SSH Honeypot-Based Intrusion Detection And Attack Analysis

E. Senthil Raja[1], V Kaaviyan[2], U Paarthiv[3], P Ramu[4]
[1,2,3,4]*Department of Computer Science and Engineering, Paavai Engineering College, Namakkal, Tamil Nadu, India*

***Abstract*—Cyber-security threats continue to evolve, necessitating proactive measures to safeguard systems and networks. One such measure is the implementation of SSH (Secure Shell) honeypots, designed to detect and deflect malicious activities targeting SSH services. This paper presents the design and deployment of an SSH honeypot infrastructure that leverages emulation and deception techniques to attract attackers. By analyzing the tactics, techniques, and procedures (TTPs) used by adversaries, the system provides valuable insights to strengthen defense mechanisms. The results demonstrate the effectiveness of honeypots in enhancing cyber-security posture and supporting threat intelligence research.**

***Index Terms*—Cyber Security, Honeypot, SSH, Intrusion Detection, Network Security, Threat Analysis.**

I. INTRODUCTION

Cyber Security, Honeypot, SSH, Intrusion Detection, Network Security, Threat AnalysisTraditional ASD diagnosis relies primarily on behavioral observation, developmental assessments, and interviews conducted by trained specialists. Diagnostic tools such as the Autism Diagnostic Observation Schedule(ADOS), Autism Diagnostic interview-Revised(ADI-R), and Modified Checklist for Autism in Toddlers(M-CHAT) are widely used by clinicians. While these methods are effective, they require extensive time, trained professionals, and repeated clinical sessions. As a result, early diagnosis may be delayed, particularly in rural and low-resource areas where specialists are not easily accessible.

Traditional security mechanisms such as firewalls and intrusion detection systems (IDS) focus mainly on preventing unauthorized access. However, they often fail to provide detailed insights into attacker behavior and evolving attack techniques. To address this limitation, the concept of honeypots has emerged as an effective cybersecurity strategy. A honeypot is a decoy system designed to mimic real services and intentionally attract attackers, allowing security researchers to observe, analyze, and understand malicious activities in a controlled environment.

This project focuses on the design and implementation of an SSH honeypot system that simulates a vulnerable SSH service to lure attackers. The system captures login attempts, records attacker commands, and logs interaction data for further analysis. By studying the tactics, techniques, and procedures (TTPs) used by attackers, valuable intelligence can be obtained to strengthen security defenses and improve threat detection capabilities.

The primary objective of this work is to develop a cost-effective and efficient honeypot solution that enhances cybersecurity awareness and provides practical insights into real-world attack patterns. The findings from this project can assist organizations in improving their security posture and preparing for emerging cyber threats.

II. RELATED WORK

Honeypots have been widely studied as an effective cyber-security mechanism for detecting and analyzing malicious activities. Several research works have focused on Secure Shell (SSH) honeypots, particularly using tools like Cowrie, to understand attacker behavior and improve intrusion detection systems.

Recent studies highlight that honeypots act as deception-based security systems that lure attackers and capture their interactions for analysis. A modern adaptive approach called Q-Cowrie integrates reinforcement learning with traditional honeypot frameworks to dynamically respond to attacker

behavior. This approach enhances attacker engagement and improves intelligence gathering compared to static honeypots.

Another study explored the use of machine learning techniques such as clustering algorithms (e.g., K-Means) to analyze SSH honeypot data. By deploying Cowrie in a controlled environment, researchers were able to capture attacker commands, IP addresses, and session details, enabling the identification of attack patterns and anomalies.

In addition, classification-based approaches have been proposed to detect brute-force SSH attacks. For instance, the Naïve Bayes algorithm has been applied to honeypot-generated logs to classify malicious login attempts and improve early detection accuracy. These methods demonstrate how integrating data analytics with honeypots can enhance threat detection capabilities.

Earlier research on hybrid honeynet systems emphasized the importance of interaction levels in honeypots. Medium-interaction honeypots are sufficient for detecting automated attacks, while high-interaction honeypots provide deeper insights into sophisticated attacker behavior by allowing full system interaction.

Furthermore, recent implementations have focused on integrating honeypots with other security tools such as Intrusion Prevention Systems (IPS). For example, combining Cowrie with Suricata enables real-time detection and automatic blocking of malicious activities, making the system more proactive and effective in real-world environments.

Overall, existing research demonstrates that SSH honeypots are valuable tools for collecting threat intelligence, analyzing attacker strategies, and improving cyber defense systems. However, challenges such as scalability, adaptability, and real-time response remain open areas for further research, which this project aims to address.

III. PROBLEM STATEMENT

With the increasing reliance on remote access protocols, Secure Shell (SSH) has become a critical component in system administration and network management. However, its widespread usage has also made it a prime target for cyber attackers, particularly for brute-force attacks, credential theft, and unauthorized access attempts.

Traditional security mechanisms such as firewalls and intrusion detection systems (IDS) primarily focus on preventing attacks but often lack the capability to analyze attacker behavior in depth. These systems generate alerts but provide limited insights into the tactics, techniques, and procedures (TTPs) used by adversaries. As a result, organizations struggle to proactively adapt their defense strategies against evolving cyber threats.

Furthermore, there is a lack of cost-effective and practical solutions that can simulate real-world attack scenarios and collect detailed attack data without risking actual system assets. Existing systems may also fail to engage attackers long enough to gather meaningful intelligence, especially in the case of sophisticated or automated attacks.

Therefore, there is a need for an efficient and scalable solution that can:

- Attract and engage attackers in a controlled environment
- Capture and log detailed attack activities
- Analyze attacker behavior and identify common attack patterns
- Provide actionable insights to strengthen cyber-security defenses

This project addresses these challenges by developing an SSH honeypot system that acts as a decoy server, enabling the monitoring, recording, and analysis of malicious activities targeting SSH services.

IV. PROPOSED SYSTEM

The proposed system is an SSH-based honeypot framework designed to detect, monitor, and analyze malicious activities targeting SSH services. It operates as a decoy server that mimics a real system environment, thereby attracting attackers and capturing their interactions without compromising actual system resources.

The system is built using a medium-interaction honeypot, which provides a realistic shell environment while maintaining control over system-level access. This approach ensures a balance between security and the ability to collect meaningful attacker data.

4.1. System Components
The proposed system consists of the following major components:

Honeypot Engine:
Simulates an SSH service and provides a fake login interface. It allows attackers to attempt authentication and execute commands in a controlled environment.

Data Logging Module:
Records all attacker activities, including:
- Login attempts (username and password)
- IP addresses and geolocation
- Commands executed during sessions
- Session duration and interaction patterns

Database Storage:
Stores captured data securely for further analysis. Structured storage enables efficient querying and reporting.

Analysis Engine:
Processes the collected data to identify:
- Brute-force attack patterns
- Frequently used credentials
- Common malicious commands

Visualization Dashboard:
Displays real-time statistics and graphical reports, helping administrators understand attack trends and system activity.
The overall workflow of proposed system consists of the following stages:

4.2. Working Principle
The working of the proposed system can be summarized as follows:
1. The honeypot is deployed on a server with an open SSH port.
2. Attackers attempt to connect and perform login operations.
3. The system accepts or simulates authentication to engage the attacker.
4. All interactions, including commands and session details, are recorded.
5. The collected data is analyzed to extract useful insights about attacker behavior.
6. Results are displayed through a monitoring dashboard for further investigation.

4.3. Key Features
The key features of the project are:
- Real-time monitoring of SSH attack attempts
- Capture of detailed attacker interaction data
- Safe and isolated environment to prevent system compromise
- Support for data analysis and visualization
- Scalable and cost-effective deployment

4.4. Advantages of the Proposed System
The Advantages of the proposed system is:
- Provides deep insights into attacker behavior
- Helps in identifying emerging cyber threats
- Enhances intrusion detection capabilities
- Supports proactive security measures
- Easy to deploy and maintain

4.5. Novelty of the System
Unlike traditional security tools, the proposed system focuses on deception-based defense, where attackers are intentionally misled into interacting with a simulated environment. This enables the collection of real-world attack data without exposing critical infrastructure, making it a valuable tool for cyber-security research and practical defense strategies.

V. METHODOLOGY

The methodology of the proposed SSH honeypot system involves a systematic approach to deployment, data collection, monitoring, and analysis of cyber-attacks. The goal is to simulate a realistic SSH environment that attracts attackers and captures their activities for further study.

1. System Setup and Deployment:
The first step involves setting up the honeypot environment on a secure server or virtual machine. A medium-interaction SSH honeypot (such as Cowrie) is installed and configured to emulate a real SSH service.
- Configure open SSH port (port 22 or alternative port)
- Create a fake file system and user environment
- Set up logging mechanisms for capturing activities

This setup ensures that the honeypot appears as a legitimate target to potential attackers.

2. Attack Simulation and Data Capture:
Once deployed, the honeypot passively waits for incoming connections from attackers. When an attacker attempts to access the system:
- Login attempts are recorded (usernames and passwords)
- Source IP addresses and timestamps are logged
- Commands executed by the attacker are captured
- Session activities are monitored in real time

3. Data Storage:
All captured data is stored in a structured format using databases or log files. This includes:
- Authentication logs
- Command history
- Session details
- Network activity

Efficient storage ensures easy retrieval and processing of large volumes of attack data.

4. Data Analysis:
The collected data is analyzed to identify patterns and extract meaningful insights. This includes:
- Detection of brute-force attack attempts
- Identification of commonly used credentials
- Analysis of frequently executed malicious commands
- Classification of attacker behavior

Basic statistical analysis or machine learning techniques can be applied to improve accuracy and detection capabilities.

5. Visualization and Monitoring:
A visualization dashboard is used to present the analyzed data in an understandable format. Tools such as graphs and charts help in:
- Monitoring real-time attack activity
- Observing trends and patterns
- Generating reports for further study

This step enhances decision-making and helps administrators respond effectively to threats.

6. Security and Isolation:
To ensure safety, the honeypot is isolated from the actual system network. This prevents attackers from accessing real resources while still allowing interaction within the simulated environment.
- Use sandboxing or virtual machines
- Restrict outbound connections
- Monitor system integrity

7. Workflow Summary:
- Deploy SSH honeypot
- Attract and capture attacker activity
- Store logs and session data
- Analyze attack patterns
- Visualize results and generate reports

VI. MODEL DESCRIPTION

The proposed SSH honeypot system is designed using a modular architecture where each module performs a specific function to ensure efficient operation and analysis of cyber-attacks. The honeypot deployment module is responsible for setting up and configuring the SSH honeypot environment. It simulates a real SSH server by creating a fake system interface, user credentials, and file structure, making it appear as a legitimate target for attackers.

The authentication and interaction module handles incoming login attempts and manages attacker interactions within the honeypot. It captures usernames and passwords entered by attackers and provides a controlled shell environment to simulate real system access. This approach helps in engaging attackers for a longer duration and allows the system to record detailed behavior.

The data logging module plays a crucial role by continuously recording all attacker activities. It logs login attempts, IP addresses, timestamps, and commands executed during sessions. This ensures that every action performed by the attacker is captured for further analysis.

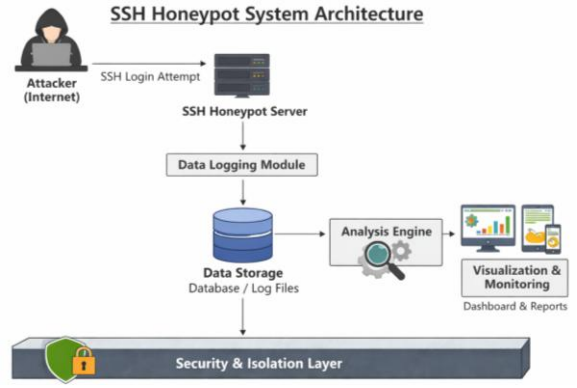
The data storage module is responsible for organizing and storing the collected data in a structured format such as databases or log files. This structured storage allows easy retrieval, efficient management, and smooth processing of large volumes of attack data.

The analysis module processes the stored data to identify meaningful patterns and insights. It detects brute-force attacks, identifies commonly used credentials, and analyzes frequently executed malicious commands. This helps in understanding attacker behavior and improving cyber-security strategies.

Finally, the visualization and security module present the analyzed data in a user-friendly format using graphs and dashboards, enabling real-time monitoring of attacks. At the same time, it ensures that the honeypot system is isolated from the actual network using virtual machines or containers, preventing any potential damage to real systems. Together, these modules form a complete and secure SSH honeypot system for effective cyber-attack detection and analysis.

VII. SYSTEM ARCHITECTURE

The system architecture of the proposed SSH honeypot is designed to provide a secure, scalable, and efficient framework for capturing and analyzing cyber-attacks. It consists of multiple interconnected components that work together to simulate an SSH environment, record attacker activities, and generate meaningful insights.



At the outer layer, the system interacts with external users or attackers through the internet. When an attacker attempts to access the system via the SSH port, the request is directed to the honeypot server instead of a real production server. This ensures that all malicious activities are safely contained within the decoy environment.

The core component of the architecture is the SSH Honeypot Engine, which emulates a real SSH service. It provides a fake login interface and a controlled shell environment that allows attackers to interact with the system. This module is designed to appear realistic so that attackers continue their actions without realizing they are being monitored.

Connected to the honeypot engine is the Data Logging Layer, which captures all incoming and outgoing activities. This includes login credentials, IP addresses, timestamps, and commands executed by the attacker. The captured data is then forwarded to the Data Storage Layer, where it is stored in structured formats such as databases or log files for further processing.

The Analysis Engine processes the stored data to identify attack patterns, detect brute-force attempts, and analyze attacker behavior. It converts raw data into meaningful insights that can help improve cyber-security strategies. This layer may also include basic statistical or machine learning techniques for enhanced analysis.

Finally, the Visualization and Monitoring Layer presents the analyzed data through dashboards, graphs, and reports. This allows administrators to monitor real-time attack activities and understand trends easily. Additionally, a Security and Isolation Layer ensures that the honeypot operates in a sandboxed or virtualized environment, preventing attackers from accessing real system resources.

VIII. STATISTICAL ANALYSIS

The performance of the proposed SSH honeypot system is evaluated using statistical metrics derived from the collected attack data. The analysis focuses on understanding attacker behavior, identifying attack patterns, and evaluating the effectiveness of the honeypot in capturing malicious activities. The evaluation is carried out in the following steps:

Step 1: Data Collection and Log Generation

The honeypot system continuously captures and stores data related to attacker interactions. The collected data includes

- Number of login attempts
- Successful and failed login attempts
- IP addresses of attackers
- Timestamps of attacks
- Commands executed during sessions

Step 2: Attack Frequency Analysis

The frequency of attacks is calculated over different time intervals (hourly, daily, or weekly). This helps in identifying peak attack periods.

Attack Frequency = Total Number of Attempts / Time Interval

This metric highlights when the system is most targeted.

Step 3: Calculation of Precision
This metric evaluates how often attackers successfully gain access to the honeypot.
Login Success Rate = Successful Logins / Total Login Attempts
It helps in understanding the effectiveness of attacker strategies and honeypot engagement.

Step 4: Failed Login Rate (Brute-Force Detection)
The failed login rate is used to detect brute-force attacks.
Failed Login Rate = Failed Attempts / Total Login Attempts
A high failed login rate indicates automated password-guessing attacks.

Step 5: IP Address Distribution Analysis
The number of attacks originating from each IP address is calculated.
IP Frequency = Number of Attempts per IP
This helps in identifying repeated attackers and potential malicious sources.

Step 6: Command Usage Analysis
The commands executed by attackers after login are analyzed using frequency distribution.
Command Frequency = Occurrence of Each Command / Total Commands
This helps in identifying common malicious actions such as file downloads or system probing.

Step 7: Session Duration Analysis
The duration of each attacker session is measured to understand engagement levels.
Average Session Time = Total Session Time / Number of Sessions
Longer sessions indicate deeper interaction with the honeypot.

Step 8: System Effectiveness Evaluation
The overall effectiveness of the honeypot is evaluated based on:
- Number of attacks captured
- Diversity of attack patterns
- Duration of attacker engagement
- Quality of collected data

This ensures that the system successfully attracts and monitors malicious activities.

IX. RESULTS AND DISCUSSION

The proposed SSH honeypot system was successfully deployed and monitored over a period of time to capture real-world attack data. During this observation phase, the system recorded a significant number of unauthorized access attempts, demonstrating that SSH services are continuously targeted by automated bots and malicious users. The honeypot effectively attracted attackers and logged detailed information such as login credentials, IP addresses, timestamps, and executed commands.

The results show a high frequency of brute-force login attempts, where attackers repeatedly tried different username and password combinations to gain access. Common usernames such as “root”, “admin”, and “user” were frequently observed, along with weak password attempts. A large proportion of login attempts resulted in failure, indicating the use of automated scripts or bots rather than manual attacks. However, in some cases, the honeypot allowed simulated successful logins, enabling deeper interaction and more detailed data collection.

Analysis of IP address data revealed that attacks originated from multiple geographic locations, indicating a distributed attack pattern. Certain IP addresses showed repeated attempts, suggesting persistent attack sources. This information can be useful for implementing IP-based blocking or enhancing firewall rules in real-world systems.

The command execution logs provided valuable insights into attacker behavior after gaining access. Common activities included system information gathering, checking network configurations, attempting to download malicious files, and executing suspicious shell commands. These patterns indicate that attackers often aim to explore the system, establish control, or deploy malware.

The session duration analysis showed that most attack sessions were short, indicating automated scanning activities. However, a few longer sessions were observed, suggesting more interactive attacks where the attacker actively explored the system. This highlights the effectiveness of the honeypot in

engaging attackers and collecting meaningful behavioral data.

Overall, the results demonstrate that the SSH honeypot system is effective in capturing and analyzing cyber-attacks in a controlled environment. The discussion highlights that such systems not only help in detecting threats but also provide deep insights into attacker strategies and techniques. These findings can be used to strengthen security policies, improve intrusion detection systems, and enhance overall cyber-security preparedness.

Despite its effectiveness, the system has certain limitations. It is primarily focused on SSH-based attacks and may not capture threats targeting other services. Additionally, highly sophisticated attackers may detect the honeypot environment and avoid interaction. Future improvements can include integrating machine learning techniques, expanding to multi-service honeypots, and implementing real-time alert systems.

Discussion

The deployment of the SSH honeypot system provides valuable insights into real-world cyber-attack behavior and highlights the growing threats targeting remote access services. The observed results clearly indicate that SSH services are continuously scanned and attacked by automated bots, primarily through brute-force techniques. This reflects the widespread use of weak credentials and the need for stronger authentication mechanisms in real systems.

One of the key observations from the analysis is the repetitive nature of login attempts using common usernames and passwords. This suggests that attackers rely heavily on predefined dictionaries and automated scripts rather than manual efforts. The high number of failed login attempts confirms the presence of brute-force attacks, while occasional successful logins (simulated by the honeypot) allow deeper analysis of attacker intentions and actions.

The study also reveals that attack sources are globally distributed, with multiple IP addresses attempting repeated access. This indicates the use of botnets or distributed attack systems, making it difficult to block threats using simple security measures. However, identifying frequently attacking IPs can still help in improving firewall configurations and access control policies.

Another important aspect is the behavior of attackers after gaining access. The command analysis shows that attackers often begin with system reconnaissance, such as checking system information, network configuration, and available resources. In some cases, attempts were made to download or execute malicious files, indicating possible intentions of system compromise or malware installation.

The variation in session durations further provides insight into attack types. Short sessions are typically associated with automated scanning tools, while longer sessions indicate human involvement or advanced attacks. This demonstrates that the honeypot is effective in capturing both basic and moderately advanced attack behaviors.

Despite its effectiveness, the system has certain limitations. Advanced attackers may detect that the environment is a honeypot and avoid further interaction, reducing the depth of collected data. Additionally, the system is limited to SSH-based attacks and does not cover other network services.

Overall, the discussion highlights that the SSH honeypot is a powerful tool for cyber-security research and threat analysis. It not only helps in understanding attacker strategies but also supports the development of stronger and more proactive defense mechanisms.

X. CONCLUSION AND FUTURE WORK

The proposed SSH honeypot system successfully demonstrates an effective approach for monitoring, capturing, and analyzing cyber-attacks targeting SSH services. By simulating a realistic environment, the system attracts attackers and records their activities without exposing actual system resources. The collected data provides valuable insights into attacker behavior, including brute-force login attempts, commonly used credentials, and post-login activities.

The results of the study highlight that most attacks are automated and follow predictable patterns, such as repeated login attempts using common usernames and passwords. Additionally, the system effectively captures attacker interactions, enabling deeper analysis of malicious commands and strategies. This proves that honeypots are a powerful tool for enhancing cyber-security by providing real-time threat intelligence and improving understanding of evolving attack techniques.

Overall, the project achieves its objective of developing a cost-effective and efficient SSH honeypot system. It contributes to strengthening cyber-security defenses by enabling proactive monitoring and analysis of threats. The insights gained from this system can help organizations improve their security policies, implement stronger authentication mechanisms, and prepare for potential cyber-attacks.

Future work:

Although the proposed system performs effectively, there are several areas for improvement and enhancement. Future work can focus on expanding the capabilities of the honeypot system and increasing its effectiveness in real-world scenarios.

One potential improvement is the integration of machine learning algorithms to automatically detect and classify attack patterns. This would enable intelligent threat detection and faster response to emerging cyber threats. Additionally, the system can be extended to support multi-protocol honeypots, allowing monitoring of other services such as HTTP, FTP, and Telnet, thereby providing broader security coverage.

Another area of enhancement is the implementation of a real-time alert system, which can notify administrators immediately when suspicious activities are detected. This would enable quicker response and mitigation of threats. The system can also be deployed in a cloud-based environment to improve scalability, accessibility, and performance.

Furthermore, advanced visualization techniques and dashboards can be incorporated to provide more detailed and interactive analysis of attack data. Improving the realism of the honeypot environment can also help in engaging sophisticated attackers for longer durations, resulting in richer data collection.

In conclusion, future enhancements will focus on making the system more intelligent, scalable, and comprehensive, thereby increasing its contribution to modern cyber-security solutions.

REFERENCES

[1] L. Spitzner, *Honeypots: Tracking Hackers*. Boston, MA, USA: Addison-Wesley, 2003.

[2] N. Provos and T. Holz, *Virtual Honeypots: From Botnet Tracking to Intrusion Detection*. Boston, MA, USA: Addison-Wesley, 2007.

[3] K. Scarfone and P. Mell, “Guide to intrusion detection and prevention systems (IDPS),” NIST Special Publication 800-94, 2007.

[4] C. Seifert and I. Welch, “Honeypots: Concepts, approaches, and challenges,” *IEEE Security & Privacy*, vol. 4, no. 5, pp. 72–76, 2006.

[5] F. Cohen, “The use of deception techniques: Honeypots and decoys,” in *Handbook of Information Security*, 2006.

[6] The Honeynet Project, “Know Your Enemy: Learning About Security Threats,” 2002.

[7] M. Nawrocki, M. Wählisch, T. C. Schmidt, C. Keil, and J. Schönfelder, “A survey on honeypot software and data analysis,” *arXiv preprint arXiv:1608.06249*, 2016.

[8] E. Alata, V. Nicomette, M. Kaâniche, M. Herrb, P. Laarouchi, and M. Dacier, “Lessons learned from the deployment of a high-interaction honeypot,” in *Proc. IEEE DSN Workshop*, 2006.

[9] G. Wagener, R. State, A. Dulaunoy, and T. Engel, “Self-adaptive high-interaction honeypots driven by game theory,” in *Proc. ACM Symp. Information, Computer and Communications Security*, 2009.

[10] S. Almeshekah and E. H. Spafford, “Planning and integrating deception into computer security defenses,” in *Proc. New Security Paradigms Workshop*, 2014.

[11] M. Bailey, E. Cooke, F. Jahanian, Y. Xu, and M. Karir, “A survey of botnet technology and defenses,” in *Proc. IEEE Conf. Cybersecurity*, 2009.

[12] D. Fraunholz, D. Krohmer, S. Duque Antón, and H. D. Schotten, “Demystifying deception technology: A survey,” *arXiv preprint arXiv:1804.06156*, 2018.

[13] S. Venkatesan and M. Albanese, “Evaluation of honeypots for intrusion detection,” in *Proc. IEEE Int. Conf. Intelligence and Security Informatics*, 2013.