WiP: Developing High-interaction Honeypots to Capture and Analyze Region-Specific Bot Behaviors

Jingyang Zou*
jzou17@jh.edu
Information Security Institute
Johns Hopkins University
Baltimore, Maryland, USA

Zhaoxi Sun*
zsun47@jh.edu
Information Security Institute
Johns Hopkins University
Baltimore, Maryland, USA

Chunyen Ku*
cku10@jh.edu
Information Security Institute
Johns Hopkins University
Baltimore, Maryland, USA

Xiangyang Li
xyli@jhu.edu
Information Security Institute
Johns Hopkins University
Baltimore, Maryland, USA

Anton Dahbura
AntonDahbura@jhu.edu
Information Security Institute
Johns Hopkins University
Baltimore, Maryland, USA

ABSTRACT

This project aims to upgrade a high-interaction honeypot system for cyber threat characterization. This paper reports the two efforts being focused on in the project: development of the honeypots and data analysis from their deployment. Several enhancements to the honeypot include integrating additional service ports to attract more traffic and embedding known vulnerabilities for more sophisticated interactions. The honeypot runs a set of common services, such as SSH, Apache, RDP, and email, to simulate a common server for a small company. The honeypot instances have been deployed in three hacker-dense regions of the U.S., East Asia, and Western Europe. Access attempts from the internet to these honeypots are captured for their information including demographics, credentials, and commands being used. They are analyzed to generate useful insights into information-gathering bots active on the internet, such as the patterns in activity intensity and behaviors of potential threats. This paper has shown the effectiveness of the honeypot design and implementation and the rich data these honeypots can collect. This ongoing effort has the potential to greatly improve the capabilities to understand the cyber environment and the threats existing in it.

KEYWORDS

High-Interaction Honeypot, Understanding Cyber Threat, Bot Behavior Analysis

ACM Reference Format:
Jingyang Zou, Zhaoxi Sun, Chunyen Ku, Xiangyang Li, and Anton Dahbura. 2024. WiP: Developing High-interaction Honeypots to Capture and Analyze Region-Specific Bot Behaviors . In *Proceedings of Symposium on the Science of Security (HotSoS '24).* ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

*The authors contributed equally to this research.

HotSoS '24, April 2-4, 2024, Virtual

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

In the current cyber threat landscape, the widespread adoption of the internet and its integration into everyday life has heightened global vulnerability to diverse cyber threats [1]. This situation is exacerbated by the proliferation of IoT and big data, with a study reporting over 15 million attacks, including DDoS, malware, and phishing [2]. In addition, various countries suffer different cyber threats. These attacks range in origin and intent; some are state-sponsored, aiming at critical national infrastructure or secrets, while others are orchestrated by independent hacker groups targeting websites for disruption or defamation. The nature of these attacks varies widely, with some countries experiencing more sophisticated assaults than others. For instance, some countries have developed advanced cyber capabilities for both defensive and offensive purposes, including electronic countermeasures and the development of viruses targeting adversary systems. Reports from various governments, including Germany, the UK, and the USA, indicate a pattern of cyber espionage and attacks, often attributing them to foreign state actors. In another example, hacker groups have targeted critical research facilities, as seen in a case where a Pakistan-based group infiltrated a prominent research center's website to display anti-national messages. These examples underscore the multifaceted and complex landscape of cyber threats faced globally [3]. Therefore, we need to improve our understanding of cyber threats and reinforce network security by implementing multiple defense mechanisms, including honeypots.

Honeypots, designed to attract and detect malicious attacks, have become a crucial tool in network security. From combating advanced botnets [4] to enhancing cloud security [5], honeypots serve varied roles of protection to intelligence gathering that can be deployed in a variety of environments like the Internet of Things and cloud platforms [6][7]. Their diverse designs, such as HoneyDOC [8], and integration with machine learning for malware detection [9], demonstrate honeypots' evolution to address increasingly complex cyber threats.

In this paper, we mainly focus on the following two efforts:

- One is developing an advanced high-interaction honeypot. This enhanced system aims to amplify its attractiveness to potential bots while simultaneously accommodating more sophisticated interactions for richer information. By integrating popular services such as SSH and RDP, the honeypot is designed to mimic real networking environments more closely, appealing to a broader spectrum of cyber threats.
- The other one is capturing and analyzing bot behavior interacting with these honeypots located in different regions. We prioritize our attention on bots for two reasons. Firstly, bots constitute a substantial portion of the current cyber threats. Secondly, complicated ethical concerns might arise if a real human user accesses the honeypots. We study bots' interaction patterns and strategies originating from different geographic regions. Understanding the regional nuances of bot behaviors may enable more targeted and effective defense strategies.

2 RELATED WORK

2.1 Low-interaction vs. High-interaction Honeypots

Based on the characteristics of the interaction, honeypots can be categorized into low-interaction honeypots and high-interaction honeypots.

Low-interaction honeypots simulate only a small set of services like SSH or FTP, and they do not provide any access to the operating system to the attacker. HoneyC [10] stands out as a low-interaction client honeypot, adept in detecting server-based attacks with minimal complexity. HoneyD [11] is a small daemon that simulates thousands of virtual hosts at the same time. The hosts can be configured to run arbitrary services, and their fingerprints can be adapted to spoof scanning tools like Nmap. Glastopf [12], on the other hand, focuses on web application vulnerabilities, enticing attackers with simulated flaws. Lastly, Cowrie [13] serves as an SSH and Telnet honeypot, mimicking a shell environment to study brute force attacks and shell interactions.

However, due to advancements in hackers' skills, low-interaction honeypots have become increasingly easy to identify. To counter this and to capture more extensive hacker activities, high-interaction honeypots are being adopted in place of their low-interaction counterparts [14] [15]. Unlike low-interaction honeypots, high-interaction honeypots can receive a significantly higher volume of packets. For instance, a high-interaction honeypot recorded 1412 script injection attempts in one experiment, compared to approximately 200 attempts by a low-interaction honeypot. The increased data collection by high-interaction honeypots is attributed to their ability to permit real malicious activities. Different high-interaction honeypots exhibit unique characteristics. For example, EMPHAsis boasts a modular and extensible architecture, making it adaptable for various deployment scenarios [16]. To enhance realism, some high-interaction honeypots even incorporate specific vulnerabilities. For instance, a Log4j vulnerability was embedded in one LDAP honeypot, enabling hackers to pivot to directory services via LDAP [17]. This approach effectively collected data on hacker activities exploiting this vulnerability, demonstrating that embedding vulnerabilities can be a strategic move in honeypot design.

In another study [18], the researcher created a Docker-based multi-services honeypot system encompassing SSH, LDAP, and a web service. Utilizing Docker, this honeypot can be effortlessly deployed across various testing environments. This capability is a good demonstration of the modularity and ease of deployment of high-interaction honeypot. In this present paper, we greatly expand upon this work by adding more services and enhancing the connections between these services.

2.2 Data Collection Using Honeypots

Recent studies in data collection using honeypots have made significant efforts to understand cyber threats. In the above paper [18], for example, used a honeypot with SSH service to perform a statistical analysis of common usernames and passwords used by bots, the files they targeted, and the geographical origins of the attack IP addresses. Another study [19] utilized the Dionaea honeypot, which includes several services such as FTP and SIP to trap and analyze malware attacks. In this study, the author tracked Source IP Addresses, Destination Ports, and Timestamps, revealing the high volume of attacks on public-facing IP addresses. Compared with these two studies, our honeypot has more frequently used services, such as RDP and Apache. Owing to our honeypot being deployed in three different hacker-dense regions, we can collect and analyze in-depth activities that represent potential cyber threats.

2.3 Ethical Consideration

The use of honeypots presents not only useful data to help alleviate cyber threats but also a complex ethical dilemma, especially regarding privacy and legal concerns [20]. Honeypots monitor and collect user information, which inevitably raises significant privacy issues. If monitoring activities are completely prohibited to maximize privacy protection, honeypots lose their effectiveness and relevance. Consequently, researchers and security professionals who deploy honeypots must ensure that their methods are legal and ethical, especially when they collect information from real humans.

3 TECHNICAL SOLUTION FOR A HIGH-INTERACTION HONEYPOT

3.1 Architecture

The main purpose of a honeypot is to simulate real production environments to attract attackers to interact with different "services" provided. In doing so, it has the capability to collect and learn from the attackers' interactions and methods. Therefore, the architecture of a honeypot requires careful consideration of requirements. In our project, the primary goal of the honeypot is to record the operations that bots use and gather information on them, such as the credentials being tried to gain access to a service. To achieve this goal, our honeypot contains a full set of common services. At the same time, we use Syslog to log information generated by each service during these interactions. The Structure of the honeypot is shown in Figure 1.

The external part of the honeypot consists of five services: SSH, LDAP, RDP, mail, and Apache. We chose these services because they are frequently targeted due to their widespread use and potential

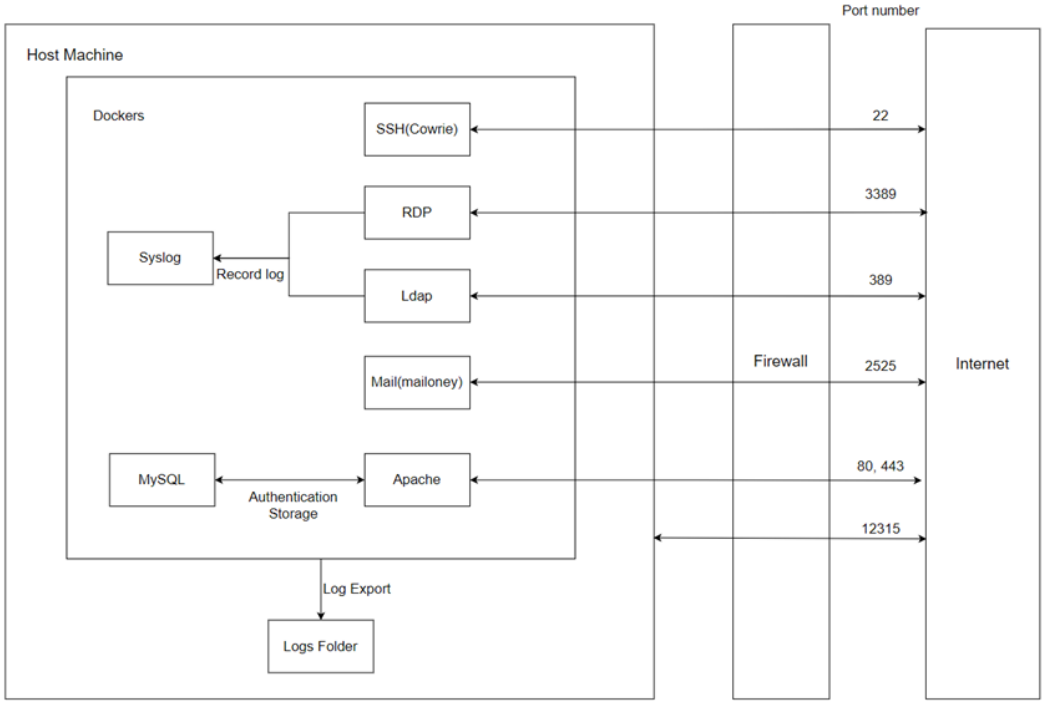

Figure 1: The Honeypot Structure of a List of Common Services and Their Interfaces

misconfiguration. For example, SSH service can capture unauthorized access attempts, and record information such as commands executed by the bot, files accessed, etc.

The honeypot's dockerized environment consists of isolated containers, each running one of the targeted services on standard ports to appear as normal operational servers. A secret port, 12315, is reserved for administrative access, while conventional ports guide bots to the honeypot containers. The Syslog service captures and centralizes the activity logs from each container, enabling us to analyze the tactics and interactions of the bots thoroughly.

3.2 Docker-compose Structure

This section explains the services that make up the honeypot. We focus on the Dockerfile for the build and additional configuration files.

We chose to use Docker-compose for unified deployment because it simplifies the deployment of services through its docker-compose.yml file, which defines services, networks, and volumes. In the docker-compose.yml file, there are three main information sections: services, networks, and volumes. The services section defines the different containers that we want to deploy. The networks section defines how containers communicate with each other, allowing inter-container interaction and including settings like IP address assignment. The volumes section is for data persistence and sharing between the host and the containers.

For the SSH service, we have mapped the traffic from port 22 to port 2222; this mapping allows the SSH docker to intercept SSH traffic; we have also assigned a static IP address within the defined docker network, ensuring consistent network addressing. The Syslog, Apache, MySQL, and RDP configurations are similar to the SSH service, except we assigned different static IP addresses. The Mailoney service is listening on port 2525, an alternative SMTP port. It employs a volume mount (./mailoney_logs:/var/log/mailoney) to persistently store logs, allowing for analysis and monitoring of interactions. The LDAP service is listening on port 389. A volume mount from ./ldap/logs to /var/log/syslog-ng in the container is configured for logging interactions and events. Additionally, it uses a JSON file logging driver with a maximum size of 200KB per file and a limit of 10 files, ensuring efficient log management and rotation.

3.3 Services

Our team has deployed Cowrie [13], an open-source SSH honeypot. It is specifically designed to mimic real SSH services, offering a medium-to-high-interaction environment that effectively deceives and engages bots. In its original configuration, Cowrie is designed to accept any password for the 'root' user login, a feature that increases its attractiveness to bots. The honeypot's interactive sessions, which respond in real-time, further contribute to the illusion of a compromised system, keeping the bot engaged longer and providing more extensive data for analysis. In our project, we have adapted Cowrie to launch within Docker, streamlining its integration and deployment within our honeypot framework.

Our team crafted a custom Docker container for RDP. We strategically incorporated various packages, including the older version Xfce4, introducing known vulnerabilities, such as CVE-2022-45062 (https://cve.mitre.org/). We have put the RDP docker within the

same network as other services in our honeypot, so that the bot can use the RDP service to access other services. There are two main special features of this RDP service. One is that it interacts with Syslog and can log all the bots' behavior. The other one is we are using Linux Pluggable Authentication Modules to configure the login process, so that any password can be used to log in successfully as long as the username is root.

We adapted a component from an open-source honeypots project [21] , hosted on GitHub, to develop a customized LDAP service using Python. This custom service is designed to record every login attempt. Furthermore, it logs the bot's exact credentials, granting us a window into their methods. Our simulation can also actively respond to these login attempts.

Our team deployed Mailoney [22], a honeypot that mimics real mail servers to divert and monitor malicious activities and collect phishing emails. This honeypot utilizes the Python smtpd module's 'SMTPServer' class to simulate SMTP operations so that the honeypot can log all the phishing emails bots send.

In our project, we built a simple blog site using Apache 2.4.49. This version has vulnerability CVE-2021-41773 (https://cve.mitre.org/). Also, the site contains basic functionality such as user login and blog browsing and does not handle errors so that bots can see more valid information. At the same time, our Apache service and MySQL service are linked together; blogs, users, and other information are stored in the MySQL service, which is easy to interact with. We use a blog.sql file to initialize MySQL service, including importing the initial blog and user information. In addition, the reason for using MySQL service is its ability to introduce vulnerabilities such as SQL injection, XSS injection, and so on.

4 DATA COLLECTION AND ANALYSIS OF BOT INTERACTIONS

4.1 Deployment of Honeypots

All honeypots utilized for this project are hosted on Microsoft Azure. These honeypots are deployed to three different regions, which are East Asia (EA), West Europe (WE), and East United States (EUS). The operating system used for all hosts is Ubuntu server 20.04, while the honeypots deployed in all three regions are identical, ensuring no differences in the data collected due to honeypot configuration. At the same time, we also configured the firewall to limit inbound traffic and outbound traffic. We only allowed traffic that targets the port number of the service. We also turned on the response to pings to increase the likelihood that the hacker would discover the IP address. We have limited outbound traffic to ensure that the bot cannot use our honeypot to launch attacks on others on the Internet. Honeypots were deployed in three regions for a duration of 20 days in the fall of 2023, during which we collected a substantial amount of data. This data includes common credentials, commands, files, and others.

During the deployment, we prioritize our attention on bots for two significant reasons. Firstly, bots constitute a substantial portion of the current cyber threats. Secondly, the project took a great deal of effort to prevent potential ethical concerns that might arise if a real human user inadvertently accesses our honeypot.

We have implemented several measures to address these ethical considerations and reduce the risk of unintentional human access. First, we have placed banners in prominent locations within the honeypot's services, such as the desktop of RDP, to alert any genuine users that they have entered a honeypot environment. Second, for SSH services, we have set up banners to be displayed either before (pre-login) or after (post-login) an interaction, providing an additional warning level. Lastly, we have opted to use only IP addresses to prevent users with non-malicious intent from accessing our honeypot via URLs. We have deliberately refrained from assigning domain names and certificates. These measures collectively ensure our honeypot's ethical compliance and security, while effectively focusing on the analysis and deterrence of malicious bot activities.

The rest of this section presents a region-based analysis of the collected data so far, considering each service's unique characteristics. However, the results do not include LDAP and mail services, due to the very small amount of data collected. For the LDAP service, this difference can be attributed to LDAP's role as a collaborated component within broader systems, leading to less standalone visibility. Meanwhile, the mail service's use of Port 2525, a non-standard port, likely results in lower malicious activity. Hackers usually do not target this port, reducing its vulnerability.

4.2 SSH Traffic

4.2.1 Traffic Volume with IP comparison: In Figure 2, the data spans from October 24 to November 14, presenting daily activity across the three servers. A key observation is that on certain days the log counts are significantly higher than the IP counts, suggesting numerous login attempts from a smaller number of IPs, which often implies that bots may be using automated scripts or bots to attempt to gain access. For instance, on November 3rd, the East Asia server recorded an exceptionally high log count of 535,502 against just 24 IPs. This disparity indicates either an aggressive brute-force attack or a possible DDoS attack effort, aiming to overwhelm the server with traffic.

The data also shows sporadic peaks in activity, such as on November 6th for the EA server, with a high log count (726,704) from 46 IPs, implying a concentrated effort, possibly from a botnet. In contrast, the WE server peaked on November 8th with 110,223 logs from 180 IPs, indicative of broad interactions. These patterns suggest different operational strategies, ranging from focused attempts in EA to more distributed activities in WE.

In conclusion, the data reflects that the East Asia server is subject to more intense and focused operations, while the Western Europe server experiences high volumes of traffic from more distributed sources. The Eastern United States server shows a more balanced access pattern, which may point to a different risk profile from the other two. Understanding these trends is vital for cybersecurity teams to tailor their defense mechanisms appropriately.

4.2.2 Comparison based on country: Table 1 shows a few common trends emerging in all three regions where we deployed our honeypots. Notably, China, South Korea, and the United States consistently appear in the top three positions across these regions, with China registering 308 in WE and 274 in EUS, South Korea with 209 in WE and 268 in EUS, and the United States reporting 245 in WE and 218 in EUS. This consistency indicates their significant presence in cyber activities across these regions, likely driven by

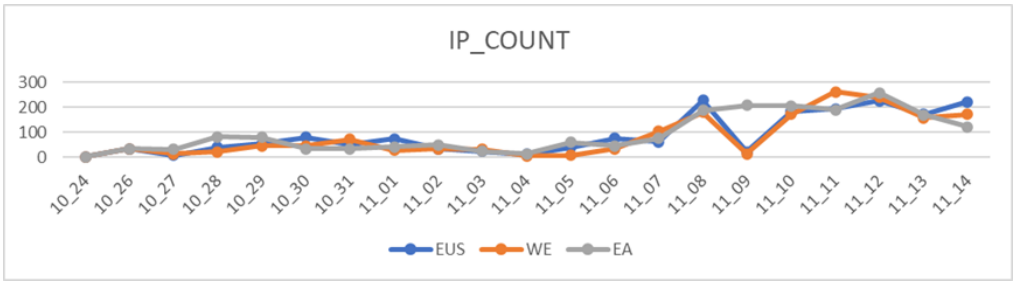

Figure 2: Daily Change of IPs to Connect to the SSH Service

the technological capabilities and resources available in these countries, along with possible geopolitical motivations driving these intrusions.

However, distinct regional differences are also evident, likely influenced by political factors. China leads in intrusion attempts in EUS (274) and WE (308), but is third in EA (241). This variation may be influenced by China's political strategies or regional dynamics within Asia. Similarly, Russia's higher intrusion attempts in EUS (101), compared to WE (78) and EA (80), suggest political motivations targeting the Eastern United States, possibly reflecting geopolitical interests or strategies. India's increased activity in EA (95) compared to EUS (78) and WE (64) might also be driven by political tensions with neighboring countries, particularly China, influencing its cyber intrusion strategies.

These patterns demonstrate the complex interplay of technological capabilities, geopolitical interests, and regional dynamics in global cyber activities. Understanding these aspects is crucial for developing effective cyber defense strategies and international cooperation to counter these threats.

4.2.3 Region-based credential analysis: According to Table 2, the considerable match between user credentials in Cowrie honeypot incidents and entries in password wordlists like rockyou.txt and sqlmap.txt—37.9% for accounts and 49.7% for passwords in the former, and 41.7% for accounts and 46.3% for passwords in the latter—signals that bots frequently exploit well-known weak passwords. This underscores the urgent need for stringent password policies and the abandonment of default credentials to bolster security against threats.

Globally, the trend toward exploiting common passwords such as "123456," "password," and "123" exposes a universal security lapse, as demonstrated in Table 3. Frequent attempts on default accounts like "root," "admin," and "ubuntu" suggest a broad, opportunistic approach, favoring automated attacks over more sophisticated methods, a trend further illustrated in Table 4.

Regionally, the high frequency of "root" access attempts in East Asia might indicate local security weaknesses or a high occurrence of systems with unchanged default settings. The recurrence of the unique account "345gs5662d34" in various regions points to shared attack tools and methods, enabling intruders to exploit the same vulnerabilities internationally. This pattern calls for a unified defense approach emphasizing strong password management, eradicating default settings, and advanced detection systems. Additionally, educating users and advancing defensive technologies are vital in addressing these persistent threats, while also acknowledging the need for region-specific security protocols within a global cybersecurity framework.

4.2.4 Temporal analysis: The Cowrie honeypot logs indicate distinct operational strategies that reflect regional behaviors and motivation variations, as shown in Figure 3. In WE, there is a noticeable fluctuation in access attempts during the morning, potentially exploiting the start of the business day when the attention of defense drops to a lower level. The attempts escalate in the evening, peaking at around 23:00, suggesting a tactic to take advantage of reduced vigilance likely during late hours. Conversely, EA experiences a surge in traffic between 11:00 and 15:00, which may correspond with peak business activities, followed by a dip in late afternoon and then a significant increase in evening peaking at 22:00. This could reflect a strategic pause for bots to assess and modify their tactics before the off-peak hours.

The EUS data exhibits a more consistent and even pattern during the day, followed by a rise in late evening hours. This could indicate that bots are capitalizing on the end of the day when active monitoring wanes, possibly synchronizing with morning hours in other regions, suggesting a continuous pressure on targets across time zones. Collectively, these patterns demonstrate a level of planning by bots that consider the daily operational rhythms of their targets. Bots in WE seem to favor late evenings and the transition times during the day, while in EA, there is a focus on peak business hours and evenings after a mid-afternoon lull. In contrast, in EUS the bots maintain a persistent threat throughout the day, with increased activity in the late evening hours. These insights underline the importance of adaptive cybersecurity measures that can respond to the predictable rhythm of business cycles and the tactical timing of cyber threats that are differentiated by regions.

4.2.5 Command analysis: In the analyzed Cowrie honeypot logs, bots frequently used commands to manipulate system security settings and gain persistent unauthorized access.

The command sequence below is used to change file attributes in the .ssh directory, the hub for SSH access keys, suggesting an attempt to remove immutability and append access keys for future entries.

- `cd ~`
- `chattr -ia .ssh`
- `lockr -ia .ssh`

Table 1: Top 5 Countries with Most Connection Attempts to the SSH Service

East Asia		Western Europe		Eastern United States	
Country	Total	Country	Total	Country	Total
United States	292	China	308	China	274
South Korea	270	United States	245	South Korea	268
China	241	South Korea	209	United States	218
Singapore	127	Singapore	105	Singapore	123
India	95	Russia	78	Russia	101

Table 2: Overall Wordlist Proportion Match in Attempts to the SSH Service

Word list	Account(%)	Password(%)
rockyou.txt	37.9	49.7
nmap.lst	8.33	7
john.lst	9.43	6.7
metasploit	1.2	2.53
sqlmap.txt	41.7	46.3

Table 3: Frequently Used Passwords in Attempts to the SSH Service

East Asia		Western Europe		Eastern United States	
Account	Count	Account	Count	Account	Count
123456	3504	123456	2903	123456	3065
123	1085	123	726	password	675
345gs5662d34	878	3245gs5662d34	552	123	658
3245gs5662d34	873	345gs5662d34	548	3245gs5662d34	505
password	844	password	538	345gs5662d34	503

Table 4: Frequently Used Accounts in Attempts to the SSH Service

East Asia		Western Europe		Eastern United States	
Account	Count	Account	Count	Account	Count
root	47518	root	3052	root	2739
admin	1607	admin	1527	admin	1392
ubuntu	1289	ubuntu	1411	ubuntu	1022
test	888	345gs5662d34	548	345gs5662d34	503
345gs5662d34	878	test	545	user	427

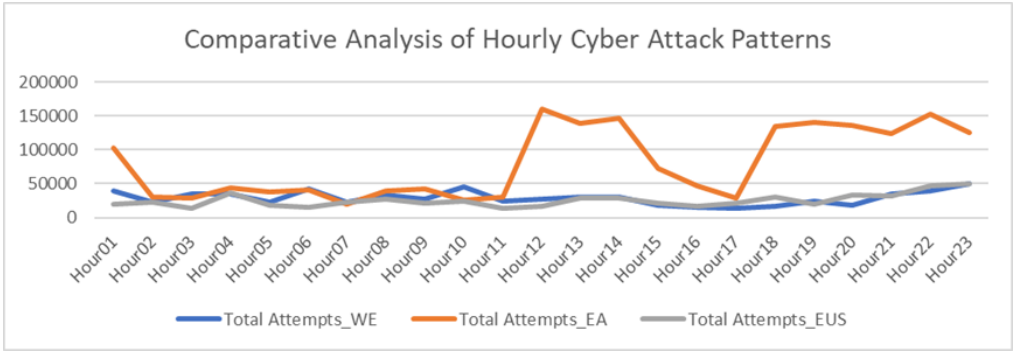

Figure 3: Hourly Change in Connection Attempts to the SSH Service

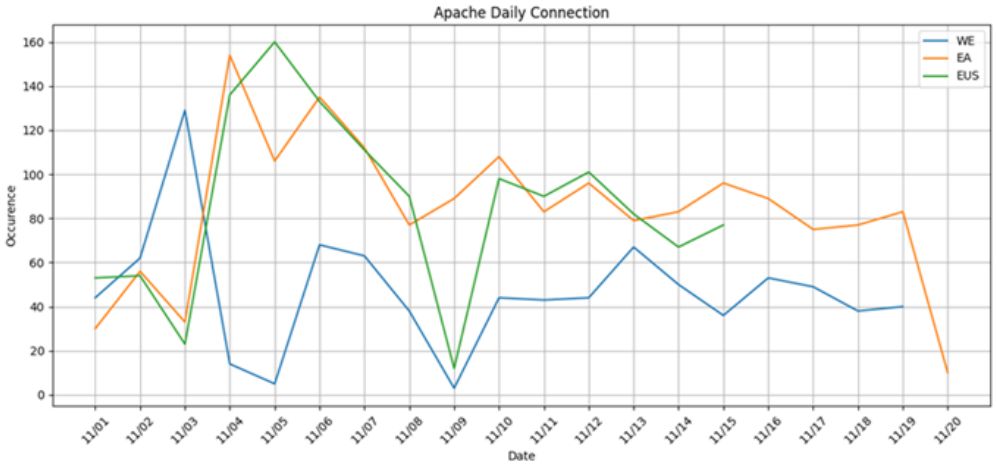

Figure 4: Daily Change of IPs to Access the Apache Service

Table 5: Top 5 Countries with Most Connection Attempts to the Apache Service

East Asia		Western Europe		Eastern United States	
Country	Total	Country	Total	Country	Total
United States	576	United States	461	United States	470
China	93	United Kingdom	33	China	75
United Kingdom	47	France	25	Singapore	37
Singapore	47	Russia	23	Netherlands	37
Netherlands	37	Germany	22	United Kingdom	36

Another set of commands appears to be a more aggressive approach to resetting the SSH configuration, removing all previous keys, and inserting a new one, granting the bot subsequent access. The set of commands are:

- `cd ~`
- `rm -rf .ssh`
- `mkdir .ssh`
- `echo "ssh-rsa AAAAB...">>.ssh/authorized_keys`
- `chmod -R go= ~/.ssh`
- `cd ~`

The execution of a mysterious script or binary, `./oinasf`, followed by attempts to read and display the system's executable content, indicates a probing strategy for vulnerabilities or valuable information. The use of `/ip cloud print` suggests that bots target MikroTik routers to access or disrupt cloud-based services, while `uname -s -m` provides them with essential details about the operating system and machine architecture, valuable for crafting further actions tailored to the system's specifics. In conclusion, these commands represent a clear strategy to infiltrate, assess, and establish control over targeted systems. They emphasize the bots' preference for direct manipulation and sustained access, highlighting the critical need for robust defenses against such common yet potentially devastating tactics.

4.3 Apache Traffic

In a distributed cloud computing setup, multiple servers run Apache services accessible via port 80. In Figure 4, the network consists of three honeypots, identified by IP addresses and geographical locations: East Asia, Western Europe, and the Eastern United States. The analysis of daily connections from November 1st to November 20th shows fluctuations in connection counts for all three regions, with EA consistently having the highest connection count, possibly making it a target for frequent cyberattacks. EUS and WE experienced simultaneous connection lows on November 9th, corresponding to a major holiday in the US. From November 10th to 19th, all the regions maintained relatively stable connection levels, possibly signifying a period of normalcy following earlier fluctuations.

Table 5 shows that Apache honeypot indicates a consistent dominance of the United States among the top 5 countries in server connections, with approximately 500 connections originating from the U.S., significantly higher than other countries. This imbalance is primarily attributed to the concentration of servers within the U.S., as it is often the default server location for many cloud service providers, leading to more connections.

Aside from the United States, many countries, including China, the United Kingdom, Singapore, the Netherlands, Germany, France, India, and Russia, consistently appear across Apache service. Several factors contribute to their frequent interaction with the honeypot. Firstly, these countries have large internet user populations, resulting in increased online interactions across various platforms,

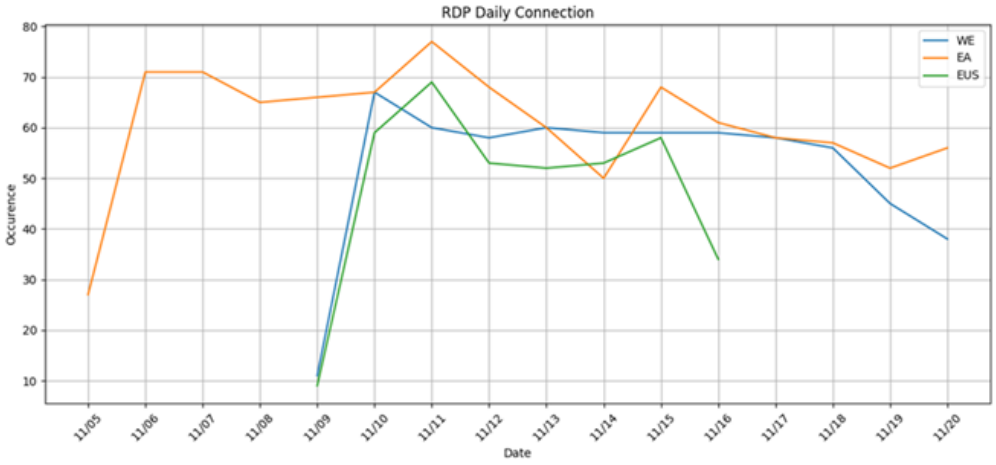

Figure 5: Daily Change of IPs to Access the RDP Service

Table 6: Top 5 Countries with Most Connection Attempts to the RDP Service

East Asia		Western Europe		Eastern United States	
Country	Total	Country	Total	Country	Total
United States	284	United States	200	United States	132
Russia	38	Russia	26	Russia	22
China	34	China	22	Germany	22
Germany	27	Germany	20	China	17
Vietnam	19	United Kingdom	14	United Kingdom	9

including honeypots. Additionally, these countries host diverse server infrastructures, ranging from private enterprise servers to public cloud services, which are crucial in their interactions with honeypots. These servers have widespread geographic distribution and density and are part of global networks actively scanning and interacting with internet addresses. Therefore, they are easy to find in collected data using a honeypot.

4.4 RDP Traffic

The data in the Figure 5 depicts daily RDP connections from three honeypots in different regions: West Europe, East Asia, and the East United States. Notably, the honeypot in East Asia consistently records a higher daily RDP connection count compared to its West European and East United States counterparts, suggesting a stronger demand for remote desktop services in East Asia.

Interestingly, all three honeypots exhibit a similar connection pattern. There is a notable increase in connectivity observed on the initial day of server deployment, possibly due to bots requiring approximately 24 hours to detect the availability of a newly exposed RDP service on the internet. This surge indicates the successful attraction of various entities, potentially including automated scripts, bots, and other sources of suspect traffic, to the honeypot.

The table 6 detailing the top 5 countries connecting to the respective RDP service reveals a trend. The United States consistently emerges as the top origin, with around 200 connections, which significantly overshadows the connection counts from other countries. This characteristic of RDP service usage closely mirrors the patterns observed in the usage of Apache service. In both cases, one country has a clear dominance in connection numbers, reflecting similar usage trends and possibly pointing to broader technological or infrastructural trends in these services.

5 CONCLUSION

We have developed a high-interaction honeypot covering various services including SSH, SMTP, RDP, LDAP, Apache, and MySQL. We have implemented the system on the Azure platform to capture information on access attempts to these services. For data collection, we strategically deployed identical honeypots in three hacker-dense regions: East United States, East Asia, and West Europe.

After analyzing the collected information, we observe a significant variance in the data collection capabilities of these services. Some services, like SSH and Apache, have proven adept at gathering extensive information. However, it is important to note that the interactions detected thus far consist of mostly bot activities rather than sophisticated hacker maneuvers. In deployment, we purposely avoided the interactions with real human users due to ethical concerns.

The preliminary progress has several limitations. Our effort to uncover complex hacker behaviors, such as the captured interactions with the LDAP and mail services and the transitioning between services or exploiting embedded vulnerabilities, has been limited. This

limitation is partly due to time constraints. It is also significantly impacted by the absence of a supporting digital presence, such as a domain name. Not using a domain name is a good way to stop casual users from accidentally entering our honeypot. However, it also reduce the honeypot's visibility for human hackers to discover our system and make it less attractive to them. This indicates that while effective in certain aspects, the honeypot configuration may require significant enhancements to capture more nuanced and intricate hacker strategies.

In our future work, enhancing the efficacy of our honeypot is a multifaceted endeavor. Crucially, acquiring a digital certificate and a domain name, with integration into popular web servers, could greatly increase our ability to attract and analyze more sophisticated hacker activities. Such advancements are not mere upgrades but are essential in transforming our honeypot into a more powerful tool for understanding the evolving cyber landscape. This also poses challenges for us to handle the ethical implications of such interactions. By persistently refining our strategies and infrastructure, hopefully, we can gain deeper and new insights into the tactics of hackers and cyber threats.

REFERENCES

[1] Yuchong Li and Qinghui Liu. A comprehensive review study of cyber-attacks and cyber security; emerging trends and recent developments. *Energy Reports*, 7:8176–8186, 2021.

[2] Andreea Bendovschi. Cyber-attacks–trends, patterns and security countermeasures. *Procedia Economics and Finance*, 28:24–31, 2015.

[3] Kathan Patel and Dhaval Chudasama. National security threats in cyberspace. *National Journal of Cyber Security Law*, 4(1):12–20p, 2021.

[4] Ping Wang, Lei Wu, Ryan Cunningham, and Cliff C Zou. Honeypot detection in advanced botnet attacks. *International Journal of Information and Computer Security*, 4(1):30–51, 2010.

[5] Poorvika Singh Negi, Aditya Garg, and Roshan Lal. Intrusion detection and prevention using honeypot network for cloud security. In *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 129–132. IEEE, 2020.

[6] Javier Franco, Ahmet Aris, Berk Canberk, and A Selcuk Uluagac. A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems. *IEEE Communications Surveys & Tutorials*, 23(4):2351–2383, 2021.

[7] Christopher Kelly, Nikolaos Pitropakis, Alexios Mylonas, Sean McKeown, and William J Buchanan. A comparative analysis of honeypots on different cloud platforms. *Sensors*, 21(7):2433, 2021.

[8] Wenjun Fan, Zhihui Du, Max Smith-Creasey, and David Fernandez. Honeydoc: an efficient honeypot architecture enabling all-round design. *IEEE journal on selected areas in communications*, 37(3):683–697, 2019.

[9] Iik Muhamad Malik Matin and Budi Rahardjo. Malware detection using honeypot and machine learning. In *2019 7th international conference on cyber and IT service management (CITSM)*, volume 7, pages 1–4. IEEE, 2019.

[10] Christian Seifert, Ian Welch, Peter Komisarczuk, et al. Honeyc-the low-interaction client honeypot. *Proceedings of the 2007 NZCSRCS, Waikato University, Hamilton, New Zealand*, 6:48, 2007.

[11] Niels Provos. Honeyd-a virtual honeypot daemon. In *10th dfn-cert workshop, hamburg, germany*, volume 2, page 4, 2003.

[12] LMushorg. Glastopf. https://github.com/mushorg/glastopf, 2014. GitHub repository.

[13] Cowrie. Cowrie. https://github.com/cowrie/cowrie, 2015. GitHub repository.

[14] Jérémy Briffaut, Jean-François Lalande, and Christian Toinard. Security and results of a large-scale high-interaction honeypot. *J. Comput.*, 4(5):395–404, 2009.

[15] Angelo Furfaro, Francesco Lupia, and Domenico Saccà. Gathering malware data through high-interaction honeypots. In *SEBD*, pages 286–293, 2020.

[16] Elang Dwi Saputro, Yudha Purwanto, and Muhammad Faris Ruriawan. Medium interaction honeypot infrastructure on the internet of things. In *2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)*, pages 98–102. IEEE, 2021.

[17] Shreyas Srinivasa, Jens Myrup Pedersen, and Emmanouil Vasilomanolakis. Deceptive directories and "vulnerable" logs: a honeypot study of the ldap and log4j attack landscape. In *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 442–447. IEEE, 2022.

[18] Alexandre Khalfallah. A high interaction docker-based honeypot. *MSSI Capstone Report*, 2023.

[19] Chris Moore and Ameer Al-Nemrat. An analysis of honeypot programs and the attack data collected. In *Global Security, Safety and Sustainability: Tomorrow's Challenges of Cyber Security: 10th International Conference, ICGS3 2015, London, UK, September 15-17, 2015. Proceedings 10*, pages 228–238. Springer, 2015.

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

[21] Qeeqbox. Qeeqbox/honeypots: 30 different honeypots in one package. https://github.com/qeeqbox/honeypots, 2021.

[22] phin3has. Phin3has/mailoney: An smtp honeypot. https://github.com/phin3has/mailoney, 2015.

[23] Yekta Kocaogullar, Orcun Cetin, Budi Arief, Calvin Brierley, Jamie Pont, and Julio C Hernandez-Castro. Hunting high or low: evaluating the effectiveness of high-interaction and low-interaction honeypots. 2022.

[24] Marcin Nawrocki, Matthias Wählisch, Thomas C Schmidt, Christian Keil, and Jochen Schönfelder. A survey on honeypot software and data analysis. *arXiv preprint arXiv:1608.06249*, 2016.

[25] Lukás Zobal, Dusan Kolár, and Jakub Kroustek. Exploring current e-mail cyber threats using authenticated smtp honeypot. In *ICETE (2)*, pages 253–262, 2020.

[26] VS Devi Priya and S Sibi Chakkaravarthy. Containerized cloud-based honeypot deception for tracking attackers. *Scientific Reports*, 13(1):1437, 2023.

[27] Marwan Abbas-Escribano and Hervé Debar. An improved honeypot model for attack detection and analysis. In *Proceedings of the 18th International Conference on Availability, Reliability and Security*, pages 1–10, 2023.

[28] Nikita M Danchenko, Anton O Prokofiev, and Dmitry S Silnov. Detecting suspicious activity on remote desktop protocols using honeypot system. In *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pages 127–128. IEEE, 2017.