245gs5662d34* and its variant *345GS5662d34* were prevalent across different honeypots worldwide, despite not corresponding to any known attack vector.

Our comparative analysis with Mirai's hardcoded credentials indicated a significant overlap, suggesting a potential vulnerability to such attacks. However, the presence of these credentials in our dataset does not necessarily imply a successful breach, as many systems may have updated these default credentials or implemented additional security measures.

In conclusion, our findings show the importance of simple but robust security practices, including the use of strong, unique passwords and regular updating of IoT devices.

Our analysis confirms Hypothesis H8, which proposed that a substantial part of attacks target IoT devices. In fact, an examination of common username and password combinations used in the observed SSH Login attempts, many of which are default or simplistic, indeed suggests that IoT devices are a significant target.

Furthermore, Hypothesis H8.1 is supported by the overlap between the hardcoded credentials used by the Mirai malware and those present in our dataset. This strongly implies that a large volume of attacks originates from Mirai botnets. The presence of the majority of Mirai usernames and a noticeable portion of Mirai passwords in our data underscores the potential vulnerability of our network to Mirai or similar types of malware.

10 Conclusions & Future Works

As we conclude our study, it's important to acknowledge how limited resources have impacted our research. Using only two honeypots from different cloud providers has limited our ability to draw clear conclusions because of the small amount of data we had. Remember, we intentionally set these limits, and while they do make our findings a bit less certain, they don't take away from the overall validity of our thesis's findings.

10.1 Work Results

This study of honeypots in cloud environments shows interesting attack patterns. For instance, **Azure sees three times more SSH attacks than AWS but twenty times fewer SMB attacks**. This highlights each platform's different security challenges and could be attributed to each provider's different security measures.

The study also shows that attack patterns can depend on the time and day of the week. For example, **attacks tend to happen more at night and are most common on Mondays**. This pattern emphasizes the importance of considering attackers' time zones when analyzing such data.

The study also found a significant spatial pattern to the attacks, with the **United States and China being the primary sources**. Smaller nations like Singapore and Hong Kong also displayed many attacks relative to their size and economic scale. Moreover, the study also pointed out that the spread of AS across different cloud services and within each AS was not equal. **Most attacks came from the AS run by Digital Ocean, indicating that most attackers leverage the cloud to carry out their attacks.**

It's also worth noting that Linux plays a significant role in the SSH attack landscape. As a matter of fact, **both providers were mostly attacked by Linux clients**. The data also showed an important link between the protocol type and the attack frequency, suggesting that attackers might choose their targets based on the protocol. It was also observed that some attackers **manipulate SSH client strings to exploit known vulnerabilities**, AWS seems to be less targeted by these types of attacks.

The study also showed that the skill levels and resources available to attackers can significantly influence their attack patterns. **Automated attacks,** which are usually less complex and last for a shorter time duration, **represent the majority of threats** (98.5%). In contrast, manual attacks, which require more skill and planning, can last for several minutes and represent a small part of the threat landscape.

The data analysis provided interesting insights into the behavior of attackers. The attackers were categorized into distinct profiles based on their interactions with honeypots, services, and cloud providers. **Most attackers (67%) specialized in**

attacking a single honeypot within a single provider, focusing on just one service. However, a significant portion (30%) showed more versatility, launching attacks on multiple honeypots across different providers. A **small but especially dangerous group of attackers (0.1%) demonstrated a capacity for broad and diverse attacks, targeting all honeypots across multiple services.**

After correlating data from honeypots in the same cloud, and hence with the the same provider, we have observed patterns indicating the presence of coordinated attacks. These attacks occur within the same cloud and, sometimes, in smaller proportions, across different clouds.

Uploaded and Downloaded Malware provided valuable data about their origins, types, and goals. **The geographical analysis showed that cyber-criminal activity is global, with significant hotspots in the United States, Singapore, and Hong Kong**. SSH attacks often aim to hijack the SSH-authorized keys, providing the attacker a backdoor for reentry into the compromised system; interestingly, a common backdoor key has been uploaded by most attackers (over 95%). **The malware download analysis revealed specific malware types, notably *Linux/Mirai.Gen*, *Linux/Malware Downloader*, and *Perl/Shellbot.NAT Trojan*, as the most common.**

We have extended our analysis of the downloaded malware, focusing specifically on an IoT-targeted approach due to the alarming occurrence of Mirai malware found on the honeypots (with over 2500 copies isolated on Azure alone). By examining Mirai's source code SSH password list, along with the fingerprints of SSH clients that launched attacks on the honeypots, we were able to confirm our hypothesis. The data pointed to a **majority of IoT-targeted attacks, carried out by one or multiple Mirai-type botnets**, within our attackers' data. This finding underscores the urgent need to fortify IoT security, particularly in light of the escalating adoption of IoT technologies and the emergence of these now-predominant threats.

In light of what we found in this research, it becomes evident that both AWS and Azure have their sets of security challenges. Azure encountered a 300 % increase in SSH attacks compared to AWS, However, Azure's resilience is showcased by its substantial resistance to SMB attacks, having encountered twenty times fewer SMB attacks than AWS. This strong contrast in attacks shows the unique security architectures and measures each platform has implemented. A significant observation relates to the behavior of SSH-downloaded malware. While Azure recorded many such incidents, AWS displayed remarkable resilience, with instances dropping to zero. This pattern suggests that **AWS may have enforced stringent security measures specifically against external malware downloads**, offering solid protection against these types of attacks. Moreover, AWS's reduced vulnerability to manipulations of SSH client strings emphasizes its strength in particular areas, adding to its robust defense profile. **Azure's susceptibility to SSH attacks is evident from the numerous attempts to log in via SSH and download malware. On the other hand, AWS's primary vulnerability is in the SMB**

attack domain. These variations in attack types, sources, and frequencies underline both platforms' distinct security challenges. Potential users must carefully evaluate these vulnerabilities in conjunction with their operational requirements, the nature of their data, and the protocols they intend to implement. This consideration will enable them to align their choices with each platform's specific risks and benefits.

Objective	Achieved	Findings
Time and Day Impact on Attacks	✓	Attacks tend to happen more at night and are most common on Mondays
Geographical Attack Patterns	✓	United States and China being the primary sources. Most attacks came from the AS run by Digital Ocean
Protocol dependency	✓	Azure experienced three times more SSH attacks than SMB, while AWS was attacked slightly more on SMB than SSH
Linux's predominant role in attacks	✓	Data showed that 98% of attacks came from Linux clients.
Attackers' Skill Levels and Resources	✓	Automated attacks represent the majority of threats (98.5%)
Behavior and Profiling of Attackers	✓	Most attackers (67%) specialized in attacking a single honeypot within a single provider, focusing on just one service. However, a significant portion (30%) showed more versatility, launching attacks on multiple honeypots across different providers. A small but especially dangerous group of attackers (0.1%) targeted all honeypots across multiple services
Tor Nodes' Predominant role	✗	It was shown that attacks coming from Tor nodes were statistically insignificant on both providers
Malware Origins	✓	The geographical analysis showed that cyber-criminal activity is global, with significant hotspots in the United States, Singapore, and Hong Kong. The malware download analysis revealed specific malware types, notably Linux/Mirai.Gen, Linux/Malware Downloader, and Perl/Shellbot.NAT Trojan, as the most common
Coordinated Attacks	✓	We have observed patterns indicating the presence of coordinated attacks across providers
Comparison Between AWS and Azure	✓	Azure's susceptibility to SSH attacks is evident from the numerous attempts to log in via SSH and download malware. On the other hand, AWS's primary vulnerability is in the SMB attack domain
Understanding IoT-Specific Malware	✓	The data pointed to a majority of IoT-targeted attacks carried out by one or multiple Mirai-type botnets

Table 6: Summary of Objectives and Findings

10.2 Honeypots: Utility for corporations

In this thesis, honeypots were used to offer insights into the continuous siege faced by all internet-connected devices. However, an important question remains - besides their utility in data collection, can honeypots significantly contribute to protecting critical servers?

The ancient Chinese warrior Sun Tzu once stated, *"All warfare is based on deception"*, and that surprise, rather than confrontation, leads to victory. These concepts also apply to honeypots—they act as a form of deception and serve as an early warning system for corporations, thereby preventing unwelcome surprises.

However, one might question their effectiveness—if every device is constantly under attack, how can a honeypot provide a reliable warning about threats? The answer can be found in an in-depth analysis of attack behaviors and methodologies. For instance, what was discussed in Chapter 6,"*Skill and Strategy: Attack Duration and Attacker Expertise*," could be pivotal in distinguishing the manual attacks on the honeypot from the routine, automated ones. These attackers likely harbor a specific interest in the corporation's network and, thus, pose a greater threat.

By understanding the intricacies of these manual attacks, honeypots can provide an early warning system tailored to address these threats. This makes honeypots an invaluable tool in a corporation's cybersecurity strategy, acting not just as a decoy but also as a detector and analyzer of sophisticated, targeted attacks.

Furthermore, another approach to identifying more dangerous attackers profiled in *Chapter 6.1: Defining Attackers' Profiles* is to deploy multiple honeypots, each strategically positioned close to a different production server across varied networks and locations. If an attacker targets multiple such honeypots, it strongly indicates their potential danger. By adopting this approach, organizations can more effectively sift through many attacks, pinpointing and responding to the most significant threats.

Future Studies

While the present study has established a foundational understanding of attackers' behaviors and patterns within cloud environments, cyber threats' dynamic and expansive nature requires further research. Delving into additional cloud providers distributed across various geographical regions can validate and consolidate this study's findings and pave the way for a more detailed and comprehensive analysis.

References

[1] Christopher Kelly, Nikolaos Pitropakis, Alexios Mylonas, Sean Mckeown, and William Buchanan. A comparative analysis of honeypots on different cloud platforms. *Sensors*, 21, 04 2021.

[2] Clifford Stoll. *The Cuckoo's Egg: Tracking a Spy through the Maze of Computer Espionage*. Simon and Schuster, 2005.

[3] B Cheswick. An evening with berferd in which a cracker is lured, endured, and studied. In *Winter USENIX Conference*, pages 20–24, 1992.

[4] L Spitzner. The honeynet project: Trapping the hackers. *IEEE Security Privacy*, 1:15–23, 2003.

[5] B. Scottberg, William Yurcik, and D. Doss. Internet honeypots: protection or entrapment? pages 387 – 391, 02 2002.

[6] S. Slahor. What is cloud computing. *ProQuest Educ. J.*, 59:10, 2011. [Google Scholar].

[7] Orca Security. Orca security '2023 honeypotting in the cloud report' reveals attackers weaponize exposed cloud secrets in as little as two minutes, 2023. Accessed: 2023-07-27.

[8] Michel Oosterhof. Cowrie ssh/telnet honeypot, 2015.

[9] V. Sethia and A. Jeyasekar. Malware capturing and analysis using dionaea honeypot. In *Proceedings of the 2019 International Carnahan Conference on Security Technology (ICCST)*, pages 1–4, Chennai, India, 2019.

[10] Daniel Fraunholz, Marc Zimmermann, Alexander Hafner, and Hans D. Schotten. Data mining in long-term honeypot data. 2021.

[11] Wes McKinney. pandas: a foundational python library for data analysis and statistics. https://pandas.pydata.org, 2011.

[12] Docker Inc. Docker: Empowering app development for developers. https://www.docker.com, 2013.

[13] Google. Two-factor authentication for linux, 2023.

[14] Inc. MongoDB. Mongodb: The most popular database for modern apps, 2023. Accessed: 2023-07-29.

[15] Prometheus Authors. Prometheus. https://prometheus.io/, 2023.

[16] Grafana Labs. Grafana: The open observability platform, 2021. Accessed: 2023-07-29.

[17] James Osgood. mongodb-grafana. https://github.com/JamesOsgood/mongodb-grafana, 2023.

[18] Nes Cohen. mongodb-grafana. https://github.com/nescohen/mongodb-grafana, 2023. Forked from: James Osgood's mongodb-grafana repository (https://github.com/JamesOsgood/mongodb-grafana).

[19] Tailscale Inc. Tailscale: Secure networks, made simple, 2023. Accessed: 2023-07-29.

[20] MaxMind Inc. Maxmind autonomous system database, 2023. Accessed: 2023-07-30.

[21] Rakshit Agrawal, Jack W. Stokes, Lukas Rist, Ryan Littlefield, Xun Fan, Ken Hollis, Zane Coppedge, Noah Chesterman, and Christian Seifert. Long-term study of honeypots in a public cloud. In *2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume (DSN-S)*, pages 1–6, Baltimore, MD, USA, June 2022. IEEE, IEEE.

[22] StatCounter. Desktop operating system market share worldwide. https://gs.statcounter.com/os-market-share/desktop/worldwide, 2023.

[23] Eclipse Foundation. 2020 iot developer survey results. Technical report, Eclipse Foundation, 2020.

[24] mitre. Cve - cve-2014-7235. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-7235, 2014.

[25] TAbdiukov. mstshash=administr explained. https://github.com/olipo186/Git-Auto-Deploy/issues/221, 2018. Accessed: July 26, 2023.

[26] MITRE Corporation. Cve-2019-0708. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-0708, 2019. Accessed: July 26, 2023.

[27] VirusTotal. Virustotal: Free online virus, malware and url scanner, 2023.

[28] Microsoft Security Intelligence. Description of the worm win32/pykspa.c, 2017.

[29] Elvina Yang. Iot devices with default passwords causing security holes, 2017. Accessed: 2023-07-23.

[30] Brian Krebs. Ddos on dyn impacts twitter, spotify, reddit. 2016.

[31] THETRIS. honeypots activity of the week 50, 2022. honeypot data of December 2022.

[32] NEC Security Technology Center. Login attempts viewed in honeypot, 2023. honeypot data of March 2023.

[33] THETRIS. honeypots activity of the week 49, 2022. honeypot data of December 2022.

[34] jgamblin. Mirai sourcecode. https://github.com/jgamblin/Mirai-Source-Code/blob/master/mirai/bot/scanner.c, 2016.

[35] Vlad Ciuleanu. Mirai malware variants for linux double down on stronger chips in q1 2022. 2022.