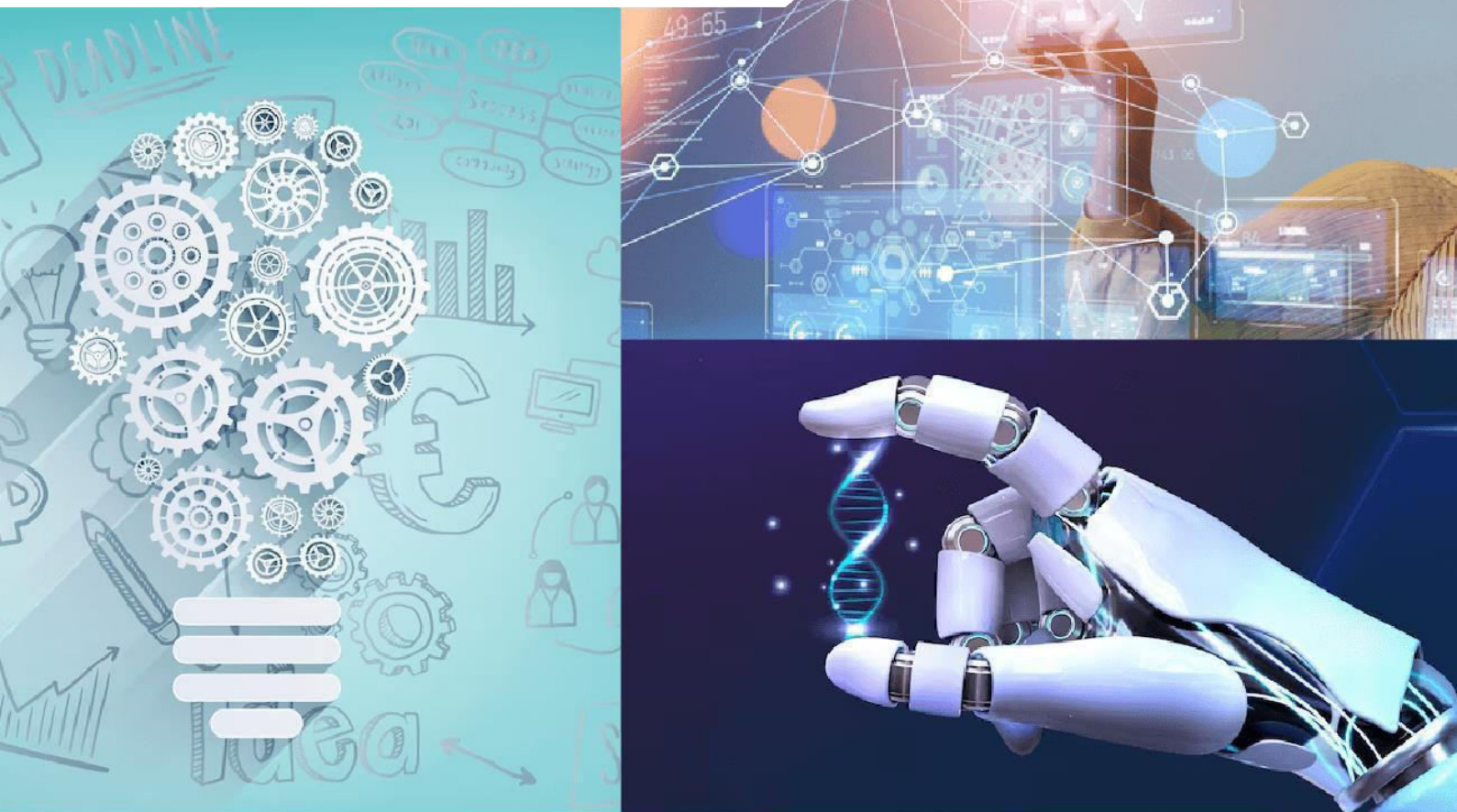




International Journal of Advanced Research in Education and Technology (IJARETY)

Volume 13, Issue 3, May - June 2026

Impact Factor: 9.971



Attack Detection and Prevention using Honeypot Network

M. Ajaykumar ¹, K. Shiva ², K. Shiva Rama Krishna³, M. Praneeth⁴

Assistant Professor, Department of CSE, Guru Nanak Institute of Technology, Hyderabad, Telangana, India ¹

Students, Department of CSE, Guru Nanak Institute of Technology, Hyderabad, Telangana, India

ABSTRACT: With the rapid increase in the number of users, there is a rise in issues related to hardware failure, web hosting, space and memory allocation of data, which is directly or indirectly leading to the loss of data. With the objective of providing services that are reliable, fast and low in cost, we turn to cloud-computing practices. With a tremendous development in this technology, there is ever increasing chance of its security being compromised by malicious users. A way to divert malicious traffic away from systems is by using Honeypot. A colossal strategy has shown signs of improvement in security of systems. Keeping in mind the various legal issues one may face while deploying Honeypot on third-party cloud vendor servers, the concept of Honeypot is implemented in a file-sharing application, which is deployed on cloud server. This project discusses the detection attacks in a cloud-based environment as well as the use of Honeypot for its security, thereby proposing a new technique to do the same.

I. INTRODUCTION

Cloud computing is a technique to store, share and access data anytime and anywhere with a device that is connected to a network, preferably the internet. Cloud computing consists of an expandable storage space with no physical storage space which is accessible from anywhere in the world using any device, by connecting it to the internet. It contains large number of computing devices connected through a real-time communication and has a common data storage area. Honeypots are viewed as a successful technique to track programmer conduct and uplift the viability of security instruments. Honeypots are specifically designed to not only purposely engage and deceive hackers but also identify malicious activities performed over the Internet and can be counted as an effective method to track hacker behaviour. Honeypots can be defined as systems or assets which are used to not only trap, monitor but to also identify erroneous requests present within a network. They vary in the interaction provided to the attackers, from low interaction to medium and high, each type has its advantages and disadvantages.

Their aim is to analyze, understand, watch and track attacker's behaviour in order to create systems that are not only secure but can also handle such traffic. It is a closely monitored computing resource that we want to be probed, attacked, or compromised. More precisely, it is an information system resource whose value lies in unauthorized or illicit use of that resource.

1.1 SCOPE OF THE PROJECT

Cloud-based Honeypots give the capacity to explore and examine assaults that hit ordinary customers. Having them permits a specialist to interpret the IP locations and malware being utilized into security content that can ensure a normal cloud environment. Once those IP addresses have been distinguished, they will then lead a ping scope and defencelessness output to discover a shortcoming in the system outline or vulnerabilities in programming that can be misused. The proposed system scope includes:

- Automated attack detection and monitoring using honeypot decoys in a cloud-based file-sharing environment.
- Multi-format support for user authentication, file upload, download, and network management through a web-based interface.
- Real-time monitoring and threat intelligence gathering from attacker behaviour patterns.
- Cost-effective security solution built on open-source technologies (J2EE, MySQL) deployable on standard cloud infrastructure.
- Integration with existing network management systems and hospital or enterprise cloud environments.
- Patient and user data privacy protection ensuring no sensitive information is exposed to unauthorized entities.
- Extensible architecture supporting future AI/ML-based threat analysis and automated response mechanisms.

II. LITERATURE SURVEY

1. Design of Privacy-Preserving Cloud Storage Framework (2025) – RuWei Huang et al.

This study proposes a privacy-preserving cloud storage framework that includes the design of data organization structure, the generation and management of keys, the treatment of changes in users' access rights, dynamic operations of data, and the interaction between participants. The framework mainly focuses on improving data confidentiality, secure access control, and efficient key management in cloud environments, enhancing scalability and supporting secure data sharing among multiple users without affecting system performance.

2. Scientific Cloud Computing: Early Definition and Experience (2024) – Lizhe Wang et al.

This research reviews recent advances in cloud computing, identifies the concepts and characteristics of scientific clouds, and presents an example of a scientific cloud for data centers. The project explains how scientific cloud computing supports large-scale scientific applications by providing high-performance computing resources, scalable storage, and flexible service management, improving resource utilization and enabling collaborative research.

3. Ensuring Data Storage Security in Cloud Computing (2023) – Cong Wang et al.

This article proposes an effective and flexible distributed scheme for ensuring cloud data storage correctness. By utilizing the homomorphic token with distributed verification of erasure-coded data, the scheme achieves the integration of storage correctness insurance and data error localization. The new scheme supports secure and efficient dynamic operations on data blocks, including data update, delete, and append, and shows resilience against Byzantine failure and malicious data modification.

4. Modified Caesar Cipher for Better Security Enhancement (2021) – Kashish Goyal and Supriya Kinger

The authors modified the traditional Caesar cipher by introducing dynamic changes during encryption based on even or odd alphabet index positions. This modification increases complexity and makes it more difficult for attackers to predict character patterns while maintaining simplicity and low computational cost, making it suitable for secure communication systems requiring lightweight encryption.

5. Cryptography Algorithm Comparison for Security Enhancement in WIDS (2019) – Gurjeevan Singh et al.

This study evaluates several encryption algorithms including AES, DES, 3DES, RC2, Blowfish, and RC6 for wireless intrusion detection environments. The comparison is based on parameters such as encryption speed, security strength, memory usage, and computational efficiency. Results show that the Blowfish algorithm provides better performance and security compared to other algorithms, concluding that efficient algorithm selection is essential for wireless intrusion detection system performance.

III. EXISTING SYSTEM

Traditional approaches to cloud security include basic firewall rules, static access control lists, and manual log monitoring — processes that are both time-consuming and heavily dependent on individual expertise. Early automated approaches relied on signature-based intrusion detection systems (IDS), but these methods struggle with zero-day attacks and advanced persistent threats (APTs). Rule-based filters and first-generation security tools offered some improvement in processing speed but limited detection accuracy due to their reliance on known attack signatures.

Cloud-based honeypots in existing systems mainly focus on detecting attacks and gathering information about threats occurring in the network environment. Once suspicious IP addresses are identified, vulnerability scanning and network analysis are performed to detect weaknesses in system architecture or software applications. However, the current systems have limitations in providing complete protection against advanced cyberattacks and maintaining secure cloud communication. Many existing solutions lack efficient real-time response mechanisms and strong data protection techniques.

Disadvantages of Existing System:

- Security issues and vulnerabilities still exist in cloud environments.
- Directly or indirectly leads to data loss and privacy risks.
- Unable to completely prevent malicious attacks and unauthorized access.
- High chances of malware injection and network exploitation.
- Existing systems may consume high computational and storage resources.
- Limited capability in detecting advanced and evolving cyber threats.

IV. PROPOSED SYSTEM

The proposed system discusses the detection of attacks in a cloud-based environment and the use of honeypots for improving cloud security by proposing a new security technique. In the proposed system, honeypots act as surveillance tools that continuously observe network activities and trap attackers by simulating real cloud services and resources. These honeypots help in detecting suspicious behaviour, collecting attack information, and generating early warnings whenever malicious activities are identified.

The proposed framework enhances cloud security by providing better intrusion detection, attack analysis, and real-time monitoring capabilities. It also helps administrators understand attacker behaviour and improve the overall defence mechanism of the cloud environment. The system is designed to reduce security risks, improve data protection, and provide a secure communication environment for cloud users. By integrating honeypot technology with cloud infrastructure, the system improves reliability, scalability, and overall network security while maintaining better performance and resource management.

Advantages of Proposed System:

- **Enhanced Security** — Honeypot technology effectively traps and monitors malicious users before they reach real system resources.
- **Real-Time Threat Detection** — The system continuously monitors network traffic and generates immediate alerts on suspicious activity.
- **Attacker Behaviour Analysis** — Detailed logs of attacker actions allow security teams to understand attack vectors and improve defences.
- **Scalability** — The cloud-based architecture efficiently handles large volumes of users and network requests.
- **Cost-Effectiveness** — Built on open-source Java EE and MySQL technologies, offering a cost-effective security solution.

V. SYSTEM ARCHITECTURE

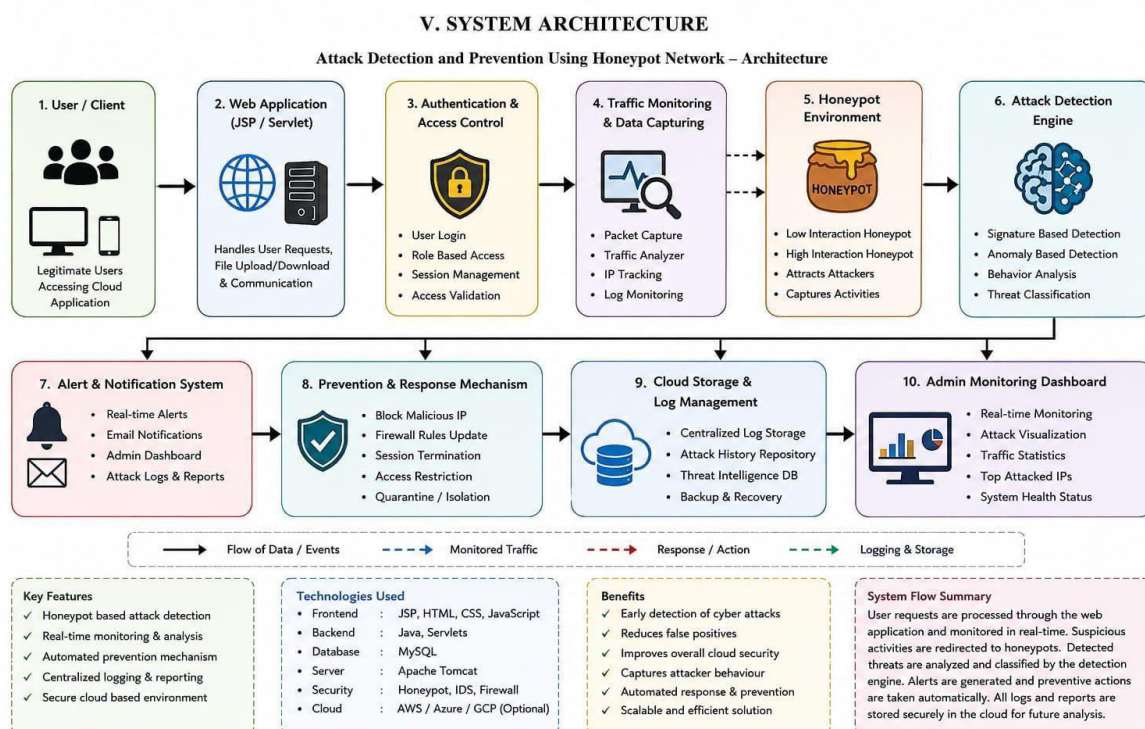


Fig. 1.1 System Architecture

Fig 1.1 System Architecture

The proposed attack detection system follows a structured architectural pipeline that transforms user interactions and attacker attempts into monitored, logged security events. The architecture integrates user management, network routing, honeypot decoy mechanisms, and database logging into a cohesive automated security workflow.

The system comprises three primary actor roles:

- User — Registers, logs in, uploads and downloads files, shares files through the router, and verifies received file data.
- Router — Creates and manages the network, views user IP addresses, accepts and shares files, and provides feedback mechanisms.
- Attacker — Attempts to access network resources either without honeypot protection (gaining file access) or with honeypot protection (redirected to decoy, gaining no real data).

The central database stores all user details, router details, and attacker interaction logs, enabling comprehensive audit trails and threat analysis. The honeypot component intercepts attacker traffic at the network level, presenting a convincing but isolated environment that reveals no actual user data while capturing full attacker behavioural data.

VI. METHODOLOGY

Module Names

Module 1: User Interface Design Module 2: Create Network Module 3: Router

Module 4: Attacker

Module 5: Honeypots

Module 6: User

Module 1: User Interface Design

To connect with server, the user must provide their username and password for authentication. If the user already exists, they can directly log in to the server; otherwise, the user must register their details such as username, password, Email ID, City, and Country. The database creates an account for each user to maintain upload and download rate. The interface provides login, registration, file management, and account update functionality through a web-based JSP front-end.

Module 2: Create Network

This is the foundational security module of the project. The server checks the authentication of the user and well improves the security, preventing unauthorized data access. The network involves one or more network controllers, each controlling a number of network elements within its domain via standard protocols. Each controller may run in multiple instances, each further managing a subset of network elements and backing up other instances to provide both scalability and high availability.

Module 3: Router

The router creates a network for new users and shows new user IP addresses. It accepts files from senders and routes them to the appropriate receiver. After accepting a file, the router stores it in shared files visible to the intended recipient. The router also collects and views feedback from users regarding network performance and security events.

Module 4: Attacker

Without Honeypot: The attacker logs in, selects an unused IP address, and opens an attack. In the absence of honeypot protection, the attacker gains access to files associated with the targeted IP address at the router waiting queue, demonstrating system vulnerability.

With Honeypot: When the attacker selects an IP address and launches an attack, the honeypot intercepts the request. No actual file data is returned to the attacker because the honeypot redirects the connection to a decoy environment, demonstrating effective protection against the attack.

Module 5: Honeypots

Honeypots are divided into two main types: Low-Interaction and High-Interaction Honeypots.

Low-Interaction Honeypots practice limited interaction. The attacker activity is limited to the level of emulation by the honeypot. They have a simple architecture and are easy to deploy and maintain with minimal risk.

However, limited information is logged and skilled attackers may eventually detect their presence. Examples include Specter, Honeyed, and KF Sensor.

High-Interaction Honeypots are complex systems providing real-time interaction with software and applications. Attackers are provided with real systems and software environments. Extensive amounts of information can be captured and an open environment is provided that captures all activity, allowing the system to learn unexpected attacker behaviours and advanced attack patterns.

Module 6: User

Authenticated users can update their accounts, upload new files to the server, share files with specific authorized recipients, check the status of shared files, and verify received file data. All file transactions are logged in the database to maintain an audit trail and support forensic analysis in case of security incidents.

Algorithm Used: IIDPS (Inside Intrusion Detection and Protection System)

The IIDPS consists of an SC monitor and filter, a mining server, a detection server, a local computational grid, and three repositories including user log files, user profiles, and an attacker profile. The enhanced protocol achieves more efficient mining performance with stronger privacy guarantees. The algorithm generates a user habit file using frequent item-set mining:

Input: U's log file where U is a user of the underlying system

Output: U's habit file

The algorithm uses a sliding window approach over the user's log file to identify recurring behavioural patterns. For each identified pattern, a binary action flag is set based on whether the user opened a specific link. If the identified pattern already exists in the habit file, its count is incremented; otherwise, the new pattern is inserted with an initial count of one. This builds a comprehensive user behaviour model that enables detection of anomalous access patterns indicative of unauthorized use or insider threats.

VII. ANALYSIS AND DISCUSSION

The proposed honeypot-based attack detection system for cloud environments demonstrates strong effectiveness in identifying and diverting malicious network traffic with improved security and reliability. The use of honeypot decoys provides a realistic simulation of cloud services, making them highly suitable for trapping and studying attacker behaviour without exposing real data.

During analysis, the network creation and routing modules play an important role in system functionality. User authentication, IP address management, and file sharing mechanisms ensure that legitimate users can access resources effectively while attackers are identified and redirected. The contrast between Without Honeypot and With Honeypot scenarios clearly demonstrates the effectiveness of the proposed security mechanism.

1. Performance Analysis

The proposed system shows strong performance in detecting and preventing unauthorized access in cloud environments. The honeypot mechanism effectively redirects attacker traffic away from real resources while capturing comprehensive behavioural data for analysis.

2. Security Analysis

The system provides improved security compared to traditional access control methods. The honeypot layer creates an additional barrier that isolates real user data from attacker access while maintaining full functionality for legitimate users.

3. Computational Analysis

The Java EE-based implementation requires moderate computational resources for operation. Processing network traffic and maintaining honeypot decoys introduces some overhead, but the security benefits significantly outweigh the performance cost in cloud deployment scenarios.

4. Data Analysis

The performance of the system depends on the quality and completeness of network monitoring data. Comprehensive logging of all attacker interactions enables better threat intelligence and improves future detection accuracy.

VIII. CONCLUSION

The proposed attack detection and prevention system using Honeypot Network provides an effective and reliable method for identifying and neutralizing cyber threats in cloud-based environments. By using honeypot decoy technology combined with a Java EE web application, the system can automatically detect unauthorized access attempts, redirect attacker traffic, and collect valuable threat intelligence without exposing real user data.

The system offers several advantages such as enhanced attack monitoring, improved security intelligence, reduced risk of data breaches, and support for cloud security professionals in incident response and threat analysis. Although challenges such as maintaining realistic honeypot environments, managing computational overhead, and preventing skilled attackers from detecting the decoys exist, the overall performance of the system is highly promising.

In conclusion, this project demonstrates that honeypot technology can play an important role in modern cloud security infrastructure by providing accurate, efficient, and intelligent attack detection solutions for enterprise and cloud computing applications.

IX. FUTURE ENHANCEMENT

The proposed attack detection system can be improved further in several ways to increase its performance and practical use. Advanced machine learning and artificial intelligence techniques can be integrated to automatically classify attack types, predict emerging threats, and enable adaptive honeypot configurations that evolve with attacker behaviour.

Future work can focus on integrating the system with Security Information and Event Management (SIEM) platforms for enterprise-grade threat correlation. Real-time automated response mechanisms can be developed to block identified attacker IP addresses instantly. Cloud-native deployment on platforms such as AWS, Azure, or Google Cloud can make the system more scalable and accessible.

Additional improvements such as 3D network visualization, automated forensic report generation, mobile alert notifications, stronger encryption for stored logs, and integration with threat intelligence feeds can make the system more efficient, accurate, and suitable for real-world enterprise cybersecurity applications.

REFERENCES

1. RuWei Huang, Si Yu, Wei Zhuang and XiaoLinGui, 'Design of Privacy-Preserving Cloud Storage Framework', 2010 Ninth International Conference on Grid and Cloud Computing, 2025.
2. Lizhe Wang, Jie Tao, Kunze M., Castellanos A.C., Kramer D., Karl W., 'Scientific Cloud Computing: Early Definition and Experience', 10th IEEE Int. Conference on High Performance Computing and Communications, pp. 825-830, Dalian, China, 2024.
3. Cong Wang, Qian Wang, KuiRen and Wenjing Lou, 'Ensuring Data Storage Security in Cloud Computing', In Quality of Service, 17th International Workshop, 2023.
4. Cong Wang, Qian Wang, Kui Ren and Wenjing Lou, 'Ensuring Data Storage Security in Cloud Computing', IEEE 2009.
5. Kashish Goyal, Supriya Kinger, 'Modified Caesar Cipher for Better Security Enhancement', International Journal of Computer Applications (0975-8887) Volume 73, No.3, July 2021.
6. Yogesh Kumar, Rajiv Munjal and Harsh Sharma, 'Comparison of Symmetric and Asymmetric Cryptography with Existing Vulnerabilities and Countermeasures', IJCSMS, Vol. 11, Issue 03, 2020.
7. Gurjeevan Singh, Ashwani Singla and K S Sandha, 'Cryptography Algorithm Comparison for Security Enhancement in Wireless Intrusion Detection System', International Journal of Multidisciplinary Research, Vol.1 Issue 4, 2019.
8. D. S. Abdul Elminaam, H. M. Abdul Kader and M. M. Hadhoud, 'Performance Evaluation of Symmetric Encryption Algorithms', Communications of the IBIMA, Volume 8, 2017.
9. Karthik Sadasivam, Banuprasad Samudrala, T. Andrew Yang, 'Design of Network Security Projects using Honeypots', University of Houston.
10. L. Spitzner, 'Honeypots: Tracking Hackers', Boston, USA: Addison Wesley, Parson Education, ISBN 0 321108957, 2003

International Journal of Advanced Research in Education and Technology

ISSN: 2394-2975

Impact Factor: 9.971