RESEARCH BY
BYUNGJI KANG

L3montree
Cybersecurity

JANUARY 2025

CYBER-RECONNAISSANCE & CREDENTIAL THEFT

USING HONEYPOTS IN MODERN THREAT LANDSCAPES

CONTACT

+49 (0)228 92998568
L3MONTREE.COM
INFO@L3MONTREE.COM

Oh-My-Honeypot GitHub Repository

Lightweight, containerized honeypot for detecting port scans and login attempts (incl. password and user name). API with statistics (MIT License).

Collected dataset

The data was collected over the period of one month in a variety of cloud providers. The data is in csv format and sql dump available.
The Dataset is free to use (MIT License).

L3montree Website

DevSecOps made in Germany. We are securing your Software Supply Chain and Cloud-Native Environment. Consulting and Hands-on implementation in secure software development, cloud native- and open source security

CYBERRECONNAISSANCE & CREDENTIAL THEFT

USING HONEYPOTS IN MODERN THREAT LANDSCAPES

Authors

Byungji Kang
Bonn-Rhein-Sieg University of Applied Sciences,
L3montree Cybersecurity Associate

Tim Bastin (Supervisor)
Senior Software Security Specialist, L3montree Cybersecurity

Acknowledgements

This research was conducted by Byungji Kang as his final thesis for the Bachelor of Science degree at the Bonn-Rhein-Sieg University of Applied Sciences.

He was supervised by employees of the company L3montree Cybersecurity, Prof. Dr. Markus Ullmann (BSI, H-BRS examiner) and Prof. Dr. Petra Haferkorn (H-BRS, examiner).

L3montree Cybersecurity would like to thank Mr. Byungji Kang for preparing this study.

TABLE OF CONTENT

ABSTRACT

This research focuses on the identification and analysis of the initial stages of a cyberattack in modern threat landscapes.

The methodology involves the generation of hypotheses based on threat intelligence reports and open-source intelligence (OSINT). Honeypots were deployed across various cloud platforms and geographic regions to gather raw data on attack behaviors.

The collected data is then analyzed to identify patterns in attack target selection and the types of attacks. Furthermore, the study investigates the relationship between targeted regions or cloud platforms and the specific attack techniques employed.

The objective of this approach is to generate actionable cyber threat intelligence and provide significant insight into cyberattacks.

The findings from the honeypot analysis serve to validate the proposed hypotheses, thereby providing deeper insights into the behaviour of the attackers and informing the formulation of more proactive cybersecurity strategies. The findings of research contribute to a more detailed and sophisticated understanding of the nature of cyber threats.

INTRODUCTION

Recent years have seen an increase in cyberattacks aimed at data breaches, resulting in financial losses for both businesses and individuals. The global average cost of a data breach increased by 10% over the previous year 2023 to approximately $4.88 million in 2024.

Beyond the immediate financial impact, these cyberattacks have a substantial time cost. Identifying and containing a breach can take months or even a year, severely disrupting business operations. On average, it takes 194 days to identify a breach and 258 days to contain it. This long recovery process uses up resources and leads to delays, lowers productivity, and financial losses.

The primary initial attack vectors for data breaches include stolen or compromised credentials (16%), phishing (15%) and cloud misconfigurations (12%). Credential-based attacks, in particular, take the longest to detect and contain, with an average of 292 days to identify and 287 days to fully contain the breach. This extended timeline occurs because it is more difficult for defenders to distinguish between legitimate user activity and malicious actions[1].

Cyberattack detection technologies have evolved to meet these challenges. Detecting threats in the early stages of an attack can reduce the potential damage, making early intervention critical[2].

To neutralize these risks as quickly as possible, early threat detection and mitigation has become a priority for businesses and organizations.

To address those risks, organisations use intrusion detection systems (IDS) like *Falco* or *Wazuh*. Those tools allow security teams to detect potential threats as they occur by using network and system activity monitoring and trying to detect suspicious behavior. Besides that, intrusion prevention systems are used to mitigate attacks in realtime or to block the attack in the first place. However, both

[1] I. Security. *Cost of a Data Breach Report 2024.*

[2] F. E. Office. *"Cyber Attack Countermeasure Technologies Using Analysis of Communication and Logs in Internal Network".*

systems are largely reactive, relying on predefined patterns (called signatures), of already known threats. They usually struggle to deal with emerging threats such as zero-day vulnerabilities and advanced persistent threats (APTs) that lack known signatures, leaving organizations vulnerable to sophisticated or unknown attacks[3].

Modern cybersecurity strategies are increasingly exploring various methods to shift from reactive to proactive approaches, focusing on prevention rather than response. To enable this proactive defense, threat intelligence has emerged as a crucial component. Threat intelligence is evidence-based knowledge generated from a variety of sources, including internet forums, social media, research reports, and public databases, as well as data from the dark web and deep web, where malicious activities can be tracked through hacker forums or marketplaces.

Another valuable source for threat intelligence is a honeypot, which plays a critical role in cybersecurity. A Honeypot is an application designed to appear as vulnerable target, attracting cyber attackers and capturing detailed information about their tactics, techniques and procedures (TTPs). Unlike other security systems, honeypots have the unique ability to collect raw data directly from attackers, offering unfiltered insights into their behavior. This makes them an invaluable tool for enhancing threat intelligence by providing a more comprehensive view of potential threats[4].

[3] X. Yang, J. Yuan, H. Yang, Y. Kong, H. Zhang, and J. Zhao. *"A highly interactive Honeypot-Based approach to network threat management"*.

[4] H. Almohannadi, I. Awan, J. Al Hamar, A. Cullen, J. P. Disso, and L. Armitage. *"Cyber Threat Intelligence from Honeypot Data Using Elasticsearch"*.

Objective of Research

The objective of this research is to examine the early-stage of cyberattacks and to generate actionable intelligence based on the data collected through honeypots.

Initially, Hypotheses were formulated based on an analysis of multiple sources, including the latest published threat intelligence reports and OSINT. The selection of honeypot deployment locations and cloud services is also informed by these sources. Subsequently, the collected data is analyzed with a focus of identifying the attack patterns and the specific types of attackers involved. These attack types were subsequently modelled using frameworks such as the Cyber Kill Chain and Tactics, Techniques, and Procedures (TTP), thereby facilitating the generation of meaningful insights.

Furthermore, the collected threat intelligence is validated through the proposed hypotheses, ensuring the accuracy and relevance of the insights gained. This approach is designed to deepen the understanding of attacker behavior and reproduce valuable intelligence, which will contribute to the development of more proactive and effective security strategies.

BACKGROUND

This section outlines key terms and concepts essential for understanding the research.

Honeypots

Honeypots are specialized cybersecurity tools designed to attract attackers by mimicking vulnerable systems or services. Their purpose is to lure malicious actors into interacting with these decoys, allowing security teams to closely monitor and analyze the attacker's behavior.

The insights gained from honeypot interactions provide a detailed understanding of the attacker's methods, techniques, and goals, making it a highly effective tool for gathering raw intelligence. By capturing unfiltered data, honeypots offer unique visibility into how attackers operate, including the discovery of new vulnerabilities and attack vectors that may not yet be documented.

From the adversary's perspective, honeypots seem to offer valuable information and real services. However, these are fake, designed to study and capture the attacker's behavior. A key benefit of using honeypots is that they create uncertainty for attackers about the value of the data they steal. This confusion often makes attackers more active, giving security teams more chances to gather useful information about their methods.

Additionally, attackers waste time and resources on these decoys, which keeps them away from more important parts of the network. Knowing they might be deceived can also make attackers think twice before launching further attacks, as it increases their sense of risk[5].

[5] A. Javadpour, F. Ja'fari, T. Taleb, M. Shojafar, and C. Benzaïd. *"A comprehensive survey on cyber deception techniques to improve honeypot performance"*.

Through the use of honeypots, organizations can enhance their threat intelligence and refine defensive strategies, making them a key component in a modern, proactive cyber-security framework[6].

Cyber Threat Intelligence and Open Source Intelligence

Cyber Threat Intelligence (CTI) can be defined as evidence-based knowledge that includes indicators, mechanisms, consequences, and actionable steps related to current or emerging cyber threats to organizational assets. The generation process begins by gathering raw data, such as IP addresses, domain names, or network activity logs. This data is then enriched, providing basic context by addressing questions like who, what, where, and when.

However, this information alone lacks the depth required for effective decision-making. When further analyzed and combined with historical data, incident reports, or attack patterns, it becomes intelligence. Unlike basic information, intelligence answers why and how, offering insights into the motivations, tactics, and potential outcomes of threats[7].

This transformation of raw data into actionable intelligence enables informed decisions and proactive responses to emerging threats. Organizations can enhance their preparedness for potential threats, allowing them to take proactive measures to prevent attacks[8].

Open Source Intelligence (OSINT) refers to the process of gathering information from publicly available sources that are legally accessible. OSINT can include data from a variety of mediums such as online publications, social media, forums, news outlets, and publicly available government reports. The primary goal of OSINT is to collect and analyze information that helps organizations or individuals

[6] L. Zobal, D. Kolar, and R. Fujdiak. *"Current State of Honeypots and Deception Strategies in Cybersecurity"*.

[7] Torres, A.E., Torres, F., Budgud, A.T. *"Cyber Threat Intelligence Methodologies: Hunting Cyber Threats with Threat Intelligence Platforms and Deception Techniques"*.

[8] P. Alaeifar, S. Pal, Z. Jadidi, M. Hussain, and E. Foo. "Current approaches and future directions for Cyber Threat Intelligence sharing: A survey".

understand the threat landscape without violating legal or ethical boundaries.

OSINT is particularly valuable in cybersecurity because it allows analysts to track emerging trends, monitor attacker behavior, and assess potential risks from external sources. For example, threat actors often share details of their attacks, tools and techniques on hacker forums or social media platforms, which can be used to predict future threats. Additionally, OSINT helps in gathering Indicators of Compromise (IoCs), such as IP addresses or domain names, that are being used in malicious activities. This intelligence can be integrated into threat detection systems to enhance an organization's ability to detect and respond to potential cyberthreats.

One of the key advantages of OSINT is its cost-effectiveness, as it leverages freely available information rather than relying on proprietary databases or closed-source intelligence feeds. However, the vast amount of data that OSINT generates requires careful analysis and filtering to ensure accuracy and relevance. By using automated tools and manual analysis, OSINT can provide critical insights into adversarial tactics and potential vulnerabilities in the organization's environment[9].

Attack Techniques

This section details four common attack techniques: Brute Force, Remote Code Execution (RCE), Port Scanning, and Directory Traversal. Each attack method exploits different vulnerabilities within a system, and understanding these techniques is crucial for improving cybersecurity defenses.

A **Brute Force attack** is a trial-and-error method used to gain unauthorized access to systems by systematically trying different combinations of passwords or encryption keys until the correct one is found. Attackers attempt all possible combinations of characters, starting

[9] S. Roy, N. Sharmin, J. C. Acosta, C. Kiekintveld, and A. Laszka. *"Survey and Taxonomy of Adversarial Reconnaissance Techniques"*.

from the most commonly used passwords to more complex variations. Variants of brute force attacks include dictionary attacks, which use predefined lists of common passwords, and hybrid attacks, where slight modifications are applied to common passwords by adding numbers or symbols. Although time-consuming, brute force attacks can be highly effective if the targeted system has weak passwords or insufficient security measures in place. The key factor that determines the success of a brute force attack is the complexity of the password and the computational resources available to the attacker.

Remote Code Execution (RCE) is a critical vulnerability that allows attackers to execute arbitrary code on a remote system. This attack takes advantage of software flaws, such as improper validation of user input, to inject and run malicious code with the same privileges as the application or system being exploited. The severity of RCE attacks lies in the fact that they can enable attackers to take full control of a compromised system, install malware or ransomware, steal sensitive data, or alter files. RCE vulnerabilities are frequently found in web applications and services that handle user inputs, where inadequate input sanitization can open doors for code injection. Preventing RCE requires robust security practices, including input validation, patching known vulnerabilities, and implementing secure coding standards[10].

Port Scanning is a technique used by attackers to identify open ports on a target system. By probing these ports, attackers can determine which services are running and identify potential entry points for an attack. Each port is associated with a specific service or protocol, such as HTTP on port 80 or SSH on port 22, and discovering open ports can provide attackers with insights into the vulnerabilities of a system. Various types of port scans exist, such as TCP connect scans, which establish full TCP connections, and SYN scans, which only initiate connections without completing them, making detection more difficult. Although port scanning is not inherently malicious, it is often a precursor to an attack, as it provides valuable information about the target's attack surface. Understanding which services are exposed allows attackers to tailor their approach to exploit known weaknesses in those services[11].

[10] S Biswas, M Sohel, M. Sajal, T Afrin, T Bhuiyan, and M. Hassan. *"A study on remote code execution vulnerability in web applications"*

Directory Traversal, also known as Path Traversal, is an attack technique that enables attackers to access files and directories outside the web server's intended root directory. By manipulating file paths, an attacker can exploit vulnerable systems to navigate through directories and access sensitive files, such as password files, configuration files, or other critical system resources. A typical directory traversal attack might involve the use of relative file paths like `../../`, which instructs the system to move up directories until it reaches the targeted files. If successful, this type of attack can lead to unauthorized data exposure, system compromise, and access to confidential information. To prevent directory traversal attacks, web applications must ensure proper validation of user inputs, restricting file access to specific, safe directories and avoiding the inclusion of user-controlled data in file paths[12].

[11] S. Roy, N. Sharmin, J. C. Acosta, C. Kiekintveld, and A. Laszka. *"Survey and Taxonomy of Adversarial Reconnaissance Techniques"*.

[12] M. Chawda, P. Sharma, and J. Patel. *"Deep Dive into Directory Traversal and File Inclusion Attacks leads to Privilege Escalation"*.

METHODOLOGY

1 The methodology used in this research starts with **the generation of hypotheses** based on observed trends in threat intelligence reports and OSINT. These hypotheses are framed within attack target selection and attack type. By drawing on existing literature and intelligence reports, the study hypothesizes which cloud platforms, services, and regions are more likely to be targeted by attackers. This step sets the foundation for the honeypot design and deployment strategy, ensuring that the experimental setup aligns with real-world attack scenarios.

2 In the next phase, the **honeypots are optimized** to increase their attractiveness to attackers while maintaining a realistic appearance. The honeypots are designed to emulate commonly targeted systems, such as small business network environments or web services with deliberate misconfigurations. Special attention is paid to simulating authentic services such as *SSH*, *HTTP*, and *PostgreSQL* to ensure that the honeypots reflect the types of systems that attackers typically target. Vulnerable banners and outdated software versions are displayed to entice attackers further. The optimization process focuses on ensuring that the honeypots are indistinguishable from legitimate systems, maximizing the likelihood of capturing meaningful data.

3 Once the honeypots are optimized, they are **strategically deployed** across multiple geographic locations and cloud platforms. These diverse deployments allow the research to capture differences in attack patterns based on geographic location and cloud infrastructure.

4 **Attack Target Analysis:** This step analyzes the geographic regions and cloud platforms targeted by attackers. It examines whether specific countries or cloud infrastructures are more frequently targeted and whether these trends align with the hypotheses generated in the first phase. The analysis aims to identify patterns in how attackers select their targets and to determine the influence of geographic or platform-specific factors on attack frequency.

Attack Type Analysis The focus in this step shifts to the types of

5

attacks launched against the honeypots. The analysis categorizes attacks based on the techniques employed, such as port scanning, brute-force attacks, or HTTP request manipulation. It investigates the frequency and distribution of different attack types, providing insights into the tactics and tools attackers favor when targeting different types of services or systems. This step involves parsing and categorizing large datasets to identify the most common and sophisticated attack vectors.

6

Analysis of Correlation Between Attack Target and Type In this section, the correlation between the targeted regions or platforms and the types of attacks used is explored. The analysis examines whether certain attack types are more prevalent in specific regions or cloud environments. It also investigates whether attackers employ different techniques depending on the vulnerability level of the system. The goal is to identify relationships between the nature of the target and the methods used in the attacks, helping to refine the understanding of attacker behavior.

7

Hypothesis Validation and Discussion The final phase involves validating the hypotheses generated at the start of the research. The analysis results are used to confirm or refute the initial hypotheses, providing a data-driven assessment of current attack trends.

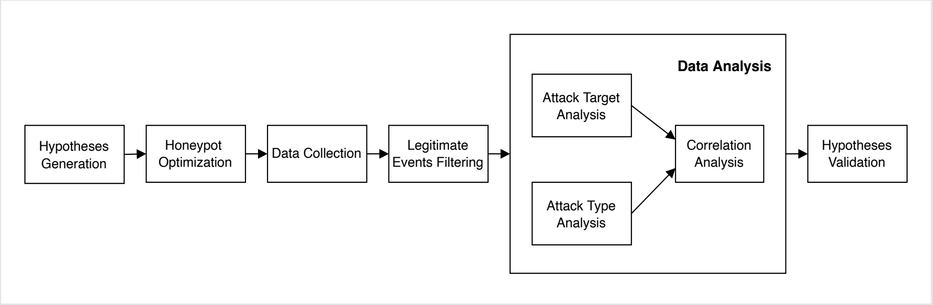


Fig 1. Overview of the methodology

HYPOTHESES

Hypotheses generation is a crucial phase in the threat hunting cycle, where security teams develop ideas about potential cyberattacks based on available information, such as known attack patterns, threat intelligence reports, anomalous network activity, and prior experiences13.

This methodology was applied in this research to identify potential threats targeting the honeypot systems. Since the research environment lacks the typical traits that attract attackers and has no prior system history, formulating highly sophisticated hypotheses poses challenges. However, by focusing on the analysis of observed attack trends and understanding the behavior of attackers interacting with the honeypots, this section aims to develop meaningful hypotheses.

Several critical sources were used to develop hypotheses:

IBM X-Force Threat Intelligence: Selected for its comprehensive insights into global cybersecurity threats, it helped identify emerging trends and patterns.

2024 Data Breach Investigations Report: Provided detailed data on initial access vectors, offering valuable information on how attackers gain entry into systems.

Shodan: This search engine for internet-connected devices was used to discover and analyze exposed systems, revealing critical vulnerabilities attackers could exploit during target selection.

Honeypot Data: Data collected from *L3montree* was analyzed to assess the tactics and techniques used by attackers, helping validate hypotheses related to attack methodologies.

This combined intelligence was instrumental in shaping the hypotheses for understanding how attackers choose targets and initiate attacks, providing a foundation for further analysis.

[13] D. B. Robert M. Lee. *"Generating Hypotheses for Successful Threat Hunting"*

Hypotheses for Honeypot Location

In 2023, Europe emerged as the region most affected by cyberattacks, accounting for 32% of global incidents, surpassing both North America and the Asia-Pacific region[14]. The rapid adoption of cloud services across Europe has significantly expanded the attack surface, making it a prime target for cybercriminals. A key factor driving this increase in attacks is the exploitation of valid cloud credentials, which make up 90% of cloud assets for sale on the dark web[14].

In the United States, the most common initial access vector in 2023 was the use of valid accounts, representing 41% of incidents. Attackers frequently exploited credentials acquired through infostealers and criminal marketplaces, bypassing security controls. Credential harvesting led as the top impact to U.S. organizations (28%), followed closely by data theft and extortion, which each made up 24% of incidents[14].

Furthermore, Brazil has 74,567,265 open ports according to Shodan[15], more than twice the number of open ports in Germany. Given the high number of open ports relative to its geographic size, Brazil's exposure could be an important indicator for attackers, making it crucial to assess how this factor might contribute to the risk of cyberattacks.

Cloud Infrastructure Trends

Major cloud service providers, including AWS, Azure, and Google Cloud, have solidified their dominance in the market, making them attractive targets for cybercriminals. In 2023, the abuse of valid accounts emerged as the most common initial access vector, responsible for 30% of reported incidents[14]. The growing number of compromised cloud credentials underscores the heightened risk facing these large cloud platforms. As the consolidation of cloud services continues, the frequency of attacks is expected to rise, particularly as attackers target the expansive and lucrative infrastructure of these providers.

[14] I. Security. *X-Force Threat Intelligence Index 2024*

[15] Shodan. *Brazil Open Ports on Shodan*

Vulnerable Systems

Software vulnerabilities remain a critical entry point for attackers, particularly in public-facing applications with known but unpatched vulnerabilities. According to the IBM report, attackers frequently exploit such vulnerabilities, especially when organizations fail to maintain up-to-date software versions[14]. This trend underscores the importance of proactive vulnerability management and the regular patching of software to reduce exposure to cyber threats. Systems running outdated software versions are often seen as easy targets, emphasizing the need for organizations to prioritize timely updates and patching.

Hypothesis 1*. The expansion of cloud infrastructure in Europe has led to an increase in cyber attacks targeting cloud account credentials. Honeypot data will reveal a higher frequency of attempts to exploit valid cloud credentials in Europe compared to other regions.*

Hypothesis 2. *SSH and HTTP servers running older versions with a higher number of known vulnerabilities (CVE entries) are targeted more frequently by attackers compared to servers running the latest, more secure versions.*

Hypotheses for Initial Access Vectors

The IBM X-Force Threat Intelligence Index (2024) highlights that valid credentials and phishing were the two leading methods of initial access in 2023, each responsible for 30% of reported incidents[14]. The increasing abuse of valid accounts is largely attributed to the widespread availability of compromised credentials on the dark web, particularly cloud account credentials, which constitute 90% of the cloud assets sold on such platforms[14].

Similarly, the 2024 Data Breach Investigations Report (DBIR) points to a sharp rise in cloud infrastructure breaches, especially in Europe, where cyberattacks have escalated significantly[16].

[16] Verizon. *2024 Data Breach Investigations Report*

Hypothesis 3: *Attackers who successfully obtain credentials from an exposed /.env file are more likely to attempt login attempts on associated services than brute force attacks.*

DATA COLLECTION

This chapter provides an overview of the raw data collected from the honeypots, which serves as the basis for analyzing attacker behaviors, tactics, and methodologies.

This research employed the low-to-medium interaction honeypot named *Oh-my-honeypot*, an open-source tool written in Golang and packaged as an OCI-Container-Image for deployment across various containerized environments. *Oh-my-honeypot* supports multiple emulated services, including SSH and PostgreSQL with a login function, HTTP with an emulated web service, and other services with ports opened for the purpose of capturing preliminary reconnaissance activities. *Oh-my-honeypot* features comprehensive data logging capabilities, enabling the capture and storage of all interactions for subsequent analysis.

Each attack event is logged with key attributes, including a unique attack ID, the source IP address, the port number targeted, the timestamp of the event, the honeypot ID (particularly useful in multi-honeypot environments), and the type of attack.

The event types are classified as follows: *Port Scanning, Login Attempt and HTTP Request*.

The Login Attempt category includes both SSH and PostgreSQL services and records the usernames and passwords used by attackers, providing valuable insights into their attempts to gain unauthorized access.

The SSH service operates on the default port number 22, while the PostgreSQL service operates on its default port number 5432.

In the event type HTTP Request, the honeypot captures a range of details from the HTTP request headers, including user-agent information, the request path, and the method employed (GET, HEAD, POST, PUT, OPTION). This information is crucial in identifying the attacker’s intent and tactics. Researchers can infer the attacker’s browser and operating system and investigate the requested path to determine which specific intent the attacker has.

Furthermore, *Oh-my-honeypot* is capable of opening a multitude of TCP/UDP ports, including those utilized for mail services, file systems, remote access, and data. While these ports do not emulate full services, they are opened to capture and log port scans. Aside from SSH, PostgreSQL, and HTTP, no other service supports detailed interaction. All these events are classified as Port scanning type.

As honeypots spread across the Internet, attackers are increasingly able to detect and evade them[17]. This reality highlighted the need to focus on enhancing the honeypot's authenticity to ensure it functions as a highly convincing decoy system. The objective was to implement a honeypot that closely resembles a legitimate system, capable of deceiving even advanced attackers and encouraging them to launch more sophisticated attacks. By mimicking the behavior and structure of a real-world system, the honeypot increases the likelihood of attracting attackers into deeper interactions, offering valuable insights into their tactics and methods[18].

Optimization

A key feature of the honeypot's realism is a custom-built web-based login interface, designed to resemble a simple NAS system used by small businesses. This interface imitates an internal file-sharing and management platform and supports HTTPS, secured with a self-signed certificate. While the certificate is not issued by a trusted authority, it reflects the basic security measures, providing an authentic experience that attackers expect. This specific setup was chosen due to its frequent targeting by attackers who seek vulnerabilities in small business systems, making it an attractive and realistic target.

The web interface is also configured to capture various web application attacks, such as SQL Injection and Cross-Site Scripting (XSS). When attackers attempt to exploit these vulnerabilities (e.g., by injecting malicious SQL queries into login forms), the honeypot logs the

[17] A. Vetterl and R. Clayton. *"Bitter harvest: systematically fingerprinting low and medium-interaction honeypots at internet scale."*

[18] M. Rabzelj, L. Južni ˇ c, M. Volk, A. Kos, M. Kren, and U. Sedlar. *"Designing and evaluating a flexible and scalable HTTP Honeypot platform: architecture, implementation, and applications"*

malicious payloads, responses, and techniques, offering valuable insights into their attack methods.

To encourage more interactions from attackers, vulnerable paths were intentionally included to simulate common misconfigurations in custom-built systems.

The first vulnerability simulated was the exposure of the .env file, which typically contains sensitive environment variables. This honeypot returned the credential information for the remote access of SSH and PostgreSQL. A typical default username *admin* was set for both services and a 15 character password, including special characters, numbers and alphabets in the password, was generated per http request to *.env*. This unique, randomly generated password made it easy to track an attacker's actions, with any subsequent login attempts providing valuable insights into their behavior and strategies.

Another simulated vulnerability involved a RCE path at /vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php, which returned an empty string when accessed. Unlike other pages that showed a standard "404 page not found" error, this empty response encouraged attackers to attempt malicious actions, allowing for the observation and analysis of their behavior.

By mimicking common real-world misconfigurations, these simulations made the honeypot an attractive target for attackers and yielded valuable data on how they exploit such weaknesses

Banner Configuration

Setting up false fingerprints of services, such as displaying outdated server versions or misleading information, is a common technique to enhance honeypots[17]. The honeypot was implemented to be able to customised with service banners that reflect vulnerable or outdated software versions, making it more attractive to attackers. For example, the SSH banner can be configured to display a vulnerable version such as `SSH-2.0-OpenSSH_5.3p1`, attracting attackers looking to exploit known vulnerabilities. Similarly, the HTTP service can be configured to send custom response headers such as *Server* or `X-Powered-By`, indicating the use of an insecure web server or platform, further encouraging attackers to target the system.

To enhance the honeypot's authenticity, a small, carefully selected set of open ports was chosen to mimic services commonly found in real-world systems, such as file sharing, remote access, and database management. By exposing only these realistic services, the honeypot avoids the suspicion that might arise from having too many open ports, which could reveal its decoy nature. This thoughtful selection helps maintain the appearance of a legitimate target, encouraging attackers to interact more fully with the system[19].

Honeypot Deployment

By deploying honeypots across various regions and cloud service providers, and varying the vulnerability levels of each system, capturing a diverse range of attack patterns and behaviors is available. This approach allows to assess how factors influence the frequency and type of cyberattacks.

Honeypots were deployed to: North America (USA), Europe (Germany), and South America (Brazil)—to capture region-specific attack patterns. These countries were selected based on research during generating hypotheses. This diversity helps assess whether certain regions attract more cyberattack activity, which may be influenced by geopolitical factors or the presence of critical infrastructure.

Honeypots were hosted on multiple cloud service platforms, including Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure. This setup enables a comparison of attack frequency and methods across different providers, helping to identify any platform-specific vulnerabilities or trends in attack behavior.

Vulnerability Level

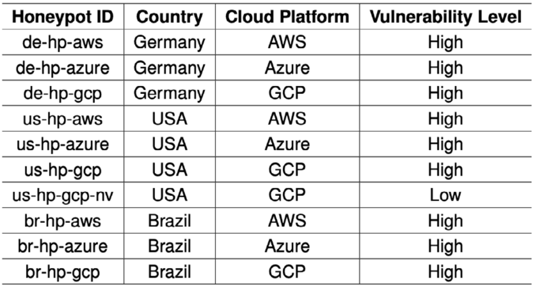
The high-vulnerability honeypot was configured with outdated versions of web server and PHP software, such as `Apache/2.2.22` (Unix) and `PHP/4.1.0`. These versions are known to have multiple vulnerabilities, making them particularly susceptible to attacks like RCE and SQL Injection. The honeypots are intentionally configured to expose

[19] R. N. Dahbul, C Lim, and J Purnama. *"Enhancing honeypot deception capability through network service fingerprinting"*.

specific HTTP headers to mimic real-world misconfigurations. The Server header reveals the web server version (`Apache/2.2.22`), and the `X-Powered-By` header exposes the PHP version (`PHP/4.1.0`), both of which attackers can use to identify known vulnerabilities. Additionally, the Content-Type header is set to `text/html; charset=ISO-8859-1`, and the Access-Control-Allow-Origin header, along with the Referrer-Policy set to `unsafe-url`, potentially exposes further misconfigurations. These settings make the honeypot an attractive target for attackers seeking to exploit common web-based vulnerabilities.

The emulated SSH-service, on the other hand, is configured with an outdated version of OpenSSH, `SSH-2.0-OpenSSH_5.8p2`. This version has known vulnerabilities that can be exploited by attackers using brute-force login attempts or other known SSH exploits. The version string `SSH-2.0-OpenSSH_5.8p2` is presented during the initial SSH handshake, providing attackers with the information they need to identify weaknesses. Misconfigurations in authentication methods, such as weak password-based authentication, make these systems even more attractive to attackers.

A low-vulnerability honeypot was deployed on GCP in the USA, named `us-hp-gcp-nv`, and configured with the latest stable software versions: `Apache/2.5.62` (Ubuntu), `PHP/8.3.0`, and `SSH-2.0-OpenSSH_9.1p10`. These versions have few or no known vulnerabilities in the CVE database, ensuring that this honeypot represents a secure, well-maintained system. This low-vulnerability honeypot serves as a control, enabling direct comparison with vulnerable honeypots deployed under similar conditions, such as in the same country and on the same cloud platform.



Honeypot ID	Country	Cloud Platform	Vulnerability Level
de-hp-aws	Germany	AWS	High
de-hp-azure	Germany	Azure	High
de-hp-gcp	Germany	GCP	High
us-hp-aws	USA	AWS	High
us-hp-azure	USA	Azure	High
us-hp-gcp	USA	GCP	High
us-hp-gcp-nv	USA	GCP	Low
br-hp-aws	Brazil	AWS	High
br-hp-azure	Brazil	Azure	High
br-hp-gcp	Brazil	GCP	High

Fig 2. Honeypot Deployment Overview

By comparing the results, this setup helps assess how much the presence of known vulnerabilities influences attackers′ behavior and whether they prioritize systems with more vulnerabilities. It also helps determine to what extent these vulnerabilities play a key role in attack strategies.

DATA ANALYSIS

This chapter presents the analysis of raw data collected from 10 honeypots over a 30-day period, from August 22, 2024, to September 22, 2024. During this period, a total of 8,457,986 events were recorded, originating from 64,362 unique attacker IP addresses.

Filtering Legitimate Network Traffic

In the modern Internet landscape, large-scale scans are commonly performed by various organizations[20]. Legitimate activities, such as scanning for vulnerabilities and conducting research, generate substantial network traffic, which must be distinguished from actual cyberattacks to ensure the accuracy of research findings. In order to solve this issue, a filtering process was conducted with the objective of identifying and excluding any research or testing activities that might be relevant.

Initially, IP ranges of known vulnerability scanning companies, including Qualys and Rapid7, were identified and filtered from the dataset. Subsequently, IP addresses associated with those activities were filtered by checking HTTP header information and other footprinting indicators. These IP addresses were then cross-referenced against a IP whitelist using a reputation-checking service, AbuseIPDB, to validate their legitimacy. Only those IP addresses confirmed as legitimate were excluded from further analysis. Following the exclusion of legitimate traffic associated with vulnerability scanners, the dataset from the 10 honeypots was reduced to 8,187,215 events and 63,142 unique IP addresses.

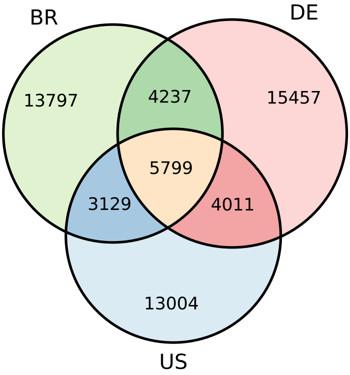
In the analysis for geographic and cloud platform, the low vulnerability system was excluded to maintain accuracy by comparing three honeypots of equal size based on geographic location or cloud service. Additionally, in the analysis based on vulnerability levels, the attack type port scanning was excluded since the vulnerabilities are only exposed by SSH and HTTP services.

[20] J. Mayer, M. Schramm, L. Bechtel, N. Lohmiller, S. Kaniewski, M. Menth, and T. Heer. "*I Know Who You Scanned Last Summer: Mapping the Landscape of Internet-Wide Scanners*"

Country

Over a 30 day period, Honeypots in Germany recorded 2,458,172 events, Honeypots in Brazil recorded 2,744,977 events and the Honeypots in United States recorded 2,304,086 events. To gain a deeper insight into the attacks, the analysis focused on unique IP addresses rather than the volume of events. This approach reduced the potential for distortions from high event frequencies and allowing for a more accurate evaluation of the actual scope of the attacks.
Figure 3 illustrates the distribution of unique attack IP addresses targeting different countries. Germany emerges as the country with the highest number of unique IP addresses (29,504), followed by the United States (26,962) and Brazil (25,943).

These figures encompass IP addresses targeting a single country as well as those involved in attacks across multiple countries. Specifically, of the 29,504 unique IP addresses targeting Germany, 14,047 (47.61%) were also associated with attacks on the United States, Brazil, or both, as indicated by the overlaps of 4,237, 5,799 and 4,011 addresses, respectively. Similarly, 47.99% of the IP addresses targeting the United States were involved in cross-border attacks, while Brazil experienced an even higher degree of overlap, with 50.75% of its attackers also targeting other countries. The data illustrates the global nature of cyberattacks, demonstrating that attackers frequently might disregard national boundaries and adopt a strategic approach to targeting multiple regions simultaneously.



n(DE) = 29504

n(US) = 26962

n(BR) = 25943

n(DE ∪ US ∪ BR) = 59434

Fig 3. Distribution of Unique Attack IP Addresses by Honeypot Location

Figure 4 illustrates the distribution of unique IP Addresses originating from the most aggressive eight source countries—China (CN), the United States (US), India (IN), Tanzania (TZ), Russia (RU), South Korea (KR), Brazil (BR), and Germany (DE)—targeting three countries. China is the largest source of cyberattacks across all three target countries. The United States also contributed a substantial number of attacks, including a considerable volume originating from within its own borders. India ranks third as a source of attacks across all targets.

Cloud Service Provider

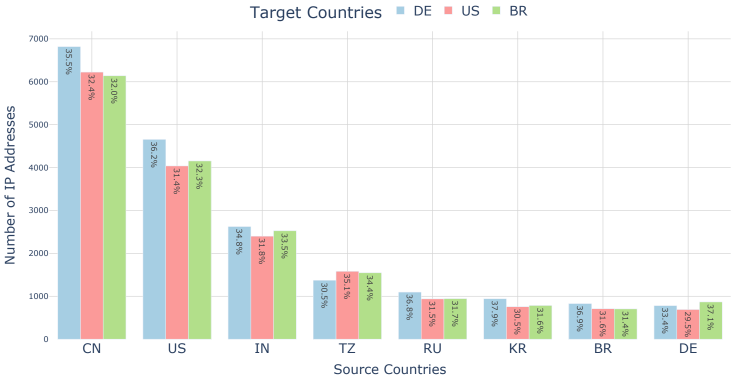


Fig 4. Number of Unique Attack IP Addresses by Top 8 Source Countries

During 30 days, AWS recorded 2,984,066 events, Azure saw 2,536,138, and GCP experienced 2,585,131 events. Similar to the geographic analysis, this section focuses on unique IP addresses rather than the total number of attack events, offering a clearer understanding of attack patterns across cloud environments. Figure 5 shows the distribution of unique IP addresses targeting the three major cloud platforms: AWS, Azure, and GCP.

The data shows that Azure was the most targeted platform, with 28,335 unique IP addresses, followed closely by GCP with 27,526, and AWS with 24,223. Similar to the geographic distribution, many IP addresses were involved in cross-platform attacks, targeting multiple cloud providers simultaneously.

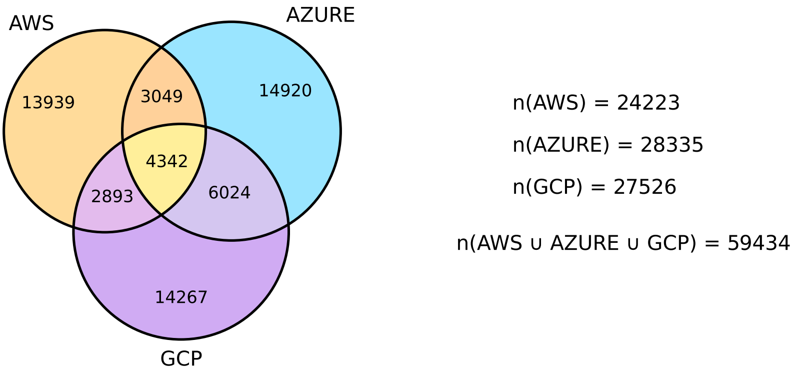


Fig 5. Distribution of Unique IP Addresses by Cloud Services

Vulnerability Level

The comparison of number of attack IP addresses between systems with a known vulnerability and a system without such a vulnerability reveals a clear preference by attackers for the vulnerable system. Figure 6 shows that the vulnerable system receives attacks from 4500 (44.7%) unique ip addresses, while the non vulnerable system was only attacked by around 3300 ip addresses (29.5%). This significant disparity underscores the tendency of attackers to focus on weaker systems, which are perceived as easier targets. Furthermore, 12.9% of attacks targeted both high and low vulnerability systems.

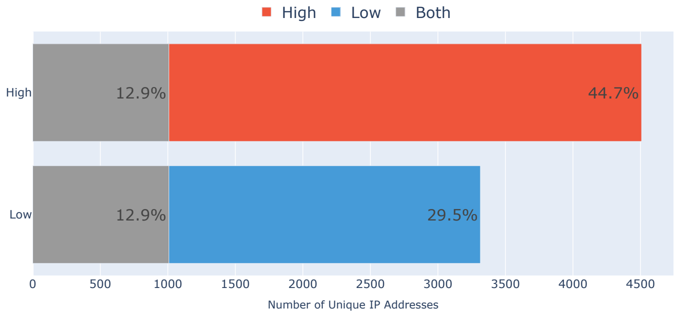


Fig 6. Number of Unique Attack IP Addresses by Vulnerability Level

Attack Type Analysis

To understand attack patterns and intent, the event type are categorized based on specific attack techniques. This categorization helps to identify how different techniques are used throughout the stages of an attack, revealing the attackers‘ strategies and objectives. By breaking down event types into distinct attack types, a clearer picture of the attack sequence and its goals emerges. Each of these attacks is further analyzed and modeled according to Tactics, Techniques, and Procedures (TTPs) and the cyber kill chain model. This structured approach enables a deeper understanding of how attackers move from initial reconnaissance, through weaponization and delivery, to exploitation.

Port Scanning

Port scanning, where attackers probe systems for open ports and services, emerged as the most frequent event type. A total of 6,599,508 port scanning instances were recorded, originating from 59,148 unique IP addresses. The scanned port numbers were analyzed based on the services they represent, revealing patterns in the systems most frequently targeted by attackers.

The analysis identified several key service groups. File sharing and network services were the most targeted, with Port 445 (SMB) leading with over 4.7 million scans. Port 21 (FTP) was also often scanned,

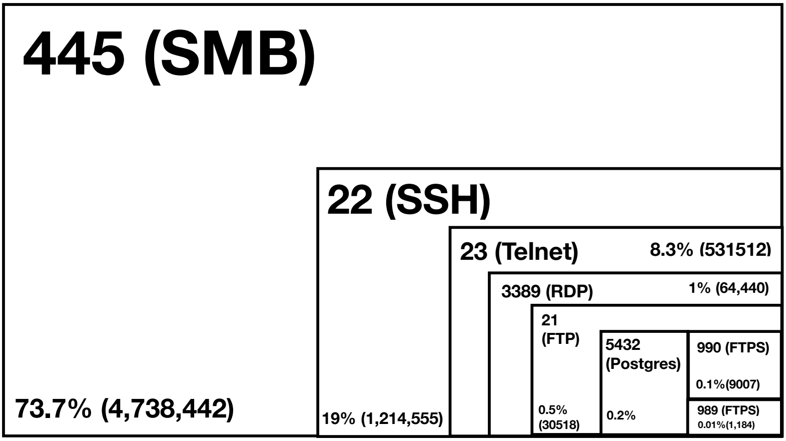


Fig 7. Port scanning distribution

recording 30,518 scans. Despite being an older protocol with known security issues, its lack of encryption makes it vulnerable to data interception and exploitation. Similarly, FTPS, which runs on Ports 990 and 989, saw 9,007 and 1,184 scans, respectively, as attackers continue probing for misconfigurations despite the added security of SSL/TLS encryption.

Remote access services were another highly targeted group. Port 22 (SSH) saw over 1.2 million scans, reflecting its frequent use in brute-force login attempts. Telnet (Port 23), despite being outdated but insecure, still attracted attention, particularly on legacy systems where encryption is absent. Port 3389 (RDP), used for remote access to Windows machines, has become a major target in recent years, especially with the rise of remote work. Attackers focus on this port to exploit vulnerabilities or weak credentials, aiming to gain unauthorized access to systems. Database services, particularly Port 5432 (PostgreSQL), were also targeted. With over 13,000 scans recorded, attackers frequently aim to exploit vulnerabilities in database systems to gain access or exfiltrate sensitive data from widely-used platforms like PostgreSQL.

HTTP Request

HTTP request events offer a wider variety of attack vectors compared to other types of events, giving more detailed insights into attacker behavior. By analyzing both the HTTP headers and body, it becomes easier to identify the attacker's intent.

Among the values in an HTTP request header, the path is particularly important because it shows the specific route the attacker is trying to access. This helps categorize their actions and intentions when interacting with the honeypot.

The most common type of attempted paths, making up 60%, involved accessing server configuration files or retrieving server information. The analysis revealed that `.env` file was the primary target. This file often contain sensitive information like API keys, database credentials, and environment variables, making them highly valuable to attackers. In some cases, `POST` requests carried payloads with Indicators of Compromise (IOCs), such as "Graber," "legion," and "androxgh0st," signaling malware activity[21]. Attackers also attempted to access files containing authentication credentials or repository configurations, like SSH keys(`/.ssh`) and Git configuration files(`.git/config`). There were attempts to retrieve specific cloud service credentials like `/.aws/credentials`, `/gcp_credentials.json` and `/azure_credentials.json`, which, if accessed, could give attackers control over systems or cloud resources.

The second most common attempts targeted default pages, while the third focused on paths known to be vulnerable, likely to check for the possibility of RCE (Remote Code Execution). In addition to configuration files, attackers sought server-specific information by targeting paths that exposed server settings, such as detailed PHP environment configurations. Common pages like login and diagnostic pages such as `/login.rsp` were also frequently targeted.

The analysis also uncovered numerous attempts at RCE, particularly through GET requests. These requests were likely used by attackers to probe for vulnerabilities in the server's code execution capabilities. By sending malicious scripts or commands within the GET request, attackers aimed to determine whether the server would execute the code, potentially allowing them to take control of the system or escalate privileges. RCE attempts often focus on exploiting known vulnerabilities in web applications, plugins, or misconfigured servers.

For example, approximately 17.2% of RCE probes specifically targeted the path `cgi-bin/luci/;stok=/locale`, which is linked to CVE-2023-1389. This vulnerability affects routers running OpenWRT firmware, where attackers can exploit the path to bypass authentication and execute commands remotely, allowing them to take control of the device or perform malicious actions.

Many cases also involved PHP code injection, where attackers used base64-encoded payloads to conceal malicious activity. This technique helps bypass security measures and input validation, as the harmful code appears as harmless data.

Another frequent attack targeted misconfigurations in PHP-CGI imple-

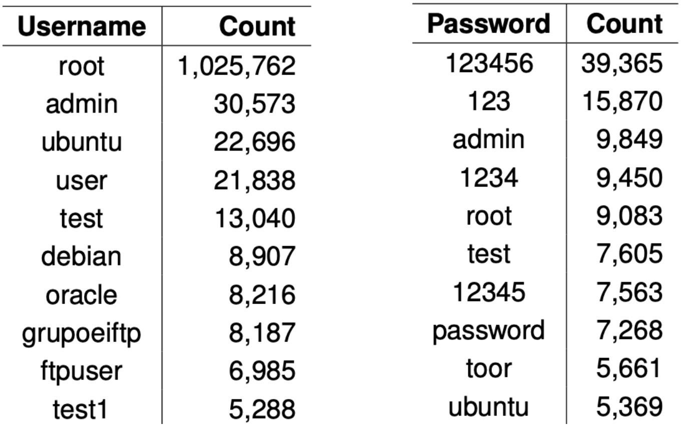
[21] F. B. of Investigation (FBI), Cybersecurity, and I. S. A. (CISA). *Known Indicators of Compromise Associated with Androxgh0st Malware*.

mentations. Attackers attempted to execute PHP commands to retrieve system information or run encoded scripts for malicious purposes. This method exploits weaknesses in the PHP environment, allowing unauthorized access or sensitive data retrieval.

An additional type of attack observed was XML External Entity (XXE) injection, which exploits vulnerabilities in XML parsers. Attackers craft malicious XML payloads to force the server to load external files or access sensitive data. XXE attacks can be used to extract confidential information, execute code, or even cause Denial of Service (DoS) by making the server process large or harmful files. Captured XXE payloads often manipulated the server's XML parsing to retrieve sensitive data or compromise system resources.

Login-Attempts

Login attempts were another significant event type, recorded 1,393,451 times from 10,340 unique IP addresses. These ranged from harmless login failures to malicious brute-force attacks, where attackers repeatedly tried common usernames and passwords. Additionally, some attackers used fake credentials obtained from `/.env`.



Username	Count
root	1,025,762
admin	30,573
ubuntu	22,696
user	21,838
test	13,040
debian	8,907
oracle	8,216
grupoeiftp	8,187
ftpuser	6,985
test1	5,288

Password	Count
123456	39,365
123	15,870
admin	9,849
1234	9,450
root	9,083
test	7,605
12345	7,563
password	7,268
toor	5,661
ubuntu	5,369

Fig 8. Most used usernames and passwords

The most targeted username was “root,” with over a million attempts. As the default superuser account on many Unix and Linux-based systems, “root” is a prime target for attackers seeking full control. A successful login with the root account grants administrative privileges, which explains why both the “root” username and password
are frequently targeted in brute force attacks.

Following “root,” there is a notable drop in attempts for the next most common username, “admin.” This username is widely used across various systems, particularly in web-based interfaces, routers, and content management systems, making it another popular target. Similarly, “admin” was the third most common password, suggesting that attackers are targeting environments where the “admin” account may use “admin” as the password, a common yet insecure setup found in default configurations.

Usernames like “ubuntu” and “debian,” which appear in both the username and password lists, indicate that attackers are specifically focusing on Linux distributions that may still use default settings. These attempts suggest an effort to exploit systems where default usernames and passwords haven’t been changed, particularly in systems running these popular distributions.

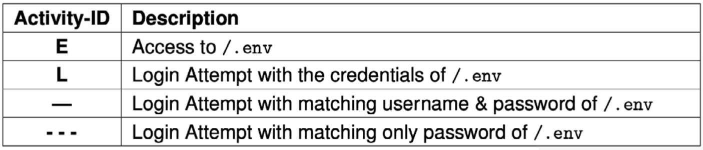
Interestingly, despite the honeypot only supporting login attempts via SSH and Postgres, usernames like “grupoeiftp” and “ftpuser” were still attempted. This suggests attackers are probing for FTP servers, possibly hoping to find misconfigured or vulnerable services.

In terms of passwords, “123456” topped the list with nearly 40,000 attempts. This weak and commonly used password, along with others like “123,” “1234,” and “12345,” reflects a focus on simple numeric combinations, particularly in environments with weak password complexity requirements. The password “password,” while generic, remains commonly used, especially in test environments.

Overall, this analysis reflects attackers’ focus on default configurations and weak security practices, making these systems easy targets for brute force attacks.

Valid Credentials

From an attacker's perspective, obtaining valid credentials is a more efficient way to save time and resources compared to using brute force attacks. To observe how attackers attempt to gain valid credentials, honeypots were set up to generate new passwords when HTTP requests were made to the `.env` path. Over the course of a month, 6,583 passwords by 344 unique IP addresses were generated, but only 13 passwords were attempted by five indicators in ssh login. Three typical patterns were identified in this login attempts.



Activity-ID	Description
E	Access to `/.env`
L	Login Attempt with the credentials of `/.env`
—	Login Attempt with matching username & password of `/.env`
- - -	Login Attempt with matching only password of `/.env`

Fig 9. Table of malicious actions involved in login attempts

The first pattern (Figure 10) involved two separate incidents where one IP address accessed the `.env` path to generate a password, followed by another IP from the same subnet attempting to log in using the retrieved credentials.

In the first incident with the pattern 1, An IP address from Brazil (2.57.171.58) accessed the `.env` path to obtain a newly generated password. Nine seconds later, a second IP (2.57.171.57) from the same subnet tried to log in using that password. However, even though the username was included in the .env data, the attacker entered the username as a hyphen(-) and used only the password.

A similar incident with the pattern 1 occurred in France on the same day. An IP address sent an HTTP request to the `.env` path. Just 25 seconds later, another IP from the same subnet attempted to log in via SSH using the password generated by the first IP. As in the first case, the attacker entered a hyphen (-) as the username. After

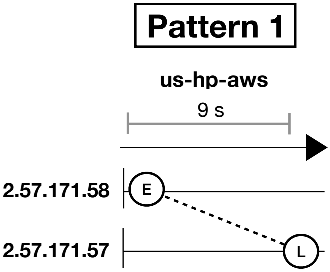


Fig 10. Timeline of the credential retrieval

the login attempt failed, no further activity was detected in both cases.

The second pattern(Figure 11) involved a single IP address accessing the `.env` de-hp-azure br-hp-azure us-hp-aws path to obtain credentials, followed by an SSH login attempt using those credentials.
This pattern was also observed twice.

In the first case with the pattern 2, an IP address from Tunsia (197.244.84.199) initiated the attack. Over the span of 1 hour and 6 minutes, the attacker sent individual `HTTP GET` requests to the `.env` path of the honeypots de-hp-azure, us-hp-gcp and br-hp-azure. The HTTP request headers indicated that the attacker used `python-requests/2.32.3`, showing that they probably automated the process of accessing the `.env` path using Python. Later, the attacker revisited the de-hp-azure system. This time, the user agent header contained the typical browser and operating system information, indicating that it was likely to have been accessed via the browser rather than Python script. After scanning port 21, the attacker attempted
an SSH login on port 22 using the credentials generated by the honeypot. After the login failed, they accessed `/phpmyadmin` and `/index.php` via the browser and then switched the username to `root`, trying the same password again.

About 57 minutes later, the attacker used Python again to access the `.env` path on the us-hp-azure system but didn't try to login.

In the second case with the pattern 2, a single IP address from Canada repeatedly performed path traversal and login attempts whenever it accessed the `/.env` file. The attacker made multiple HTTP requests to various paths and even uploaded a PHP script returning a string

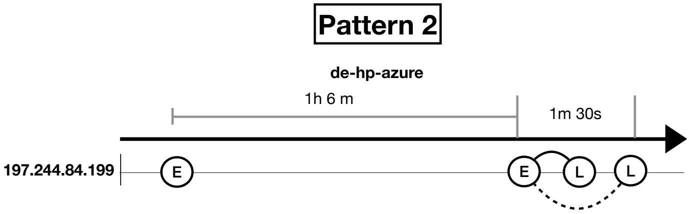


Fig 11. Login attempts from the same ip address

to `vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php`. Shortly after accessing the `/.env` file, the attacker attempted an SSH login on port 22, with an average delay of just four seconds, indicating that these actions were probably automated by a script.

Even though the login attempts failed, the attacker continued scanning the website for vulnerabilities. This attack spanned two days, and the attacker targeted a total of seven honeypots, repeating this pattern throughout.

In the third pattern, only one case was captured by the honeypots. In this case, three of the four IP addresses from the Netherlands (45.86.200.12, 45.86.200.16, 185.82.72.1) conducted path traversal to gather sensitive information by accessing specific files and paths. Once these IPs obtained credentials, a single main IP address (45.86.200.11) attempted to log in using the collected data via SSH.

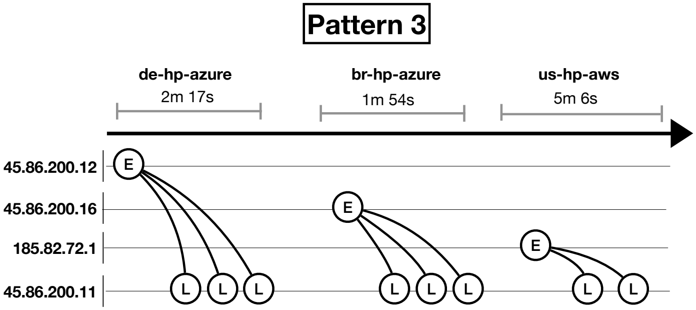


Fig 12. Login attempts using various different ips

This coordinated approach, where different IPs are used for reconnaissance and another for login attempts, shows a clear strategy to divide the attack stages. By separating the information-gathering phase from the login attempts, the attacker likely aimed to reduce the chance of detection. This method makes it harder for defenders to track the entire attack path, as reconnaissance and login attempts appear to come from different sources, complicating defense measures.

The observed patterns highlight the growing sophistication of attackers in credential harvesting and login attempts. Automating these actions through scripts allows attackers to increase their speed and efficiency. The use of multiple IP addresses, often from the same subnet and the coordination of reconnaissance and login phases across different IPs, as seen in the third pattern, strongly suggests the involvement of botnets at one service provider.

Threat Modeling

The diagram 13. illustrates the various stages of a cyberattack, highlighting key techniques used at each stage within kill chain model. In the recon stage, two notable attack types were observed: Port Scanning and Directory Traversal. Both of these techniques fall under Directory Traversal (T1595).

Data analysis revealed that after port scanning, attackers often attempted to proceed with further actions, such as launching a web attack through a second step involving Directory Traversal. However, many instances were also observed where attackers bypassed the recon phase entirely, directly moving on to the delivery stage.

Specifically, brute force attacks (T1110) targeting SSH or Postgresql services were attempted without any preceding port scanning. Similarly, HTTP requests also skipped the Directory Traversal phase and proceeded straight to RCE or XML External Entity (XXE) attacks (T1059). However, for valid credentials (T1078), the attackers had to access the `/.env` page during the Directory Traversal to acquire the credential.

In the delivery stage, brute force attacks typically involve preparing a dataset of usernames and passwords. On the other hand, in cases of XXE and RCE attacks, attackers might be preparing malicious scripts to execute.

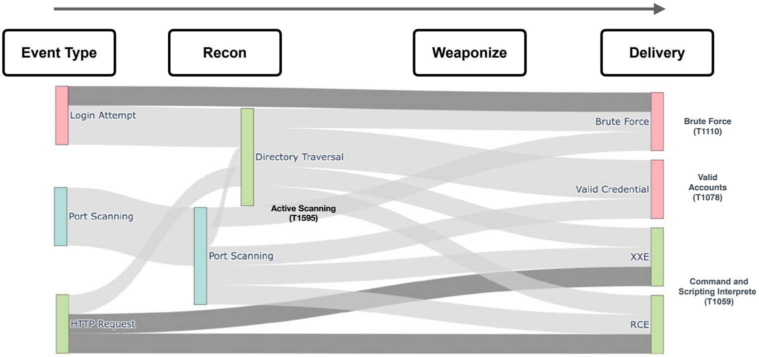


Fig 13. Threat modeling with captured attack types

DISCUSSION

This section presents the validation of three generated hypotheses related to cyberattacks on cloud environments, vulnerability exploitation, and credential theft. Each hypothesis is tested and analyzed in detail, providing insights into attacker behavior and preferences, particularly in relation to cloud infrastructure, system vulnerabilities, and brute-force login attempts. The results offer valuable information on how specific factors, such as geographic location and system configuration, influence the likelihood of an attack.

Attacks Targeting Cloud Environments

Hypothesis 1. *The expansion of cloud infrastructure in Europe has led to an increase in cyberattacks targeting cloud account credentials. Honeypot data will reveal a higher frequency of attempts to exploit valid cloud credentials in Europe compared to other regions.*

The data collected from the honeypots showed differences in attack unique indicator across geographic regions. Germany emerged as the primary target among the three countries.

The observed number of attack unique IP addresses solely targeted for Germany was 15,457, for Brazil 13,797, and for the United States 13,004, giving a total of 42,258 unique IP addresses. Germany's prominence as the most attacked region may be due to the high adoption of cloud services across Europe. This supports the validity of Hypothesis 1.

The analysis of HTTP request events also highlights a higher volume of attacks targeting Germany compared to other countries. Malware like AndroxGh0st, which exploits vulnerabilities in cloud services such as AWS to steal valid credentials, was frequently detected in Germany. With its extensive cloud infrastructure, Germany appears to have been disproportionately affected by these types of attacks compared to the United States and Brazil.

Vulnerability and Attack Preferences

Hypothesis 2*. SSH and HTTP servers running older versions with a higher number of known vulnerabilities (CVE entries) are targeted more frequently by attackers compared to servers running the latest, more secure versions.*

The data analysis supports the hypothesis that systems with more known vulnera-
bilities are targeted more frequently than those with fewer vulnerabilities. Attackers appear to prioritize older, high-vulnerability systems as easier targets. Specifically, 44.7% of the attacks (4,508 events) were directed at high-vulnerability systems, while only 29.5% (3,315 events) targeted low-vulnerability systems, out of a total of 7,823 recorded events.

This strong preference suggests that attackers actively seek out systems running outdated software with more known CVEs, as they present more exploitable opportunities. Further analysis revealed that while HTTP request activity was comparable across both high- and low-vulnerability systems, brute-force login attempts showed a marked disparity. High-vulnerability systems experienced significantly more login attempts, indicating that attackers are more aggressive in exploiting these systems. This behavior further supports the hypothesis, highlighting the attackers' preference for systems they perceive as easier to compromise.

Brute Force and Credential Theft

Hypothesis 3*. Attackers who successfully obtain credentials from an exposed /.env file are more likely to attempt login attempts on associated services than brute force attacks.*

During the one-month observation period, honeypots were configured to generate new passwords whenever an HTTP request was made to the `/.env` path. In total, 6,583 passwords were generated from 344 unique IP addresses. However, despite the large number of generated passwords, only 13 of these passwords were used in login attempts,

and those attempts were made by just 5 unique IP addresses.
This means that less than 1.5% of the attackers who accessed the `/.env` file and generated credentials actually attempted to use those credentials for SSH logins.

These findings suggest that the majority of attackers did not follow up with login attempts using the credentials they obtained. However, it is important to note that while only 5 IP addresses used the credentials from the `/.env` file for login attempts, the remaining 339 IP addresses were not entirely inactive. All 339 IP addresses made at least one login attempt, and 169 of these made 10 or more login attempts. Furthermore, there were 5 IP addresses that made even over 10,000 brute-force login attempts.

Based on this data, the hypothesis that attackers who successfully obtain credentials from an exposed /.env file are more likely to attempt login attempts on associated services rather than resorting to brute force attacks is **not** strongly supported.

While some attackers did use the credentials for login attempts, the number of attempts was minimal compared to the total number of generated passwords. Additionally, the fact that a significant portion of the attackers resorted to brute force attempts, with many making repeated login efforts, suggests that brute force remains a preferred method for many attackers, even when credentials are potentially available.

Limitation

Despite the valuable insights gained through this research, several limitations must be acknowledged. These limitations affect the scope and accuracy of the findings, particularly in relation to the time frame of data collection, the inherent limitations of honeypot technology, and the challenges in distinguishing legitimate activities from malicious attacks.

The data collection for this research was limited to only one month. As a result, it is difficult to definitively conclude the cyberattack trend for a specific target and attack type. Furthermore, observing attack patterns associated with Advanced Persistent Threats (APT), which often observe the target over extended period, is challenging with a short research. A longer monitoring period is necessary to capture the cyberattacks and make more statistically significant assumptions about attack patterns or target selection.

Although efforts were made to optimize the honeypots to avoid detection by attackers, there is still a possibility that more advanced attackers could recognize the honeypot as a decoy. This limitation means that the full range of attack behaviors may not have been captured, as attackers may have chosen to disengage upon detecting the honeypot, reducing the realism of the interactions.

Before data analysis, filtering out the legitimate events was conducted. The clear identifiers in HTTP Header and the published IP range on internet made easier to recognize the legitimate security testing or research. However, Port Scanning and Login Attempt events were not even possible to filter out and there are still possibility that some legitimate events were incorrectly labeled as attacks, or conversely, some attacks may have been filtered out, potentially affecting the accuracy of the analysis. This challenge underlines one of the limitations of this research, as not all scanning or probing activities can be accurately classified without detailed knowledge of the intent or source.

CONCLUSION

This research provides valuable insights into cyber attacker behavior by utilizing honeypots to capture and analyze various attack patterns. By deploying honeypots across multiple cloud platforms and geographic regions, this study was able to observe how attackers target specific systems and services, adjusting their techniques based on the vulnerabilities and environmental factors present.

The data analysis confirmed several hypotheses, such as the tendency for attackers to prioritize high-vulnerability systems. This was evident in the significant number of attacks directed at older SSH and HTTP servers with known vulnerabilities. Additionally, the study highlighted the continued prevalence of brute-force attacks,particularly when targeting exposed SSH services.

These findings emphasize the importance of maintaining up-to-date security configurations and patching known vulnerabilities to mitigate the risk of exploitation.

Overall, the research demonstrates that by understanding attack patterns and correlating them with specific system vulnerabilities, organizations can better prepare defensive strategies. Honeypots proved to be an effective tool for gathering realworld threat intelligence and enhancing the understanding of attacker tactics, techniques, and procedures (TTPs).

Future research could focus on refining honeypot designs to attract more sophisticated attacks and exploring the potential of integrating honeypot data with other cybersecurity systems to improve detection and response capabilities.

About L3montree

SECURING YOUR SOFTWARE & CLOUD NATIVE ENVIRONMENT

We consult and support you in the efficient development and modern operation of secure software with DevSecOps and cloud technologies. We place particular emphasis on open source and technical sovereignty.

L3montree
Cybersecurity

This research analyzes the early stages of cyberattacks by deploying honeypots across cloud platforms and regions to identify attack patterns and techniques, aiming to generate actionable threat intelligence and enhance proactive cybersecurity strategies.

L3MONTREE.COM