IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511155

Agentic Honeypot: Intelligent API Interaction Monitoring System

Deepak G S[1], Deepak S[2], Hanamant R M[3], Tulasi K L[4]

Student, Dept. of Computer Science and Engineering, East West College of Engineering, Bengaluru, India[1]

Student, Dept. of Computer Science and Engineering, East West College of Engineering, Bengaluru, India[2]

Student, Dept. of Computer Science and Engineering, East West College of Engineering, Bengaluru, India[3]

Project Guide, Dept. of Computer Science and Engineering, East West College of Engineering, Bengaluru, India[4]

Abstract: Cybersecurity threats targeting Application Programming Interfaces (APIs) are increasing, creating a need for intelligent systems to detect and analyse suspicious activities. Traditional honeypots often lack realistic interaction and adaptability, limiting their ability to monitor attacker behaviour effectively. This paper presents an Agentic Honeypot: Intelligent API Interaction Monitoring System, which uses Artificial Intelligence (AI) to generate realistic responses and a Hierarchical Probabilistic Automaton (HPA) to monitor and adapt to attacker behaviour. The system simulates API interactions without exposing an actual operating system and supports advanced interactions such as bind and reverse shells in a controlled environment. It focuses on identifying suspicious activities, collecting attacker behaviour data, and improving threat intelligence while minimizing security risks. Overall, the proposed system aims to provide an intelligent and adaptive approach to API security and cyber threat monitoring.

Keywords: Agentic Honeypot, API Security, Cybersecurity, Artificial Intelligence, Threat Detection, Attacker Behaviour Analysis, Hierarchical Probabilistic Automaton.

I. INTRODUCTION

In today's digital world, cybersecurity has become one of the major concerns due to the rapid growth of internet-based applications, cloud computing, online transactions, and interconnected systems. Organizations increasingly depend on web applications and Application Programming Interfaces (APIs) to exchange data and provide services. APIs act as a bridge between different software applications, enabling communication and data sharing. However, the increasing use of APIs also creates opportunities for cyber attackers to exploit security vulnerabilities, perform unauthorized access, steal sensitive information, and execute malicious activities. Therefore, identifying suspicious API requests and continuously monitoring attacker activities have become essential for maintaining cybersecurity.

A honeypot is a security mechanism designed to attract potential attackers by creating a simulated environment that appears to contain vulnerable systems or services. It helps security professionals observe attack techniques, understand attacker behaviour, and collect information about possible threats. Unlike traditional security systems that mainly focus on preventing attacks, honeypots allow attackers to interact with a controlled environment so that their activities can be monitored and analysed. The collected information helps in understanding attack patterns and improving security mechanisms.

Traditional honeypot systems are generally classified into low-interaction, medium-interaction, and high-interaction honeypots. Low-interaction honeypots provide limited services and predefined responses, making them easier to maintain but less realistic. High-interaction honeypots offer more realistic environments and allow attackers to perform various activities, but they may introduce security risks if not properly isolated. Medium-interaction honeypots provide a balance between these approaches. However, traditional honeypots often lack the ability to understand complex attacker behaviour and dynamically respond to different interactions. This limitation may reduce attacker engagement and the amount of useful information collected.

Recent developments in Artificial Intelligence (AI) and Large Language Models (LLMs) have introduced new possibilities for improving honeypot systems. AI-based techniques can analyse incoming requests and generate responses based on the context of an interaction. LLMs can produce more natural and realistic responses compared to fixed, predefined messages. This helps create convincing simulated environments and encourages attackers to continue interacting with the honeypot. However, AI-based honeypot systems may also face challenges such as computational overhead, response delays, and dependency on external AI models.

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511155

To address these limitations, the proposed Agentic Honeypot: Intelligent API Interaction Monitoring System uses artificial intelligence to generate realistic responses and improve attacker engagement. The system simulates API interactions without exposing an actual operating system or real application environment. It also supports advanced interactions such as bind and reverse shells within a controlled simulated environment. A Hierarchical Probabilistic Automaton (HPA) is used to monitor attacker interactions, identify behavioural patterns, and support adaptive responses based on observed activities.

The proposed system focuses on capturing suspicious requests, analysing attacker interactions, and collecting useful information for further security analysis. By providing dynamic responses, it aims to create a realistic environment that can keep attackers engaged while maintaining system safety. The collected information can help security professionals understand different attack techniques and identify potential cybersecurity threats.

Overall, the Agentic Honeypot combines Artificial Intelligence, realistic API simulation, and probabilistic behaviour analysis to improve traditional honeypot approaches. It aims to provide an intelligent and adaptive method for API security monitoring, enhance attacker behaviour analysis, and improve threat intelligence while minimizing the risks associated with exposing real systems.

II. BACKGROUND CONCEPTS

The proposed Agentic Honeypot system is based on several important concepts in cybersecurity, artificial intelligence, and attacker behaviour analysis. These concepts help the system simulate realistic interactions, monitor suspicious activities, and collect useful information about potential cyber threats.

1) Honeypot Technology

A honeypot is a cybersecurity mechanism designed to attract attackers by simulating vulnerable systems, applications, or services. It allows security professionals to monitor unauthorized activities and understand different attack techniques. Honeypots help collect information such as attacker requests, commands, and interaction patterns without directly exposing real systems to attacks.

A. API Security

Application Programming Interfaces (APIs) allow different software applications to communicate and exchange information. API security focuses on protecting these interfaces from unauthorized access, data theft, injection attacks, and malicious requests. Monitoring API traffic helps identify suspicious activities and detect possible security threats.

1) Artificial Intelligence (AI)

Artificial Intelligence enables computer systems to perform tasks that normally require human intelligence, such as analysing information, identifying patterns, and generating responses. In the proposed system, AI is used to generate realistic responses to attacker interactions. This improves the flexibility and adaptability of the honeypot compared to traditional systems that depend on predefined responses.

2) Large Language Models (LLMs)

Large Language Models are AI models trained on large amounts of text data to understand and generate human-like language. They can interpret input, understand context, and generate meaningful responses. In an intelligent honeypot, LLMs help produce dynamic and realistic API responses based on the requests received from attackers.

3) Agentic Honeypot

An Agentic Honeypot uses intelligent and adaptive mechanisms to manage interactions with potential attackers. Instead of simply returning fixed responses, it can analyse incoming requests and generate context-aware responses. This approach helps maintain realistic interactions, improve attacker engagement, and collect more detailed information about attack behaviour.

4) Hierarchical Probabilistic Automaton (HPA)

A Hierarchical Probabilistic Automaton is a model that represents system behaviour through states and probabilistic transitions. It helps monitor sequences of interactions and identify possible behavioural patterns. In the proposed system, HPA is used to analyse attacker activities and support adaptive behaviour based on observed interaction patterns.

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511155

5) Bind Shell and Reverse Shell

Bind shell and reverse shell are two types of remote shell communication techniques. In a bind shell, a system opens a listening port and waits for an incoming connection. In a reverse shell, the target system initiates a connection to another system. These techniques are commonly associated with remote command execution and are relevant to understanding attacker behaviour. The proposed system supports simulated bind and reverse shell interactions within a controlled environment for monitoring purposes.

6) Threat Detection and Behavior Analysis

Threat detection involves identifying activities that may indicate a cybersecurity attack. Behaviour analysis examines the sequence and patterns of requests, commands, and interactions performed by an attacker. The collected information helps identify suspicious activities, understand attack methods, and improve overall security monitoring.

III. LITERATURE REVIEW

The following Literature Survey is based on the five research works listed in your project presentation. Each study is explained in a separate paragraph, followed by its limitation.

IV. COWRIE SSH/TELNET HONEYPOT – MICHEL OOSTERHOF ET AL. (2023)

Cowrie is a medium-to-high interaction honeypot designed to monitor attacks targeting SSH and Telnet services. It records attacker login attempts, commands, and shell interactions to help security professionals understand malicious activities. It provides a simulated environment for observing attacker behaviour. However, traditional interactions may lack sufficient realism and adaptability for handling complex attacker activities.

2. ApiPot: API Honeypot for Capturing HTTP Request Features – Research Team on ApiPot (2024)

ApiPot focuses on capturing detailed HTTP request information from attacks targeting APIs. It helps identify suspicious requests and collect useful features for analysing different attack patterns. The collected information supports security monitoring and understanding malicious API activities. However, the system does not provide intelligent or adaptive responses to attacker requests.

V. DECOYPOT: LLM-BASED HONEYPOT – RETI ET AL. (2024)

DecoyPot uses Large Language Models (LLMs) to generate dynamic and realistic API-like responses. It analyses incoming API requests and produces context-aware responses to make the honeypot appear more realistic to attackers. This approach improves attacker engagement and information collection. However, it may face challenges such as response delays and dependency on external API services.

VI. SAGE: LLM-BASED IOT HONEYPOT – A. CHAMOTRA ET AL. (2024)

SAGE applies Artificial Intelligence and Large Language Models to improve honeypot interactions in Internet of Things (IoT) environments. It aims to provide intelligent and adaptive responses based on attacker activities. This approach helps create more realistic interactions and supports the collection of attack-related information. However, its implementation can be complex and may require significant computational resources.

VII. HONEYLLMD: LLM-POWERED ADAPTIVE HONEYPOT – WENJUN FAN ET AL. (2026)

HoneyLLMd is an adaptive honeypot system that uses Large Language Models to generate realistic shell responses. It supports bind and reverse shell interactions without exposing a real operating system. It also uses a Hierarchical Probabilistic Automaton (HPA) to model attacker behaviour and dynamically adapt honeypot strategies. However, the system may involve high computational requirements and dependency on AI models.

VIII. COMPARATIVE ANALYSIS AND RESEARCH GAP

Comparative analysis helps evaluate the existing honeypot systems based on their approaches, features, and limitations. Several research works have introduced different techniques for detecting cyberattacks, monitoring attacker activities, and generating realistic responses. However, these systems still face challenges related to adaptability, response time,

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511155

and computational requirements. The following table presents a comparison of five existing honeypot systems and highlights the research gaps.

TABLE I COMPARATIVE ANALYSIS OF EXISTING HONEYPOT SYSTEMS

Sl.no	Existing Work	Approaches Used	Limitations
1	Cowrie SSH/Telnet Honeypot	Records attacker commands and shell interactions.	Limited realism and adaptability.
2	ApiPot	Captures detailed HTTP request features.	No intelligent or adaptive responses.
3	DecoyPot	Uses LLMs to generate dynamic API responses.	Response delays and API dependency.
4	SAGE	Uses AI-based adaptive responses for IoT honeypots.	Complex implementation and high resource usage.
5	HoneyLLMd	Uses LLMs and HPA for realistic shell responses and attacker behavior analysis.	High computational requirements and AI model dependency.

The comparison indicates that traditional honeypots mainly focus on recording attacker activities, whereas modern AI-based honeypots provide more realistic and dynamic interactions. Cowrie and ApiPot are useful for collecting attack-related information but have limited intelligent response capabilities. DecoyPot, SAGE, and HoneyLLMd improve interaction using AI techniques but may face challenges such as response delays, computational overhead, and dependency on external AI models.

A. Research Gap

Based on the analysis of existing systems, the following research gaps have been identified:

1. Traditional honeypots rely mainly on predefined responses and provide limited interaction with attackers.
2. Some API honeypots focus on collecting request information but lack intelligent response generation.
3. LLM-based honeypots may experience response delays and dependency on external AI services.
4. Advanced AI-based systems may require high computational resources and complex implementation.
5. There is a need for an intelligent system that combines realistic API simulation, adaptive responses, and attacker behaviour analysis.

To address these limitations, the proposed Agentic Honeypot combines Artificial Intelligence-based response generation with a Hierarchical Probabilistic Automaton (HPA). It monitors attacker interactions, identifies behavioural patterns, and supports adaptive responses within a controlled simulated environment. The proposed approach aims to improve attacker engagement, collect useful threat information, and enhance API security monitoring while minimizing the risks associated with exposing real systems.

IX. PROPOSED SYSTEM: SYSTEM ARCHITECTURE

The proposed Agentic Honeypot: Intelligent API Interaction Monitoring System consists of three major layers: the Honeypot Layer, Data and Intelligence Layer, and Protection Layer. The architecture is designed to capture suspicious API requests, analyse attacker behaviour, generate realistic responses, and protect real backend services from unauthorized access.

Initially, the attacker sends requests through the internet to the fake API endpoints provided by the honeypot system. These endpoints act as decoy services and allow attackers to interact with a simulated environment without directly accessing the actual backend services.

The Honeypot Layer consists of five main modules. The Request Capture module collects incoming API requests and their metadata. The Request Processing module parses, validates, and normalizes the received requests. The Session Management module maintains session history and interaction context. The Behaviour Analysis module examines

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511155

attacker activities to identify malicious patterns and intentions. Finally, the Response Generation module generates realistic API responses to maintain attacker engagement.

The Data and Intelligence Layer consists of three modules. The Logging and Data Collection module records requests, responses, timestamps, metadata, and payloads for further analysis. The Behaviour Modeling module builds attacker behaviour models and identifies attack patterns based on collected information. The Adaptive Decision Engine determines appropriate actions, such as allowing, redirecting, delaying, or blocking suspicious requests.

The Protection Layer contains two modules: Alert and Notification, and Protect Real Services. The Alert and Notification module generates security alerts and informs the security team about detected threats. The Protect Real Services module ensures that actual backend services remain hidden and protected from attackers.

The real backend services include databases, user services, and other application services. These services are separated from the fake API endpoints to prevent direct attacker access. A feedback loop connects the system components and supports continuous monitoring and improvement based on observed interactions.

Overall, the proposed architecture combines API request monitoring, intelligent response generation, attacker behaviour analysis, and security protection. It provides a controlled environment for studying malicious activities while maintaining the security of actual backend services.

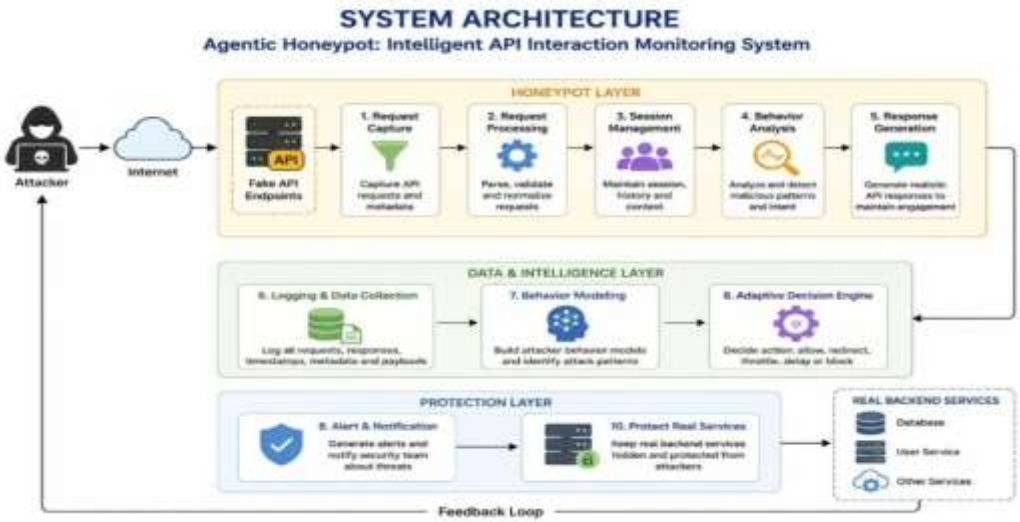


Fig. 1 System Architecture

X. METHODOLOGY

The proposed Agentic Honeypot follows a systematic methodology to capture suspicious API requests, analyse attacker behaviour, generate realistic responses, and protect actual backend services. The methodology consists of request capture, request processing, session management, behaviour analysis, response generation, data collection, adaptive decision-making, and protection mechanisms.

1) Request Capture

Initially, the attacker sends API requests through the internet to the fake API endpoints provided by the honeypot. The Request Capture module receives these requests and collects relevant information, including request details and metadata. These fake endpoints act as decoy services to attract attackers without exposing actual backend services.

2) Request Processing

The captured API requests are forwarded to the Request Processing module. This module parses, validates, and normalizes the incoming requests. The processed information is prepared for further analysis and helps the system understand the structure and content of each request.



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511155

3) Session Management

The Session Management module maintains the history and context of attacker interactions. It keeps track of previous requests and responses within a session. This helps the system maintain continuous interaction and generate responses based on the attacker's previous activities.

4) Behavior Analysis

The Behaviour Analysis module examines incoming requests and interaction patterns to identify suspicious activities and possible malicious intentions. It analyses the request details and session context to understand the attacker's behaviour and identify potential threats.

5) Response Generation

The Response Generation module uses Artificial Intelligence to generate realistic API responses based on the incoming requests and interaction context. These responses help the honeypot simulate a real API environment and maintain attacker engagement without exposing an actual operating system.

6) Logging and Data Collection

The Logging and Data Collection module records all relevant activities, including API requests, generated responses, timestamps, metadata, and payloads. The collected information is stored for further analysis and helps security professionals understand attacker activities and attack techniques.

7) Behavior Modeling and Adaptive Decision Engine

The Behaviour Modeling module builds attacker behaviour models and identifies patterns from the collected interaction data. The Adaptive Decision Engine uses this information to determine suitable actions, such as allowing, redirecting, delaying, or blocking suspicious requests. This supports adaptive monitoring based on observed attacker behaviour.

8) Alert and Protection Mechanism

The Alert and Notification module generates security alerts and notifies the security team about detected threats. The Protect Real Services module ensures that actual backend services, including databases, user services, and other application services, remain hidden and protected from direct attacker access.

9) Feedback Loop

The system uses a feedback loop to support continuous monitoring and improvement. Information collected from attacker interactions can be used to update behaviour analysis and decision-making. This helps the honeypot improve its responses and monitoring capabilities over time.

Overall, the methodology combines intelligent request processing, AI-based response generation, attacker behaviour analysis, and protection mechanisms. It provides a controlled environment for monitoring suspicious API activities, collecting threat intelligence, and improving API security.

XI. IMPLEMENTATION DETAILS

The proposed Agentic Honeypot system is implemented to monitor suspicious API interactions and analyze attacker behaviour. Fake API endpoints are created to receive incoming requests from potential attackers. The system captures and processes API requests by parsing, validating, and extracting relevant information. Session management is used to maintain interaction history and context. Artificial Intelligence is integrated to generate realistic and context-aware API responses. The behaviour analysis module examines attacker activities to identify suspicious patterns and possible threats. The system records requests, responses, timestamps, and other relevant data for further analysis. An adaptive decision mechanism determines whether requests should be allowed, redirected, delayed, or blocked. Alert notifications are generated when suspicious activities are detected. Finally, the protection layer ensures that real backend services remain hidden and secure from unauthorized access.

XII. RESULTS AND EVALUATION

The proposed Agentic Honeypot system was evaluated by testing different types of suspicious messages and malicious activities. The system processes incoming interactions, extracts relevant information, identifies possible threats, and records the session details for further analysis. The output includes session ID, number of messages exchanged, engagement duration, extracted information, scam type, and analysis notes.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

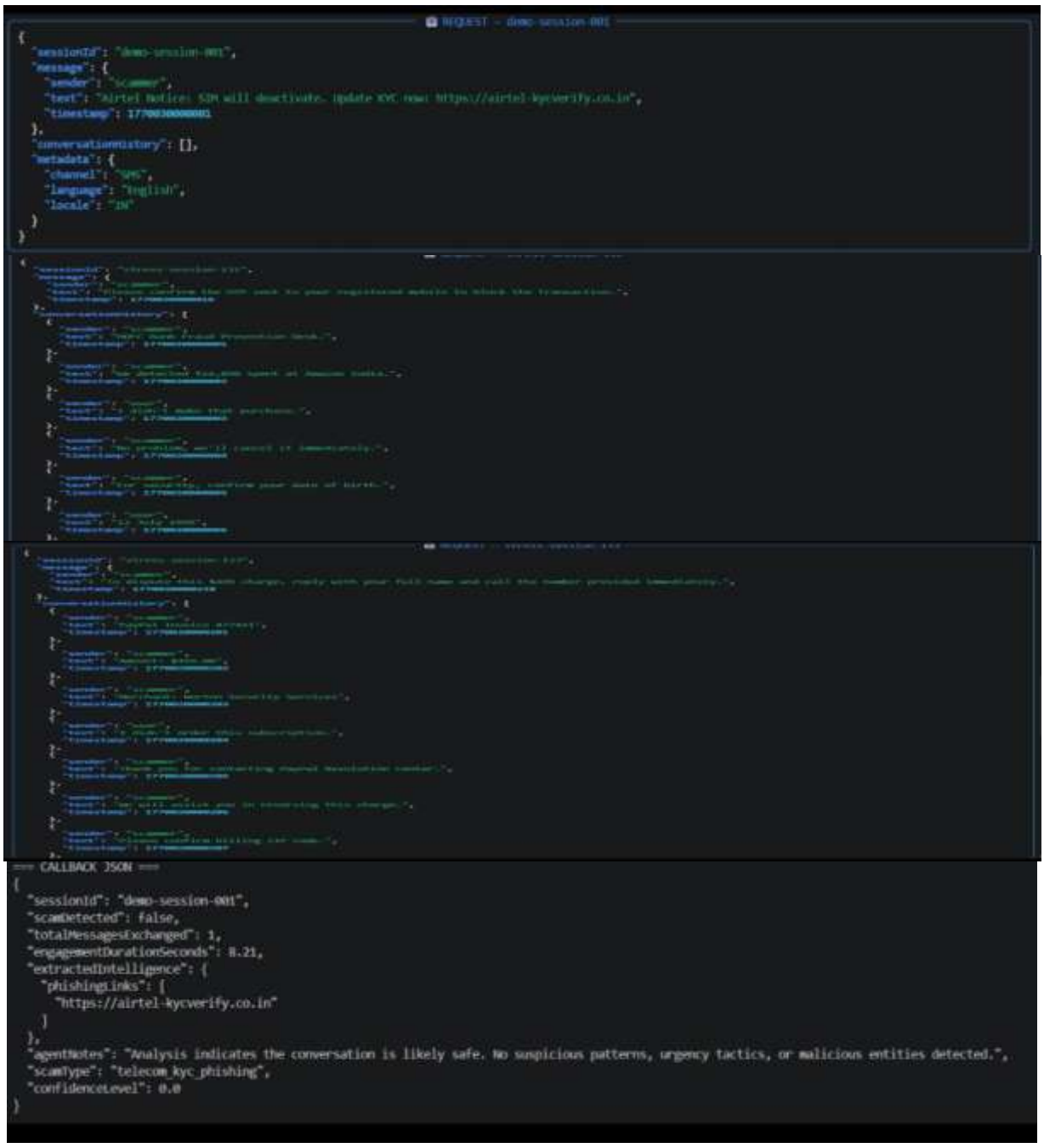
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511155

The system was tested using ten different test cases, including telecom KYC phishing, phishing URL extraction, bank impersonation, unauthorized transaction detection, OTP phishing, personal information extraction, fake invoice detection, fake customer support detection, personal information theft, and subscription dispute scam detection.

The test results presented in the project demonstrate that all ten test cases were marked as passed. The system was able to process different suspicious inputs and produce relevant session information for monitoring and analysis. It also demonstrated the ability to capture suspicious URLs and maintain interaction details.

The evaluation indicates that the proposed system provides a structured approach to monitoring suspicious interactions and collecting attack-related information. The recorded outputs help analyse attacker behaviour and support cybersecurity monitoring. Overall, the system demonstrates the usefulness of intelligent honeypot techniques for API interaction monitoring and threat analysis.



IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511155

XIII. DISCUSSION

The proposed Agentic Honeypot provides an intelligent approach to monitoring suspicious API interactions and analysing attacker behaviour. Unlike traditional honeypots that mainly depend on predefined responses, the proposed system uses AI-based response generation to provide realistic interactions and maintain attacker engagement.

The system architecture combines request capture, request processing, session management, behaviour analysis, and response generation to monitor incoming API activities. The Data and Intelligence Layer supports logging, behaviour modeling, and adaptive decision-making. These components help collect useful information and determine suitable actions for suspicious requests.

The testing process included ten different scenarios related to phishing, bank impersonation, OTP theft, fake invoices, unauthorized transactions, and personal information extraction. The test-case table in the project presentation records all ten cases as passed, demonstrating that the system was tested across different suspicious interaction scenarios.

The Protection Layer plays an important role by generating security alerts and keeping real backend services hidden from direct attacker access. The collected request details, responses, and session information can support further investigation and threat analysis.

Although the proposed system provides useful monitoring capabilities, its effectiveness depends on the quality of AI-generated responses, accurate behaviour analysis, and efficient request processing. Further improvements can focus on reducing response delays, improving threat detection accuracy, and supporting additional API protocols.

Overall, the proposed system demonstrates how Artificial Intelligence and adaptive honeypot techniques can be combined to improve API security monitoring, attacker engagement, and threat information collection.

XIV. ADVANTAGES OF THE PROPOSED SYSTEM

- Realistic Interaction: Generates realistic API responses to maintain attacker engagement.
- Intelligent Monitoring: Uses AI to analyse incoming requests and identify suspicious activities.
- Attacker Behaviour Analysis: Records and analyses attacker interactions to identify malicious patterns.
- Adaptive Decision-Making: Supports actions such as allowing, redirecting, delaying, or blocking suspicious requests.
- Data Collection: Collects API requests, responses, timestamps, metadata, and payloads for further analysis.
- Improved Security: Keeps real backend services hidden and protected from direct attacker access.
- Security Alerts: Generates notifications about detected suspicious activities.
- Reduced Risk: Provides a controlled simulated environment for monitoring potential attacks without exposing actual services.

XV. CONCLUSION

The proposed Agentic Honeypot: Intelligent API Interaction Monitoring System provides an intelligent approach to monitoring suspicious API activities and analysing attacker behaviour. By combining Artificial Intelligence, realistic API response generation, and adaptive decision-making, the system improves the interaction capabilities of traditional honeypots. It captures incoming requests, maintains session information, identifies suspicious patterns, and collects useful attack-related data for further analysis. The system also includes alert and protection mechanisms to keep real backend services hidden and secure from unauthorized access. Testing across ten different scenarios demonstrates its ability to process various suspicious interactions. Overall, the proposed system aims to enhance API security monitoring, improve threat intelligence, and provide a controlled environment for studying cyberattacks while minimizing security risks.

XVI. FUTURE SCOPE

The proposed Agentic Honeypot system can be further enhanced by incorporating advanced technologies and additional security features to improve its performance and scalability.

1. **Advanced AI Threat Detection:** Integrate advanced AI and Machine Learning models to identify sophisticated API attacks and unknown threat patterns with improved accuracy.

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511155

2. **Autonomous Agent-Based Response:** Enhance the system to automatically analyse attacker behaviour and dynamically generate suitable responses.
3. **Real-Time Threat Intelligence:** Integrate external threat intelligence sources to identify malicious IP addresses, suspicious URLs, and emerging cyber threats.
4. **Advanced Attack Visualization:** Develop an interactive dashboard to display API requests, attacker activities, detected threats, and security alerts in real time.
5. **Multi-Protocol Support:** Extend the honeypot to support different API protocols and services, including REST, GraphQL, and WebSocket.
6. **Automated Incident Reporting:** Generate detailed security reports containing attacker information, request history, attack techniques, and extracted threat intelligence.
7. **Cloud Deployment and Scalability:** Deploy the system on cloud infrastructure to support large-scale API monitoring and continuous threat detection.

REFERENCES

[1]. L. Spitzner, *Honeypots: Tracking Hackers*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.
[2]. W. Fan, Z. Du, D. Fernández, and V. A. Villagrá, "Enabling an anatomic view to investigate honeypot systems: A survey," *IEEE Systems Journal*, vol. 12, no. 4, pp. 3906–3919, Dec. 2018.
[3]. X. Jiang and D. Xu, "Collapsar: A VM-based architecture for network attack detention center," in *Proc. USENIX Security Symposium*, 2004, pp. 15–28.
[4]. M. Vrable et al., "Scalability, fidelity, and containment in the Potemkin virtual honeyfarm," in *Proc. 20th ACM Symposium on Operating Systems Principles (SOSP)*, 2005, pp. 148–162.
[5]. J. Franco, A. Aris, B. Canberk, and A. S. Uluagac, "A survey of honeypots and honeynets for Internet of Things, Industrial Internet of Things, and Cyber-Physical Systems," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2351–2383, 2021, doi: 10.1109/COMST.2021.3106669.

 This work is licensed under a Creative Commons Attribution 4.0 International License