J-INTECH (Journal of Information and Technology)
Accredited Sinta 4 Ministry of Higher Education, Science and Technology
Republic of Indonesia SK No. 10/C/C3/DT.05.00/2025
E-ISSN: 2580-720X || P-ISSN: 2303-1425

J-INTECH
Journal of Information and Technology

Application of Honeypot in Network Security for Detecting Cyber Attacks on it Infrastructure

Carlos Susanto[1*], Moh. Ali Romli [2]
[1,2]Universitas Teknologi Yogyakarta, Fakultas Sains & Teknologi, Program Studi Informatika, Jl. Siliwangi, Sendangadi, Kec. Mlati, Kab. Sleman, Daerah Istimewa Yogyakarta, Indonesia

Keywords
AI Log Analysis; Cyberattack Detection; Honeypot; Intrusion Monitoring; Network Security

***Corresponding Author:**
carlos.5220411101@student.uty.ac.id

Abstract

The high security risks that are susceptible to hacking and exploitation by malicious actors to steal data or information often arise due to a lack of awareness regarding the critical importance of implementing deceptive network security using honeypots. Negligence can create vulnerabilities that are easily exploited, allowing attackers to initiate breaches. A notable network security approach involves using Honeypots, a method that creates a decoy server to mimic an authentic one. Honeypots are deliberately engineered to attract the attention of cyber attackers and facilitate their access to the trap server, thereby enabling the monitoring and analysis of their activities without compromising the integrity of the primary server. To achieve optimal network security, comprehensive testing of Honeypots is essential. This testing process serves as a fundamental metric in evaluating the efficacy and performance of Honeypot systems in mitigating cyber threats.

1. Introduction

The increasing sophistication and frequency of cyberattacks have posed significant risks to critical infrastructure, prompting the need for advanced and proactive cybersecurity measures. In response to these evolving threats, organizations are increasingly adopting innovative techniques to detect and mitigate cyber risks. One such technique is the implementation of honeypots intentionally vulnerable systems designed to attract and monitor malicious activities[1]. By simulating a vulnerable environment, honeypots provide an opportunity to observe the tactics and behaviors of cybercriminals in a controlled setting, contributing valuable data to the field of cybersecurity[2].

However, despite the effectiveness of honeypots in capturing attack data, there remains a significant gap in the ability to classify and assess the severity of these threats. Traditional security systems often fail to distinguish between high-risk intrusions and minor or non-threatening activities, leading to inefficient threat response strategies. This research aims to address this limitation by implementing a honeypot system designed to emulate a vulnerable web server, track attacker behavior, and analyze the resulting logs with the assistance of Artificial Intelligence (AI), specifically by leveraging the analytical capabilities of the Gemini API[3]. By classifying the severity of the detected threats, this study seeks to provide cybersecurity practitioners with a more refined tool for prioritizing and responding to cyber threats effectively[4]. The primary research question guiding this study is: How effectively can a honeypot system, enhanced by AI-driven log analysis using the Gemini API, detect, classify the severity of, and provide mitigation recommendations for simulated cyberattacks on IT infrastructure.

The motivation for this research stems from the growing need for a more sophisticated approach to threat detection, one that not only identifies intrusions but also categorizes them according to their potential impact. Existing systems predominantly focus on detecting malicious activities without offering sufficient insights into the risk level associated with each event[5]. By utilizing the Gemini API to analyze the attack logs generated by the honeypot, this study intends to enhance the accuracy and efficiency of threat classification, thereby enabling more informed decision-making in cybersecurity defense[6].

While prior research has explored the use of honeypots in capturing and monitoring cyberattacks, and some have begun to explore AI integration, many studies have not fully leveraged the capabilities of advanced large language models for deep log analysis[7]. This research distinguishes itself by integrating AI-powered log analysis into the honeypot framework, offering a novel approach to classifying the severity of detected intrusions[8]. Through this integration, the study aims to make a significant contribution to the existing body of knowledge in cybersecurity by advancing the capabilities of honeypot systems and enhancing their effectiveness in real-world applications[9].

By improving the way threats are classified and prioritized, this research provides an important step toward more efficient cybersecurity practices. The integration of AI technologies into honeypot-based detection systems holds the potential to transform how organizations detect, analyze, and respond to cyber threats, ultimately strengthening their defenses against the growing number of sophisticated cyberattacks[10].

2. Research Method

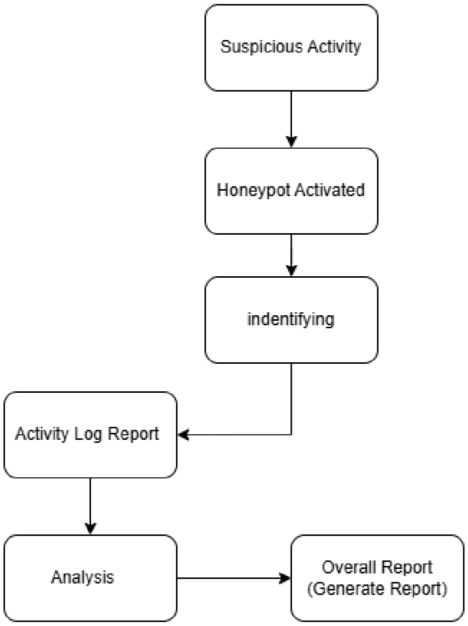


Figure 1. Research Stages

This section provides a detailed account of the methods employed to investigate the use of *honeypot* systems for detecting and analyzing *cyberattacks*. The methodology outlined here is designed to ensure that future researchers can replicate this study effectively. It includes the research flow, tools used, data collection methods, data validation procedures, and the study environment[11].

In this research, the "respondents" are simulated cyberattackers, not human participants. These simulated attacks are designed to mimic various real-world threats, such as unauthorized access, and malware deployment. The honeypot system is set up to interact with these simulated attackers, providing a controlled environment to track attack behaviors and gather valuable data[12].
The type of simulated *cyberattacks* that will be observed include:

a) Port Scanner: A scanning technique used to identify open ports on a network, often the first step in attempting to gain unauthorized access. SSH Console
b) SSH-Console Attacks: Unauthorized attempts to gain access to a system via SSH (Secure Shell) using brute-force or dictionary-based password guessing. Botnet Attacks
c) Metasploit: A widely used penetration testing tool designed to exploit known vulnerabilities. In this research, it will be used to simulate advanced attack techniques targeting the *honeypot*.
d) Botnet Attacks: Automated attacks conducted by a network of compromised devices controlled by the attacker, typically used for large-scale DDoS (Distributed Denial of Service) or spamming operations.

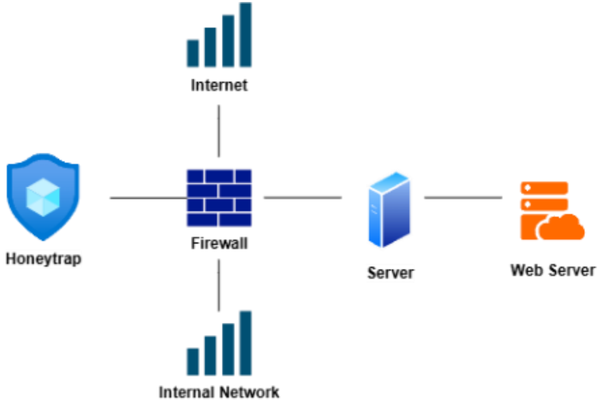


Figure 2. Network Security Schema with Honeytrap

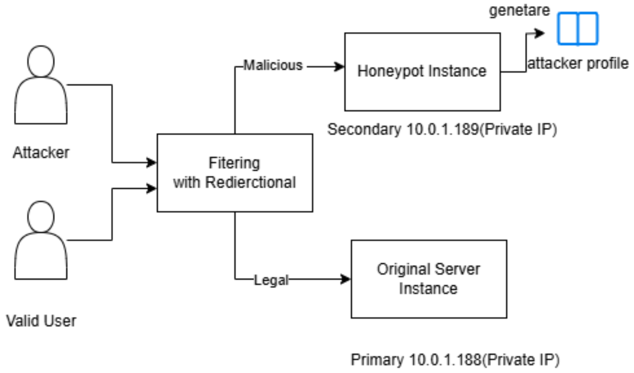


Figure 3. Honeypot As a Service in Cloud

The diagram represents a network security system using a honeypot (specifically, a honeypot instance) as part of a cloud-based service. The cloud service is provided by Domainesia with the specifications of the cloud infrastructure including the SUPER package. The VM specifications include 5 GB of Cloud SSD storage, 100% CPU resource allocation (equivalent to 1 vCPU), and 1 GB of RAM. Here's an explanation of the components and flow shown in the diagram:

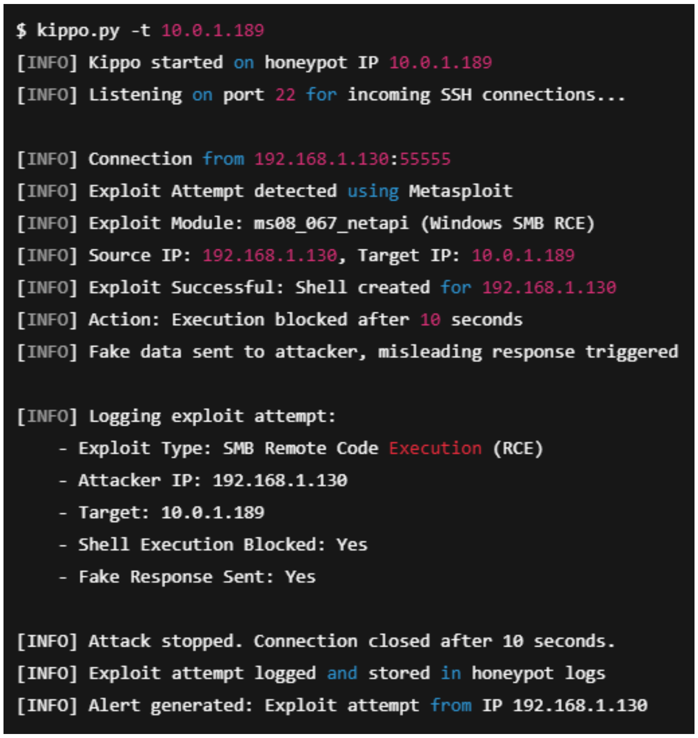
a) Malicious User: This represents an attacker who attempts to access the system or service.
b) Legal User: This represents a legitimate user who is trying to access the original server instance.
c) Filtering with Redirection: When a user attempts to access the system, a filtering mechanism is used. This filter directs malicious traffic (attackers) to the *honeypot instance* while allowing legitimate users to access the actual server.
d) Honeypot Instance: This is a decoy system set up to look like a vulnerable target to attract malicious users. It doesn't serve any real purpose for the attacker other than to gather information about their methods. When an attacker accesses this instance, their activity is logged, and an *attacker profile* is created. This helps in tracking and analyzing the attacker's behavior without risking the real system.

e) Original Server Instance: This is the actual system or service that legitimate users intend to access. Malicious users are redirected away from it and instead interact with the *honeypot*.

Data collection in this study focuses on recording the activities of simulated *cyberattackers* interacting with the *honeypot*. The following tools and methods will be employed for data collection[13]:

a) Honeypot System Setup: The *honeypot* system is deployed using tools such as *Honeyd* or *Kippo*, which simulate vulnerable services and are designed to attract attacks. Honeypot is configured within a virtual machine (VM) environment on cloud infrastructure to ensure isolation from live systems.

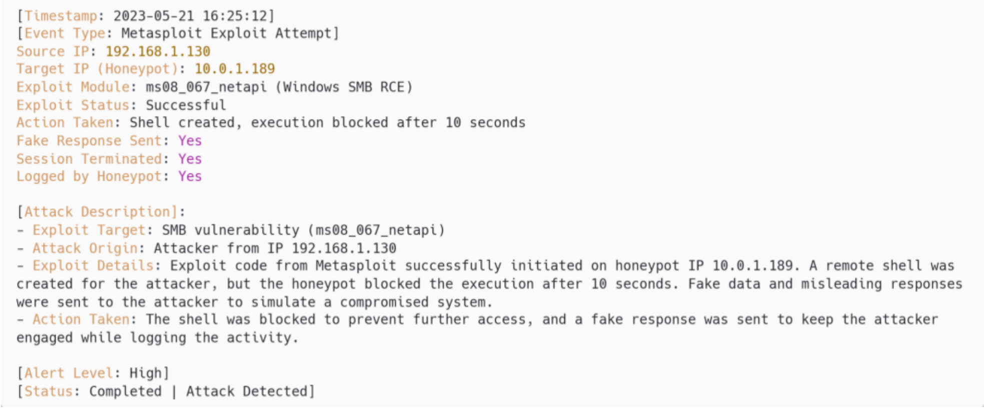
b) Simulated Attacks: Attack scenarios will be created using tools like *Metasploit*, *OWASP ZAP*, and custom scripts. These tools simulate various types of attacks to ensure that the *honeypot* system encounters a wide range of malicious activities.



```
$ kippo.py -t 10.0.1.189
[INFO] Kippo started on honeypot IP 10.0.1.189
[INFO] Listening on port 22 for incoming SSH connections...

[INFO] Connection from 192.168.1.130:55555
[INFO] Exploit Attempt detected using Metasploit
[INFO] Exploit Module: ms08_067_netapi (Windows SMB RCE)
[INFO] Source IP: 192.168.1.130, Target IP: 10.0.1.189
[INFO] Exploit Successful: Shell created for 192.168.1.130
[INFO] Action: Execution blocked after 10 seconds
[INFO] Fake data sent to attacker, misleading response triggered

[INFO] Logging exploit attempt:
    - Exploit Type: SMB Remote Code Execution (RCE)
    - Attacker IP: 192.168.1.130
    - Target: 10.0.1.189
    - Shell Execution Blocked: Yes
    - Fake Response Sent: Yes

[INFO] Attack stopped. Connection closed after 10 seconds.
[INFO] Exploit attempt logged and stored in honeypot logs
[INFO] Alert generated: Exploit attempt from IP 192.168.1.130
```

Figure 4. Response to a Metasploit Exploit Attempt

c) Log Collection: During the attack simulations, all activities performed by the simulated attackers will be logged in .log files[14]. These logs will include detailed records, such as IP addresses, types of attacks, timestamps, and other attack-related data. Even if the attacker successfully compromises the service, the system will still log all activities, including attack techniques, attack duration, and penetration methods, ensuring that data on successful attacks is still captured[15].

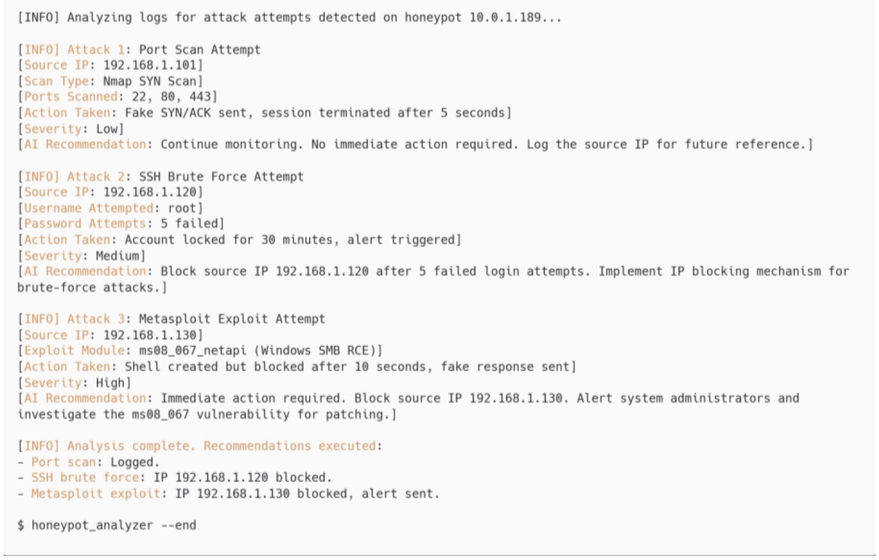


```
[Timestamp: 2023-05-21 16:25:12]
[Event Type: Metasploit Exploit Attempt]
Source IP: 192.168.1.130
Target IP (Honeypot): 10.0.1.189
Exploit Module: ms08_067_netapi (Windows SMB RCE)
Exploit Status: Successful
Action Taken: Shell created, execution blocked after 10 seconds
Fake Response Sent: Yes
Session Terminated: Yes
Logged by Honeypot: Yes

[Attack Description]:
- Exploit Target: SMB vulnerability (ms08_067_netapi)
- Attack Origin: Attacker from IP 192.168.1.130
- Exploit Details: Exploit code from Metasploit successfully initiated on honeypot IP 10.0.1.189. A remote shell was
created for the attacker, but the honeypot blocked the execution after 10 seconds. Fake data and misleading responses
were sent to the attacker to simulate a compromised system.
- Action Taken: The shell was blocked to prevent further access, and a fake response was sent to keep the attacker
engaged while logging the activity.

[Alert Level: High]
[Status: Completed | Attack Detected]
```

Figure 5. Log Collection

d) Data Storage and Organization: The collected logs will be securely stored in a database and organized for easy access and analysis. The data will be categorized based on attack type, success, and severity of the intrusion.

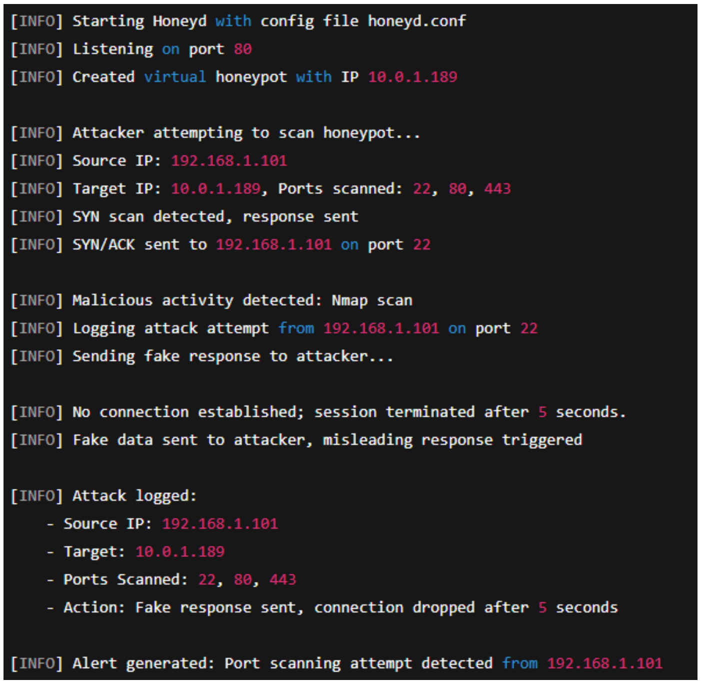
e) AI Log Analysis: Collected logs are subsequently processed and analyzed using Artificial Intelligence. Specifically, this research utilizes the Google Gemini API, a large language model, to interpret the log data, identify patterns indicative of malicious activity, classify the severity of detected threats, and generate actionable recommendations. The Gemini API's capabilities in natural language understanding and generation are leveraged to provide comprehensive insights from the raw log files. Prompts are carefully crafted to guide the API in its analysis and recommendation generation process.



```
[INFO] Analyzing logs for attack attempts detected on honeypot 10.0.1.189...

[INFO] Attack 1: Port Scan Attempt
[Source IP: 192.168.1.101]
[Scan Type: Nmap SYN Scan]
[Ports Scanned: 22, 80, 443]
[Action Taken: Fake SYN/ACK sent, session terminated after 5 seconds]
[Severity: Low]
[AI Recommendation: Continue monitoring. No immediate action required. Log the source IP for future reference.]

[INFO] Attack 2: SSH Brute Force Attempt
[Source IP: 192.168.1.120]
[Username Attempted: root]
[Password Attempts: 5 failed]
[Action Taken: Account locked for 30 minutes, alert triggered]
[Severity: Medium]
[AI Recommendation: Block source IP 192.168.1.120 after 5 failed login attempts. Implement IP blocking mechanism for
brute-force attacks.]

[INFO] Attack 3: Metasploit Exploit Attempt
[Source IP: 192.168.1.130]
[Exploit Module: ms08_067_netapi (Windows SMB RCE)]
[Action Taken: Shell created but blocked after 10 seconds, fake response sent]
[Severity: High]
[AI Recommendation: Immediate action required. Block source IP 192.168.1.130. Alert system administrators and
investigate the ms08_067 vulnerability for patching.]

[INFO] Analysis complete. Recommendations executed:
- Port scan: Logged.
- SSH brute force: IP 192.168.1.120 blocked.
- Metasploit exploit: IP 192.168.1.130 blocked, alert sent.

$ honeypot_analyzer --end
```

Figure 6. AI Analysis

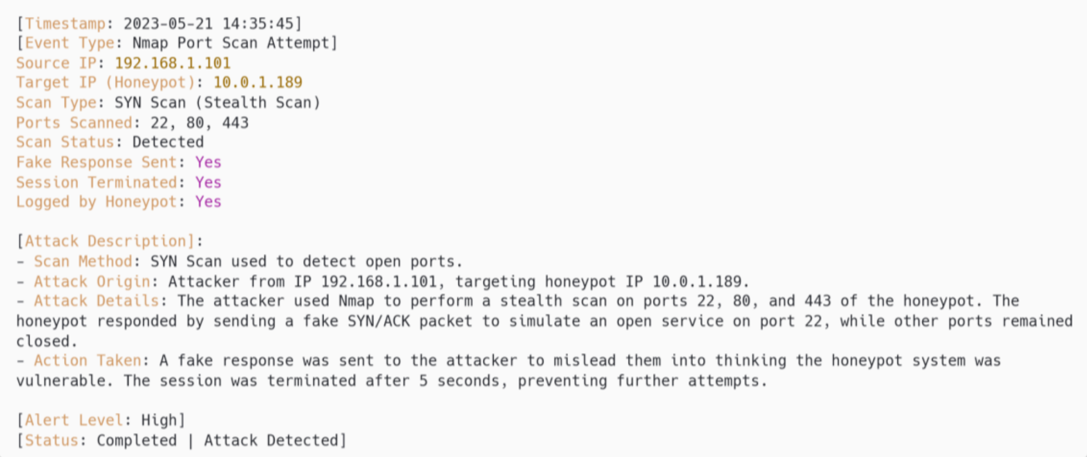
To ensure the validity and reliability of the collected data, the following measures will be taken repeated same simulated attacks will be repeated multiple times under similar conditions to ensure consistency in the results[16]. This repetition helps eliminate any inconsistencies caused by external factors, such as network latency.



```
[INFO] Starting Honeyd with config file honeyd.conf
[INFO] Listening on port 80
[INFO] Created virtual honeypot with IP 10.0.1.189

[INFO] Attacker attempting to scan honeypot...
[INFO] Source IP: 192.168.1.101
[INFO] Target IP: 10.0.1.189, Ports scanned: 22, 80, 443
[INFO] SYN scan detected, response sent
[INFO] SYN/ACK sent to 192.168.1.101 on port 22

[INFO] Malicious activity detected: Nmap scan
[INFO] Logging attack attempt from 192.168.1.101 on port 22
[INFO] Sending fake response to attacker...

[INFO] No connection established; session terminated after 5 seconds.
[INFO] Fake data sent to attacker, misleading response triggered

[INFO] Attack logged:
    - Source IP: 192.168.1.101
    - Target: 10.0.1.189
    - Ports Scanned: 22, 80, 443
    - Action: Fake response sent, connection dropped after 5 seconds

[INFO] Alert generated: Port scanning attempt detected from 192.168.1.101
```

Figure 7. Log for Nmap Scan

Figure 7 represents the *honeypot*'s behavior in response to an Nmap scan and is used to analyze the attack patterns while keeping the primary system secure. Figure 8 shows a log entry from a honeypot system related to a simulated cyberattack using a port scan.



```
[Timestamp: 2023-05-21 14:35:45]
[Event Type: Nmap Port Scan Attempt]
Source IP: 192.168.1.101
Target IP (Honeypot): 10.0.1.189
Scan Type: SYN Scan (Stealth Scan)
Ports Scanned: 22, 80, 443
Scan Status: Detected
Fake Response Sent: Yes
Session Terminated: Yes
Logged by Honeypot: Yes

[Attack Description]:
- Scan Method: SYN Scan used to detect open ports.
- Attack Origin: Attacker from IP 192.168.1.101, targeting honeypot IP 10.0.1.189.
- Attack Details: The attacker used Nmap to perform a stealth scan on ports 22, 80, and 443 of the honeypot. The
honeypot responded by sending a fake SYN/ACK packet to simulate an open service on port 22, while other ports remained
closed.
- Action Taken: A fake response was sent to the attacker to mislead them into thinking the honeypot system was
vulnerable. The session was terminated after 5 seconds, preventing further attempts.

[Alert Level: High]
[Status: Completed | Attack Detected]
```

Figure 8. Log Collection at Nmap Scan Attempt

3. Result and Discussions

Honeypot Messages Over Time, comparing two datasets: Original Data (represented by the red line) and With Offset (represented by the green line). Both datasets exhibit fluctuations over time, reflecting variations in the response times or message timings recorded by the honeypot system during simulated attack attempts or interactions. The original data represents raw, unmodified response times, whereas the offset data has been adjusted, likely to replicate real-world conditions such as network delays or alterations in system behavior[17]. While the two datasets display similar trends, the green line is marginally shifted due to the applied offset, highlighting the potential impact of system adjustments on the honeypot's performance[18]. The consistent fluctuations observed in both datasets indicate that the honeypot is actively responding to different simulated attack patterns or interactions, with the offset providing a mechanism for modeling potential real-world disruptions or variations in system responses[19].

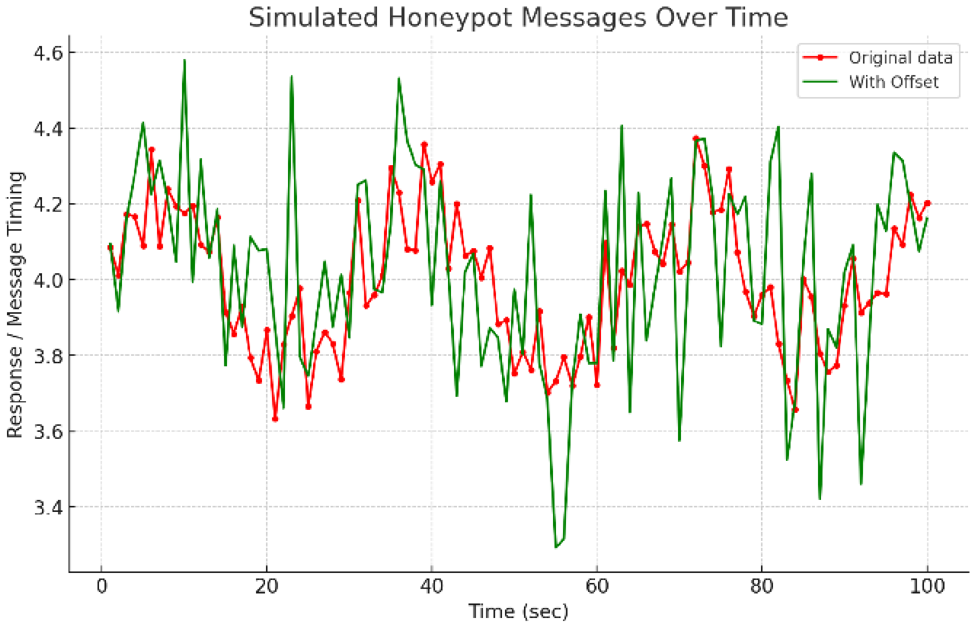


Figure 10. Honeypot Messages Over Time

a) Original Data (Red Line): This likely represents the raw data collected directly from the honeypot system. The red line fluctuates based on the actual response times, messages, or interactions that were logged by the honeypot during simulated attack attempts or normal activity.

b) With Offset (Green Line): This data has been adjusted by adding an "offset." The green line shows the same data but modified, possibly to account for time shifts, added noise, or adjustments that simulate how the system would react under different conditions, or to highlight certain patterns.

Table 1. Summary of Detected Cyberattack Tests on Honeypot

Test	Detected Honeypot	Target	Date
Port scan (Nmap)	Yes	10.0.1.189	4/11/2025 2:35:45 PM
Metasploit	Yes	10.0.1.189	4/11/2025 3:21:11 PM
SSH	Yes	10.0.1.189	4/11/2025 3:29:08 PM

Overall, the AI plays a critical role in analyzing and classifying attack severity while providing appropriate mitigation recommendations. An AI-integrated honeypot system enhances the detection and response effectiveness against threats, This approach offers significant advantages over non-AI honeypot implementations or those using less sophisticated AI. For instance, traditional honeypots might only log events, requiring extensive manual analysis. Basic AI might offer simple pattern matching. In contrast, the Gemini API can provide deeper contextual understanding of logs, more nuanced threat classification, and generate more detailed and actionable mitigation advice[20]. This leads to potential improvements in response time (due to faster, automated analysis), accuracy (due to better understanding of complex attack patterns), and threat classification granularity[21], providing valuable insights for planning future network defense strategies[22].

4. Conclusions and Future Works

This study effectively demonstrates the successful application of a honeypot system, enhanced by log analysis via the Gemini API, for the detection, classification, and provision of mitigation insights regarding simulated cyberattacks. Deployed with the IP address 10.0.1.189, the honeypot proficiently identified diverse attack vectors, including Port Scans, SSH Brute Force attempts, and Metasploit Exploits. The analytical power of the Gemini API proved crucial in dissecting these attacks, categorizing them by severity, and generating actionable mitigation recommendations, thereby underscoring the system's practical value in contemporary cybersecurity.

Building upon these strong and actionable findings, future development will focus on further refining and expanding the system's capabilities, reflecting modern cybersecurity needs. Immediate, short-term (next 6-12 months) priorities include the meticulous refinement and optimization of prompts utilized with the Gemini API to achieve even greater accuracy and contextual awareness in log analysis and recommendation generation. Concurrently, more extensive testing with a broader spectrum of simulated attacks will be undertaken to rigorously validate the system's robustness and the AI's classification precision, ensuring its readiness for more complex threat landscapes.

Strategically, medium-term goals (1-2 years) will explore the integration of real-time or near real-time log streaming to the Gemini API, significantly enhancing incident response times, and will also initiate the development of dynamic honeypot behavior, allowing the system to adapt its emulated services based on detected attack patterns. Looking further ahead, long-term development (2+ years) will investigate the integration of more advanced AI algorithms, such as deep learning through fine-tuning large language models (like those accessible via Gemini API) or exploring reinforcement learning for adaptive defense against sophisticated threats like Advanced Persistent Threats (APTs). Furthermore, a key long-term objective is the creation of a collaborative honeypot network for sharing anonymized attack data and AI-driven insights globally, alongside improving automated mitigation for zero-day vulnerabilities through integration with global threat intelligence. These comprehensive advancements aim to substantially strengthen the honeypot's capacity for providing robust, real-time defense against the continuously evolving cyber threat panorama.

5. References

[1] J. Williams, M. Edwards, and J. Gardiner, "Time-to-Lie: Identifying Industrial Control System Honeypots Using the Internet Control Message Protocol," 2024. [Online]. Available: https://arxiv.org/abs/2410.17731

[2] A. Albaseer, N. Abdi, M. Abdallah, M. Qaraqe, and S. Alkuwari, "FedPot: A Quality-Aware Collaborative and Incentivized Honeypot-Based Detector for Smart Grid Networks," 2024. [Online]. Available: https://arxiv.org/abs/2407.02845

[3] N. Kaur and L. Gupta, "Explainable AI for Securing Healthcare in IoT-Integrated 6G Wireless Networks," 2025. [Online]. Available: https://arxiv.org/abs/2505.14659

[4] P. B. Lopez, P. Nespoli, and M. G. Perez, "Cyber Deception Reactive: TCP Stealth Redirection to On-Demand Honeypots," 2024. [Online]. Available: https://arxiv.org/abs/2402.09191

[5] M. Kahlhofer and S. Rass, "Application Layer Cyber Deception Without Developer Interaction," in *2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&amp;PW)*, IEEE, Jul. 2024, pp. 416–429. doi: 10.1109/EuroSPW61312.2024.00053.

[6] Y. L. Aung *et al.*, "HoneyWin: High-Interaction Windows Honeypot in Enterprise Environment," 2025. [Online]. Available: https://arxiv.org/abs/2505.00465

[7] U. Ubaidillah, T. Taryo, and A. Hindasyah, "Analisis dan Implementasi Honeypot Honeyd Sebagai Low Interaction Terhadap Serangan Distributed Denial Of Service (DDOS) dan Malware," *JTIM : Jurnal Teknologi Informasi dan Multimedia*, vol. 5, no. 3, pp. 208–217, Oct. 2023, doi: 10.35746/jtim.v5i3.405.

[8] Y. Wang, Z. Su, A. Benslimane, Q. Xu, M. Dai, and R. Li, "Collaborative Honeypot Defense in UAV Networks: A Learning-Based Game Approach," 2023. [Online]. Available: https://arxiv.org/abs/2211.01772

[9] A. Said, "On explaining recommendations with Large Language Models: a review," *Front Big Data*, vol. 7, Jan. 2025, doi: 10.3389/fdata.2024.1505284.

[10] Y. Wang, T. Gu, Y. Teng, Y. Wang, and X. Ma, "HoneypotNet: Backdoor Attacks Against Model Extraction," 2025. [Online]. Available: https://arxiv.org/abs/2501.01090

[11] M. Nawrocki, J. Kristoff, R. Hiesgen, C. Kanich, T. C. Schmidt, and M. Wählisch, "SoK: A Data-driven View on Methods to Detect Reflective Amplification DDoS Attacks Using Honeypots," in *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, IEEE, Jul. 2023, pp. 576–591. doi: 10.1109/EuroSP57164.2023.00041.

[12] J. Landsborough, N. C. Rowe, T. D. Nguyen, and S. Fugate, "WiP: Deception-in-Depth Using Multiple Layers of Deception," 2024. [Online]. Available: https://arxiv.org/abs/2412.16430

[13] A. Javadpour, F. Ja'fari, T. Taleb, M. Shojafar, and C. Benzaïd, "A comprehensive survey on cyber deception techniques to improve honeypot performance," *Comput Secur*, vol. 140, p. 103792, May 2024, doi: 10.1016/j.cose.2024.103792.

[14] K. Highnam, Z. Hanif, E. Van Vogt, S. Parbhoo, S. Maffeis, and N. R. Jennings, "Adaptive Experimental Design for Intrusion Data Collection," 2023. [Online]. Available: https://arxiv.org/abs/2310.13224

[15] L. Sousa, J. Cecílio, P. Ferreira, and A. Oliveira, "Reconfigurable and Scalable Honeynet for Cyber-Physical Systems," 2024. [Online]. Available: https://arxiv.org/abs/2404.04385

[16] Y. Otoum, A. Asad, and A. Nayak, "Blockchain Meets Adaptive Honeypots: A Trust-Aware Approach to Next-Gen IoT Security," 2025. [Online]. Available: https://arxiv.org/abs/2504.16226

[17] Z. Peng, Y. He, J. Ni, and B. Niu, "Bypassing DARCY Defense: Indistinguishable Universal Adversarial Triggers," 2024. [Online]. Available: https://arxiv.org/abs/2409.03183

[18] H. Q. Ngo, M. Guo, and H. Nguyen, "Catch Me if You Can: Effective Honeypot Placement in Dynamic AD Attack Graphs," 2023. [Online]. Available: https://arxiv.org/abs/2312.16820

[19] Irfan Murti Raazi, Ima Dwitawati, and Putri Nabila, "Uji Vulnerability Assessment Dalam Mengetahui Tingkat Keamanan Web Aplikasi Sistem Informasi Laporan Diskominfo Dan Sandi Aceh," *J-INTECH: Journal Of Information Technology*, vol. 4, no. 1, pp. 1–15, Feb. 2023, doi: 10.22373/j-intech.v4i1.2409.

[20] A. Ebunoluwa and A. James, "AI-Powered Honeypots: Enhancing Deception Technologies for Cyber Defense," Jun. 2025.

[21] M. B. -, "AI-enhanced Honeypots for Zero-Day Exploit Detection and Mitigation," *International Journal For Multidisciplinary Research*, vol. 6, no. 6, Dec. 2024, doi: 10.36948/ijfmr.2024.v06i06.32866.

[22] Z. Zhang *et al.*, "Soft Thinking: Unlocking the Reasoning Potential of LLMs in Continuous Concept Space," 2025. [Online]. Available: https://arxiv.org/abs/2505.15778