www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 03, March 2025, pp : 1723-1730

e-ISSN : 2583-1062
Impact Factor : 7.001

HONEYPOTT3R: AN OPEN-SOURCE MULTI-LAYERED SECURITY ANALYSIS FRAMEWORK FOR HONEYPOT VULNERABILITY ASSESSMENT

Ayushman Bilas Thakur[1], Samrat Dey[2], Siddiqua Rahman[3], Aditya Chaurasia[4]

[1]Assistant Professor, Dept. of Computer Science & Engineering, Adamas University, India.

[2,3,4]Dept. of Computer Science & Engineering, Adamas University, India.

DOI: https://www.doi.org/10.58257/IJPREMS39220

ABSTRACT

Honeypots are deceptive security mechanisms designed to attract, detect, and analyze cyber threats by mimicking real systems. They are widely used in cybersecurity research and defense strategies to study attack patterns and adversary techniques. However, if not configured properly honeypots can be easily detected, making them potential targets for attackers to gain access to the main server. This paper introduces Honeypott3r, an open-source, multi-layered security analysis framework designed to systematically assess the security of honeypots. The framework focuses on SSH and HTTP-based honeypots, specifically Cowrie, Conpot, and Wordpot, which are deployed in Docker containers on AWS EC2 (Debian**)**. Honeypott3r performs detection, log evasion, privilege escalation, reverse exploitation, and denial-of-service (DoS) simulations, leveraging tools like Trivy, Bandit, Safety-CLI, Nmap, Nikto, and WPScan. It also searches for Metasploit modules to identify potential exploits. The collected data is stored in MongoDB and visualized through an interactive dashboard for further analysis. The results reveal various weaknesses in honeypots, highlighting the need for improved deception techniques. This research contributes to the cybersecurity community by providing a structured approach to honeypot vulnerability assessment, helping defenders enhance their security posture and mitigate risks more effectively.

Keywords: Honeypots, Cowrie, Conpot, Wordpot, Vulnerability Assessment

1. INTRODUCTION

Honeypots play a crucial role in cybersecurity by acting as deceptive systems designed to attract, monitor, and analyze malicious activities. They help security researchers and organizations gain insights into attack patterns, identify emerging threats, and enhance defensive strategies. However, while honeypots are primarily used for detection and intelligence gathering, they can also introduce security risks if not properly configured or monitored. Attackers, instead of being trapped, may exploit vulnerabilities in honeypots to gain unauthorized access, evade detection, or launch further attacks.

Several research efforts have focused on honeypot fingerprinting, where attackers attempt to detect and evade honeypots based on unique system characteristics. While many papers exist on identifying and bypassing honeypots, limited studies have systematically explored the vulnerabilities within honeypots themselves. Most existing works concentrate on analyzing attacker behavior rather than assessing whether honeypots can be exploited like real-world systems.

To address this gap, we introduce Honeypott3r, an open-source, multi-layered security analysis framework designed to assess vulnerabilities in honeypots.

The framework targets SSH and HTTP-based honeypots, specifically Cowrie, Conpot, and Wordpot, deployed in Docker containers on AWS EC2 (Debian). The honeypots were selected on the basis of having a larger user base and being commonly known. Honeypott3r performs vulnerability detection, log evasion, privilege escalation, reverse exploitation, and denial-of-service (DoS) simulation using various security tools. The findings are stored in MongoDB and visualized through an interactive dashboard for easy analysis.

While there might be existing works on evaluating honeypot security, we could not find any comprehensive framework that automates multi-layered vulnerability assessment of honeypots, particularly in the areas of code security, privilege escalation, and reverse exploitation.

Therefore, we present Honeypott3r as a novel approach to systematically test honeypots for weaknesses, demonstrating the need for enhanced deception techniques. This research aims to contribute to the cybersecurity community by offering a structured approach to honeypot vulnerability assessment, helping organizations enhance their defensive capabilities and reduce potential security risks.

www.ijprems.com
editor@ijprems.com
INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 03, March 2025, pp : 1723-1730
e-ISSN : 2583-1062
Impact Factor : 7.001

2. RELATED WORK

Honeypots have been a key area of research in cybersecurity, playing a crucial role in attack detection, threat intelligence, and intrusion analysis. Various studies have explored different aspects of honeypot security, including their classification, vulnerabilities, and resistance to fingerprinting. While much research has been conducted on detecting and bypassing honeypots, limited studies have systematically analyzed their security weaknesses from an attacker's perspective.

Titarmare et al. [1] provided an overview of honeypot systems, categorizing them into low, medium, and high-interaction honeypots. Low-interaction honeypots mimic only a few services, making them lightweight but easier to detect, while high-interaction honeypots simulate full-fledged systems, providing deeper insights into attacker behavior at the cost of increased risk if compromised. Their study highlights the importance of choosing the appropriate honeypot type based on security objectives.

Javid and Lighvan [2] examined honeypots' vulnerabilities to backdoor attacks, emphasizing that while honeypots serve as deceptive security mechanisms, they can themselves become targets for exploitation. Their research pointed out that conventional security technologies rely on known attack signatures or statistical methods, whereas honeypots are designed to capture novel attacks. However, if attackers compromise the host system, the honeypot loses its effectiveness. The study specifically analyzed Kfsensor, a Windows-based honeypot, and demonstrated how attackers could exploit its weaknesses, leading to unauthorized access.

Hetzler et al. [3] investigated the effectiveness of SSH honeypots, particularly Cowrie, by comparing an uncloaked version (default configuration) with a cloaked version (modified setup). Their study found that attackers rely on specific indicators—such as predefined usernames, filesystem structures, and command outputs—to identify and evade honeypots. The cloaked version of Cowrie, which introduced modifications like removing default users, altering filesystem structures, and adding unique command responses, proved to be more resilient against attacker evasion techniques and attracted significantly more malicious interactions.

Srinivasa et al. [4] proposed a multistage framework for honeypot fingerprinting, incorporating probe-based and metascan-based techniques. Probe-based methods include port scanning, banner grabbing, HTTP response analysis, SSL/TLS certificate checking, protocol handshakes, and static command responses, all of which can help attackers distinguish real systems from honeypots. The metascan-based method leverages Shodan and Censys to identify whether a system is a honeypot based on publicly available information. Their framework highlighted the growing sophistication of attackers in identifying and bypassing honeypots, reinforcing the need for better deception techniques.

While prior research extensively examines honeypot fingerprinting, attack detection, and security risks, there is a noticeable gap in systematic vulnerability assessment of honeypots from a security perspective. Existing studies have primarily focused on how attackers evade detection, but little work has been done on evaluating whether honeypots can be exploited just like real-world systems.

To address this gap, we introduce Honeypott3r, a multi-layered, open-source security analysis framework for evaluating vulnerabilities in honeypots. Unlike existing studies that focus solely on attacker behavior, Honeypott3r performs privilege escalation testing, log evasion, reverse exploitation, and denial-of-service (DoS) simulations against honeypots like Cowrie, Conpot, and Wordpot deployed in Docker containers on AWS EC2 (Debian). The framework integrates automated vulnerability scanning, post-exploitation techniques, and attack simulations, storing results in MongoDB for visualization through an interactive dashboard.

While some previous works may have explored aspects of honeypot security, we did not find a comprehensive framework that automates multi-layered vulnerability assessment. Therefore, we position Honeypott3r as a novel approach that fills this gap, providing a structured methodology to test honeypots for weaknesses. Our research contributes to the cybersecurity community by offering a systematic evaluation framework that can help organizations enhance honeypot deception techniques and mitigate potential security risks. We have not found any prior research that directly aligns with our approach of systematic vulnerability assessment and exploitation of honeypots. However, given the increasing focus on honeypot security, deception techniques, and adversarial machine learning, there may be ongoing or unpublished work exploring similar ideas. Our study aims to bridge this gap by introducing an automated security evaluation framework that can assess honeypot weaknesses from an attacker's perspective, providing valuable insights for both researchers and security practitioners.

www.ijprems.com
editor@ijprems.com
INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 03, March 2025, pp : 1723-1730
e-ISSN : 2583-1062
Impact Factor : 7.001

3. METHODOLOGY

Honeypott3r is a CLI-based security analysis tool developed in Python, designed to evaluate the security of honeypots through automated scanning and exploitation techniques. It focuses on identifying vulnerabilities in open-source honeypots, which are widely used for threat intelligence and deception-based security. The tool requires minimal user input, including the test name, honeypot type, IPv4 address, port, username, password (for SSH-based honeypots), and HTTP link (for web-based honeypots). Once these details are provided, the tool performs a structured security assessment to uncover weaknesses.

The framework automates multiple security tests, such as privilege escalation, log evasion, reverse exploitation, and denial-of-service (DoS) simulations, to determine the resilience of honeypots against real-world attacks. By leveraging scanning techniques and attack simulations, it provides valuable insights into how adversaries can exploit honeypot weaknesses. The results are systematically collected and stored for further analysis, helping security researchers and practitioners enhance honeypot defenses.

3.1 Honeypot Deployment

In this research, we utilize three open-source honeypots: Cowrie, Conpot, and Wordpot, each serving a distinct purpose in simulating real-world attack surfaces. Cowrie, an SSH and Telnet honeypot, is used to analyze brute-force attacks and unauthorized access attempts. Conpot, an ICS/SCADA honeypot, is deployed specifically for HTTP-based interaction, allowing us to assess its resilience against web-based threats. Wordpot, designed to emulate a WordPress-based environment, is included due to the widespread use of WordPress in web applications, making it a frequent target for cyber attacks.

To host these honeypots, we utilize an AWS EC2 instance running Debian. Cowrie and Conpot are deployed using their official Docker images, ensuring quick and consistent deployment with minimal configuration overhead. However, Wordpot does not have an official Docker image, so we created a custom Docker image to containerize and deploy it efficiently. The honeypots are configured with open network ports, allowing unrestricted inbound traffic to simulate a real-world attack environment. This setup ensures that the honeypots can effectively capture various attack attempts and security events for further analysis.

3.2 Vulnerability Assessment Technique

The Honeypott3r tool is designed to run on any Debian-based Linux distribution, with Kali Linux being the preferred choice due to its pre-installed security tools such as Nmap, Nikto, and Metasploit Framework (MSF). The tool can be set up by cloning the repository from GitHub and installing dependencies using the provided install.sh script, which ensures that all required libraries and external tools are installed. Once installed, the tool is executed using ‘sudo honeypott3r’, which automatically starts the tool, initiates the interactive dashboard, and sets up MongoDB for data storage.

The tool provides four primary commands:

start – Initializes the user input process, where details such as test name, honeypot type, IPv4 address, port, username and password (for SSH), and HTTP link (for HTTP-based honeypots) are provided.

scan – Initiates the scanning process and performs multiple security assessments.

reset – Clears user inputs to allow a fresh test configuration.

exit – Terminates the tool.

All user inputs are stored in a JSON file, allowing different modules within the tool to access and use them throughout the assessment process.

In Honeypot Detection assessment Honeypott3r identifies honeypots by analyzing the network responses and system behaviors. It utilizes Netcat to extract banners, responses, and interaction flags from the honeypots and compares them against a predefined dataset of known honeypot signatures. If the extracted details match any of the predefined honeypot fingerprints, the system flags it as a honeypot.

For Command Injection & Data Leakage assessment in SSH honeypots, the tool executes a series of predefined system commands and compares the responses with those of a legitimate system. If the output differs significantly, it indicates the presence of an exploitable weakness. Similarly, for HTTP honeypots, the tool interacts with various endpoints and analyzes the server responses to detect unexpected data exposures, which may include configuration leaks, sensitive file access, or other exploitable data leaks.

Although direct log evasion techniques were difficult to implement, an alternative approach using log flooding was adopted. This technique involves overloading the honeypot's logging mechanism with a massive number of irrelevant

www.ijprems.com
editor@ijprems.com
INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 03, March 2025, pp : 1723-1730
e-ISSN : 2583-1062
Impact Factor : 7.001

entries, making it difficult for defenders to differentiate real attacks from noise. This method can hinder forensic analysis by security teams.

To identify potential privilege escalation paths, Trivy, a container security scanner, is used to analyze the Docker images of deployed honeypots. The scan results include identified vulnerabilities (CVEs) in the Docker environment, which could provide potential avenues for escalating privileges from the honeypot environment to the underlying host system. If critical vulnerabilities are found, further research can be conducted to determine whether they can be exploited for root access.

In Reverse Exploitation analysis since the honeypots are open-source, their source code and dependencies can be analyzed for vulnerabilities. The tool uses Bandit, a Python security scanner, to detect weaknesses in the honeypots' codebase, such as unsafe function usage, hardcoded credentials, or command injection risks. Additionally, Safety-CLI is used to scan the third-party Python packages installed within the honeypots, identifying known vulnerabilities in the dependencies. This approach allows for discovering inherent security flaws in honeypot implementations, which can potentially be exploited.

In Denial-of-Service (DoS) Simulation, it assesses the resilience of honeypots against resource exhaustion attacks, the tool employs Hping3 and SlowHTTPTest. Hping3 is used to generate high volumes of TCP/UDP/ICMP traffic, simulating bandwidth depletion attacks, while SlowHTTPTest is used to conduct Slowloris attacks that consume the server's connection pool, preventing legitimate users from accessing the honeypot. These tests evaluate the honeypots' ability to handle sustained attack traffic and identify any potential weaknesses.

After the primary assessment, additional security scans are performed for further vulnerability identification. Nmap is used for general network and port scanning across all honeypots, while Nikto is specifically used for scanning HTTP-based honeypots like Conpot and Wordpot. If Wordpot is detected, WPScan is used to analyze WordPress-specific vulnerabilities.

Finally, any discovered CVEs from previous scans are cross-referenced with the Metasploit Framework (MSF) to check for existing exploit modules. If matching exploits are found, they can be executed to test real-world attack feasibility.

By integrating these techniques, Honeypott3r provides a structured and automated approach to evaluating the security of honeypots from an attacker's perspective.

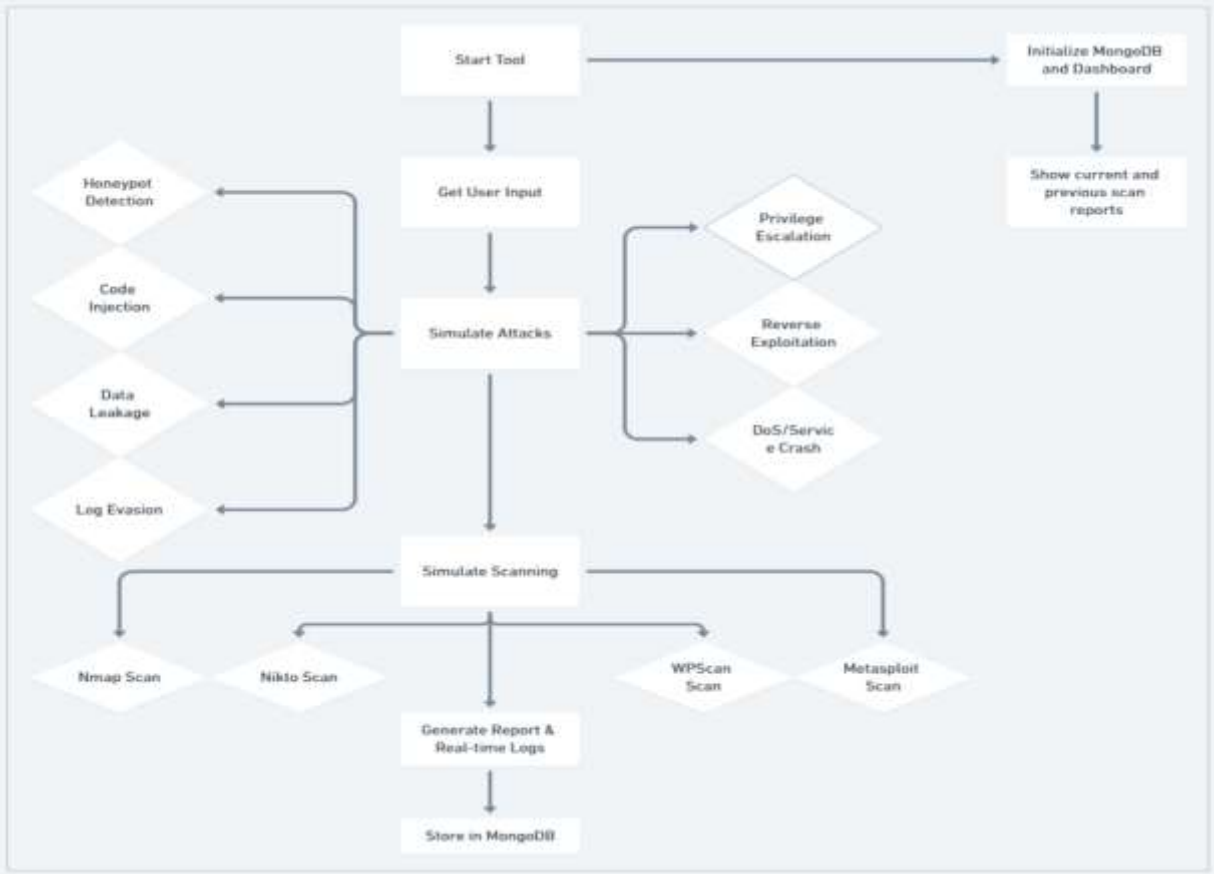

Figure 1: Workflow Overview

www.ijprems.com
editor@ijprems.com
INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 03, March 2025, pp : 1723-1730
e-ISSN : 2583-1062
Impact Factor : 7.001

3.2 Data Collection & Dashboard

Honeypott3r ensures structured data collection by storing the assessment results in a well-defined JSON format along with the complete terminal output in a log file. Both these data components are stored securely in MongoDB using GridFS, which allows efficient storage and retrieval of large files. MongoDB was selected for ease of storing logs in JSON format. This approach ensures that all scan results, logs, and historical data remain accessible for further analysis and review.

For visualization and easy access to the collected data, Honeypott3r provides a web-based dashboard accessible at http://localhost:5050/. The dashboard presents a user-friendly interface where both the latest and previous reports can be viewed, allowing users to track vulnerability trends, compare assessments, and gain deeper insights into the security posture of deployed honeypots.

Figure 2: Scan History Table

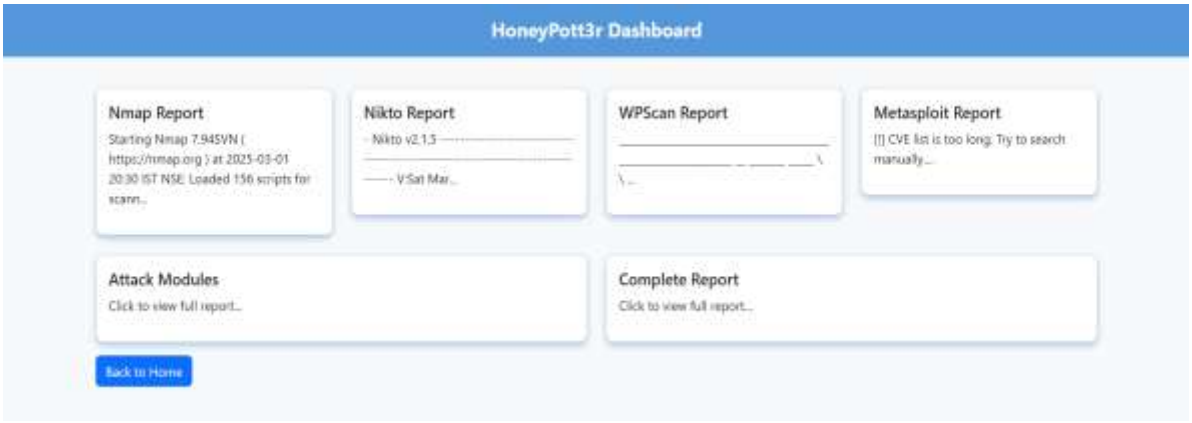

Figure 3: Detailed Scan Report

4. DISCUSSION AND ANALYSIS

The tool provides a structured approach to assessing honeypots by integrating multiple scanning, exploitation, and reporting functionalities. Below is a summary of its key features:

Table 1: Features Overview

Category	Feature	Description
Detection	Honeypot Detection	Identifies and fingerprints honeypots deployed in the network.
Scanning	Vulnerability Scanning	Integrates tools like Nmap, Nikto, WPscan, and Metasploit to find vulnerabilities.
Exploitation		
1	Code Injection Attacks	Simulates injection attacks (e.g., SQL, Command) to evaluate honeypot defenses.
2	Data Leakage Exploitation	Attempts to extract sensitive information stored in the honeypot.
3	Reverse Exploitation	Uses the honeypot's own vulnerabilities to launch counterattacks.
4	Privilege Escalation	Tests for vulnerabilities that allow unauthorized access to higher privilege levels.
Stealth	Log Evasion Techniques	Implements methods to bypass or manipulate honeypot

		logging mechanisms.
Simulation	DoS Attack Simulation	Simulates denial-of-service attacks to test the resilience of the honeypot.
Configuration	Customizable Configurations	Allows easy configuration of scan settings and attack parameters using YAML files.
Reporting	Integrated Reporting	Logs results of scans, exploits, and generates a detailed final report for analysis.
Storage	MongoDB Localhost Storage	Stores scan reports and logs in a MongoDB database for future reference.
Visualization	Web UI Dashboard	Provides a Flask-based web dashboard to view scan reports and analysis in real time.
Tool Integration	Multi-Tool Integration	Seamlessly integrates external tools like Nmap, Metasploit, Nikto, and WPscan for enhanced testing.
Usability		
1	User-Friendly Interface	Centralized `main.py` script to manage all operations with a simple CLI or menu interface.
2	Modular Design	Organized file structure for easy scalability and maintenance.

4.1 Key Insights from the Study

One of the major findings of this study is that honeypots are highly static in nature, making them vulnerable to fingerprinting and detection. The honeypots tested exhibited predictable behaviors, allowing easy identification through banner grabbing and protocol inconsistencies. Static honeypots tend to return the same responses to specific queries, making them stand out in a network scan. To counteract this, implementing dynamic honeypots that randomize responses and simulate real user behavior would enhance authenticity and reduce detection risks.

Another crucial observation is the vulnerability of honeypots to log flooding and denial-of-service (DoS) attacks. Because honeypots depend on logs to capture attacker behavior, excessive log generation can quickly deplete storage and make it difficult to filter out genuine threats from noise. Some honeypots also exhibited poor rate-limiting mechanisms, making them susceptible to excessive requests that could lead to system crashes. To mitigate these issues, rate-limiting should be enforced, log management must incorporate anomaly detection mechanisms, and honeypots should be deployed in a distributed manner to withstand DoS attacks more effectively. The study also found that honeypots themselves can be exploited due to outdated dependencies and weak configurations. Some honeypots allowed privilege escalation, making it possible for attackers to gain control over the system. Additionally, attackers could use compromised honeypots as stepping stones to pivot into real networks, posing significant security risks. Regular updates, sandboxing, and strict access controls are necessary to prevent attackers from turning honeypots into weapons against legitimate systems. Furthermore, while the tool successfully performed exploitation and attack simulations, its effectiveness was somewhat limited by its protocol support. The attacks were primarily conducted over HTTP and SSH, while other protocols like RDP, FTP, SMB, and DNS were not included in the scope. Expanding the tool's capabilities to cover multiple protocols and integrating machine learning-based attack pattern recognition would greatly enhance its functionality.

4.2 Limitations of the Tool

Despite its extensive feature set, the tool has certain limitations that need to be addressed. One key constraint is its limited protocol support, as it only focuses on HTTP and SSH honeypots. Expanding support for RDP, SMB, FTP, and DNS would make the tool more versatile. Additionally, most tests were conducted on low-interaction honeypots, which provide limited attack simulation capabilities. Incorporating high-interaction honeypots would allow for a more realistic attack scenario and deeper analysis. Another limitation is performance constraints, particularly on lower-end systems where extensive scanning and exploitation can be resource-intensive. Optimizing scanning algorithms, incorporating multithreading, and offloading computations to cloud-based environments would improve performance and scalability. The tool also lacks advanced evasion techniques, making it easy for security teams to detect its scanning activity. Introducing randomized responses and mimicry techniques would help improve stealth. Additionally, the tool does not include automated correlation of attack patterns across different honeypots, which

limits its ability to analyze large-scale attacker behavior. Integrating AI-driven threat intelligence would allow the tool to identify patterns and detect anomalies more effectively. Lastly, scalability remains a challenge, as the tool is primarily designed for individual honeypots rather than large-scale deployments. Using containerized environments like Docker or Kubernetes would enhance its ability to test multiple honeypots simultaneously.

4.3 Ethical Considerations

Honeypott3r is designed strictly for research and security assessment purposes only. It should not be used for unauthorized testing on production systems. All experiments and attack simulations conducted in this study were performed **on our own AWS EC2 instances** and **our own resources**, ensuring that no unauthorized systems or third-party networks were affected. Users should ensure compliance with organizational and legal guidelines before deploying attack simulations. Unauthorized use of Honeypott3r on external systems may violate legal and ethical guidelines, including **Computer Fraud and Abuse Act (CFAA)** regulations and organizational security policies.

5. FUTURE WORK

In future enhancements for the framework will focus on improving modularity and scalability to support a broader range of attack scenarios. Extending protocol coverage beyond SSH and HTTP, integrating real-time threat intelligence, and incorporating automated log analysis will enhance its effectiveness. Performance optimizations will be made to reduce resource consumption on lower-end systems. Additionally, enhancing the dashboard with advanced visualization and interactive reporting will provide deeper insights for security analysis.

6. CONCLUSION

This study presents a comprehensive approach to honeypot assessment, integrating various scanning, exploitation, and evasion techniques. The developed tool successfully automates honeypot detection, vulnerability scanning, privilege escalation testing, and reporting, offering a structured methodology to evaluate honeypot security. The MongoDB-backed storage and Flask-based dashboard enable efficient data management and visualization of reports, ensuring both real-time and historical analysis of security assessments.

The findings reveal that many honeypots exhibit static characteristics, making them easily detectable by adversaries. Implementing dynamic deception mechanisms, such as adaptive banners, realistic file structures, and unpredictable responses, can significantly enhance honeypot effectiveness. Additionally, weaknesses such as log flooding vulnerabilities, privilege escalation risks, and reverse exploitation possibilities underscore the need for regular updates, secure configurations, and robust monitoring. Ideally, honeypots should not only simulate real systems convincingly but also resist fingerprinting attempts and detect unauthorized activity proactively.

While the tool provides an extensive evaluation framework, its current limitations include support for only SSH and HTTP-based honeypots. Expanding coverage to additional protocols and incorporating more advanced detection methodologies will improve its versatility. Performance optimization for lower-end systems and the integration of machine learning-driven anomaly detection could further enhance its utility in real-world deployments.

Overall, this research highlights the evolving nature of honeypot security and the necessity for continuous adaptation against sophisticated attackers. By leveraging scalable architectures, automation, and advanced threat intelligence, honeypots can serve as powerful assets in cybersecurity, contributing to proactive threat detection and mitigation strategies.

7. REFERENCES

[1] Neha Titarmare, Nayankumar Hargule, Anand Gupta, “An Overview of Honeypot Systems”, International Journal of Computer Sciences and Engineering, Vol. 7, Issue 2, Feb 2019, pp. 394-397

[2] Farshad Javid, Mina Zolfy Lighvan, “Honeypots Vulnerabilities to Backdoor Attack”, 2021 IEEE International Conference on Information Security and Cryptology (ISCTURKEY), Dec 2021, pp. 161-166

[3] Connor Hetzler, Zachary Chen, Tahir Khan, "Analysis of SSH Honeypot Effectiveness", Advances in Information and Communication, Proceedings of the 2023 Future of Information and Communication Conference (FICC), Volume 2, March 2023, pp. 759-782

[4] S. Srinivasa, J. M. Pedersen, and E. Vasilomanolakis, "Gotta catch 'em all: a Multistage Framework for honeypot fingerprinting," arXiv preprint arXiv:2109.10652v1, Sep. 2021. [Online]. Available: https://arxiv.org/abs/2109.10652

[5] Xingyuan Yang, Jie Yuan, Hao Yang, Ya Kong, Hao Zhang, and Jinyu Zhao, "A Highly Interactive Honeypot-Based Approach to Network Threat Management," Future Internet, Mar 2023. Available: https://doi.org/10.3390/fi15040127

[6] M.R. Amal and P. Venkadesh, "Review of Cyber Attack Detection: Honeypot System," Webology, Vol. 19, No. 1, Jan 2022, pp. 5497-5514.

[7] Faldi Faldi, Dinamita Romadoni, Muhammad T Sumadi, "THE IMPLEMENTATION OF NETWORK SERVER SECURITY SYSTEM USING HONEYPOT," JIKO (Jurnal Informatika dan Komputer), Vol. 6, No. 2, Aug 2023, pp. 122-130.

[8] Matthew L. Bringer, Christopher A. Chelmecki, and Hiroshi Fujinoki, "A Survey: Recent Advances and Future Trends in Honeypot Research," International Journal of Computer Network and Information Security (IJCNIS), Vol. 4, No. 10, Sep. 2012, pp. 63-75.

[9] Pavol Sokol, Jakub Míšek, and Martin Husák, "Honeypots and honeynets: issues of privacy," EURASIP Journal on Information Security, Feb 2017. [Online]. Available: https://jis-eurasipjournals.springeropen.com/articles/10.1186/s13635-017-0057-4

[10] Anushka Purwar, Diksha Soni, Gaurangi Rawat, Nisha Yadav, and Vanshika Gupta, "HONEYPOT-BASED DATA BREACH AVOIDANCE SYSTEM," International Journal of Engineering Applied Sciences and Technology, Vol. 8, Issue 12, Apr 2024, pp. 362-366.

[11] Marwan Abbas-Escribano and Hervé Debar, "An improved honeypot model for attack detection and analysis," The 18th International Conference on Availability, Reliability and Security (ARES), Aug 2023, pp. 1-10. DOI: 10.1145/3600160.3604993.

[12] HoneyPott3r: Project Documentation, GitHub Wiki. Available: https://github.com/3rr0r-505/HoneyPott3r/wiki