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 1, January 2026

DOI: 10.17148/IJARCCE.2026.151129

HONEYPOT-BASED WEB SECURITY MONITORING SYSTEM WITH WEB DASHBOARD

Manoj SP[1], Seema Nagaraj[2]

Department of MCA, BIT, K.R Road, V.V Pura ,Bangalore , India[1]

Assistant Professor, Department of MCA, BIT, K.R Road, V.V Pura ,Bangalore , India[2]

Abstract: Web-based systems are frequently targeted by malicious users due to their public accessibility and widespread usage. In many cases, security incidents are detected only after damage has occurred, as conventional protection mechanisms focus primarily on access control rather than behavioral observation. This paper presents a Honeypot-Based Web Security Monitoring System supported by a web dashboard, implemented using PHP, HTML, CSS, and JavaScript. The system deploys intentionally deceptive web components within an isolated environment to attract unauthorized interactions and record them for analysis. All captured events are stored and presented through a structured dashboard that supports manual inspection. The solution emphasizes simplicity, transparency, and educational value, making it suitable for internal academic evaluation and small-scale deployments.

Keywords: Honeypot System, Web Security Monitoring, Attack Logging, PHP-Based Dashboard, Cybersecurity Analysis

I. INTRODUCTION

The expansion of web technologies has led to an increased dependence on online applications for academic, administrative, and commercial activities. Alongside these developments, web servers are continuously exposed to suspicious access attempts, automated scanning, and malicious input submissions. While traditional security controls aim to restrict access, they often provide limited information regarding how attackers interact with exposed services.

A honeypot operates as a controlled decoy that mimics a vulnerable system. Its primary objective is not to block attacks but to observe them in a safe environment. By monitoring attacker behavior, valuable insights can be gained regarding commonly targeted resources and attack techniques. Such systems are particularly useful in academic contexts, where understanding real-world attack behavior is a learning objective.

This project focuses on building a web-based honeypot integrated with a monitoring dashboard. The implementation avoids artificial intelligence and advanced automation, relying instead on deterministic logging mechanisms and clear visualization to support internal assessment and learning.

1.1 Project Description

The system presents simulated web interfaces such as authentication forms and data entry pages that appear functional to external users. Interactions with these pages are treated as suspicious events. Backend components developed in PHP collect request metadata including source address, request method, submitted parameters, and access time. The collected information is securely stored and made accessible through an administrative dashboard.

1.2 Motivation

Many educational and small-scale web applications do not employ advanced security monitoring due to cost and configuration complexity. As a result, attempted intrusions remain undocumented. This project is motivated by the need for a lightweight monitoring mechanism that demonstrates attack behavior clearly without relying on complex frameworks. The design prioritizes clarity, ease of deployment, and suitability for internal academic review.

II. RELATED WORK

Existing research highlights the usefulness of honeypots in observing intrusion attempts within controlled environments. Prior studies show that decoy systems help identify frequently used attack methods and provide contextual understanding of adversarial behavior.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 1, January 2026
DOI: 10.17148/IJARCCE.2026.151129

Web-oriented honeypot implementations often focus on collecting HTTP request information to study brute-force access attempts and malicious parameter submissions. These approaches underline the importance of structured data collection for post-incident analysis.

Additional work emphasizes the role of dashboards in security monitoring. Clear presentation of recorded events improves human interpretation and supports efficient review of suspicious activities.

III. METHODOLOGY

A. Honeypot Interface Design

The honeypot environment consists of carefully designed web pages that resemble genuine application components. Although they do not perform real operations, their structure increases the likelihood of attracting unauthorized access attempts.

B. Attack Data Collection

Each interaction with the honeypot pages triggers logging routines that capture relevant request attributes such as origin address, request type, and payload content. These actions are recorded silently without notifying the user.

C. Backend Processing Workflow

Incoming requests are processed in an isolated manner to ensure that no actual system resources are affected. All interactions are redirected to logging mechanisms, preserving system integrity.

D. Dashboard-Based Visualization

The monitoring dashboard provides tabular summaries and statistical views of recorded events. This allows administrators to identify repeated access attempts and commonly targeted pages through manual review.

E. System Execution Flow

1. A user accesses a decoy web page
2. Request information is captured by the backend
3. Event data is stored in the database
4. Administrator logs into the dashboard
5. Logged activities are examined

F. Interpretation of Recorded Data

By organizing logs into readable formats, the dashboard supports identification of abnormal access patterns and repeated intrusion attempts.

G. Hardware and Software Requirements

Hardware:

The system can be deployed on a basic desktop or laptop with a minimum of 4 GB RAM. Continuous internet access is required for hosting the application.

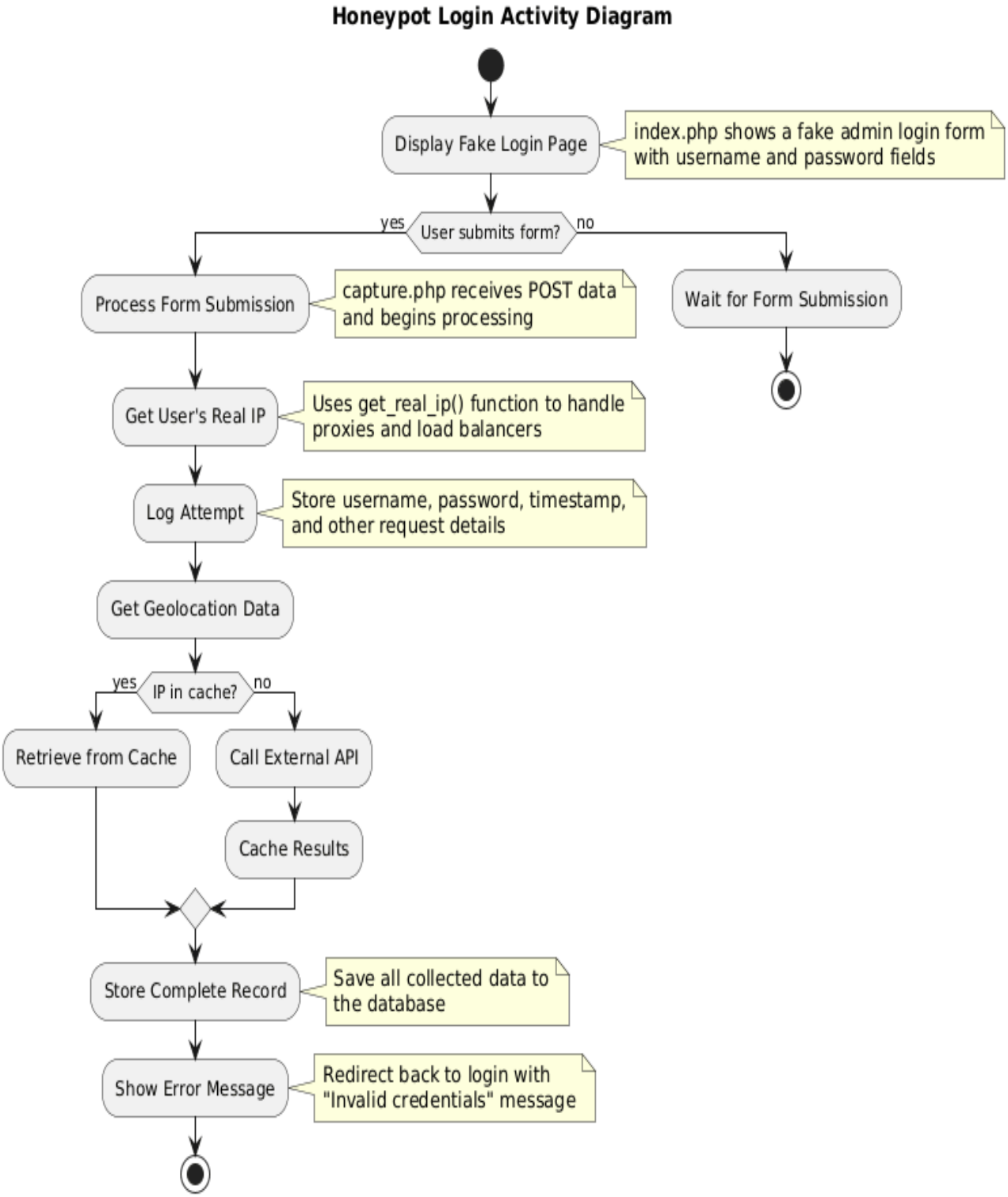
Software:
The solution is implemented using PHP for server-side logic and HTML, CSS, and JavaScript for the interface. Apache server and MySQL database (via XAMPP) are used for deployment. Any modern browser can be used to access the dashboard.

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 1, January 2026

DOI: 10.17148/IJARCCE.2026.151129



IV. SIMULATION AND EVALUATION FRAMEWORK

The evaluation process focuses on verifying whether the system reliably captures suspicious interactions and presents them in a clear manner. Controlled test scenarios were used to assess system behavior.

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 1, January 2026

DOI: 10.17148/IJARCCE.2026.151129

A. System Architecture Overview

The architecture includes three independent modules: decoy interface, logging storage, and visualization dashboard. Each module interacts through defined data flows.

B. Test Setup

Simulated scenarios included repeated invalid login attempts, abnormal input values, and direct access to restricted pages.

C. Evaluation Criteria

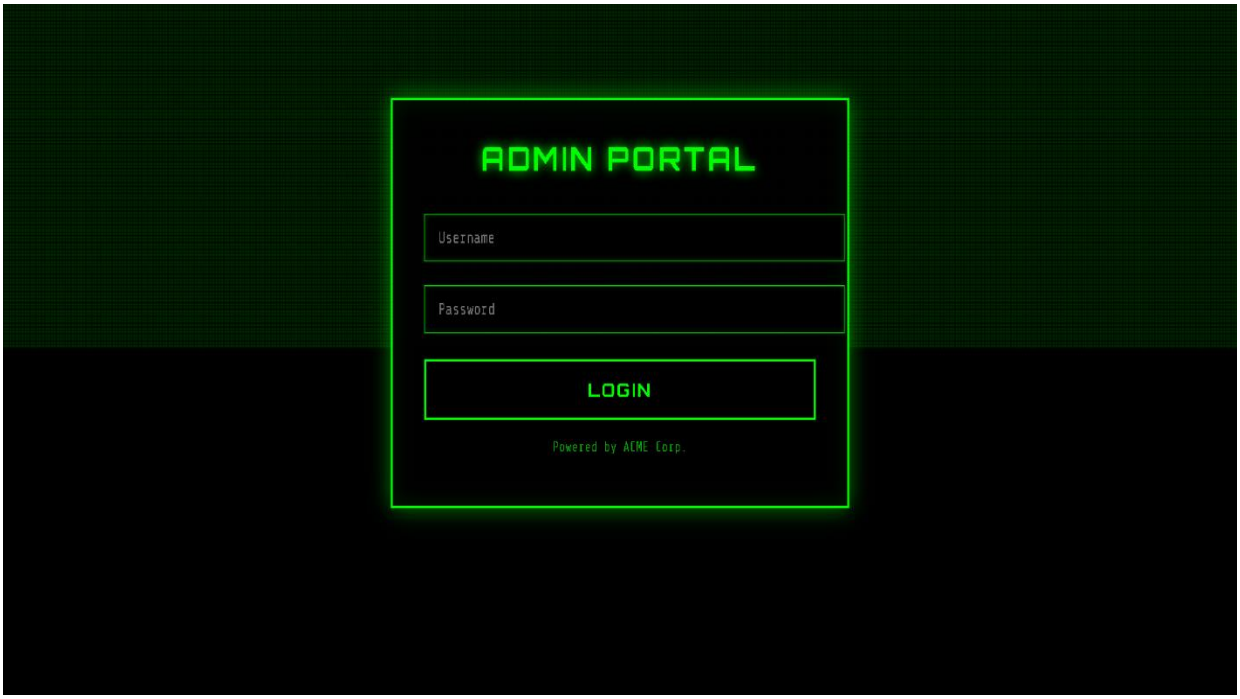
The system was assessed based on log completeness, operational stability, and clarity of displayed information.

D. Observations

All simulated interactions were successfully recorded. The dashboard remained responsive and provided consistent data views.

V. RESULTS AND DISCUSSION

The observed results confirm that the honeypot-based monitoring approach provides practical insight into web-based threats. The collected records revealed recurring access patterns and frequent targeting of specific interface components. Compared to generic server logs, the structured records generated by the honeypot were easier to interpret. The dashboard interface further supported effective manual analysis.



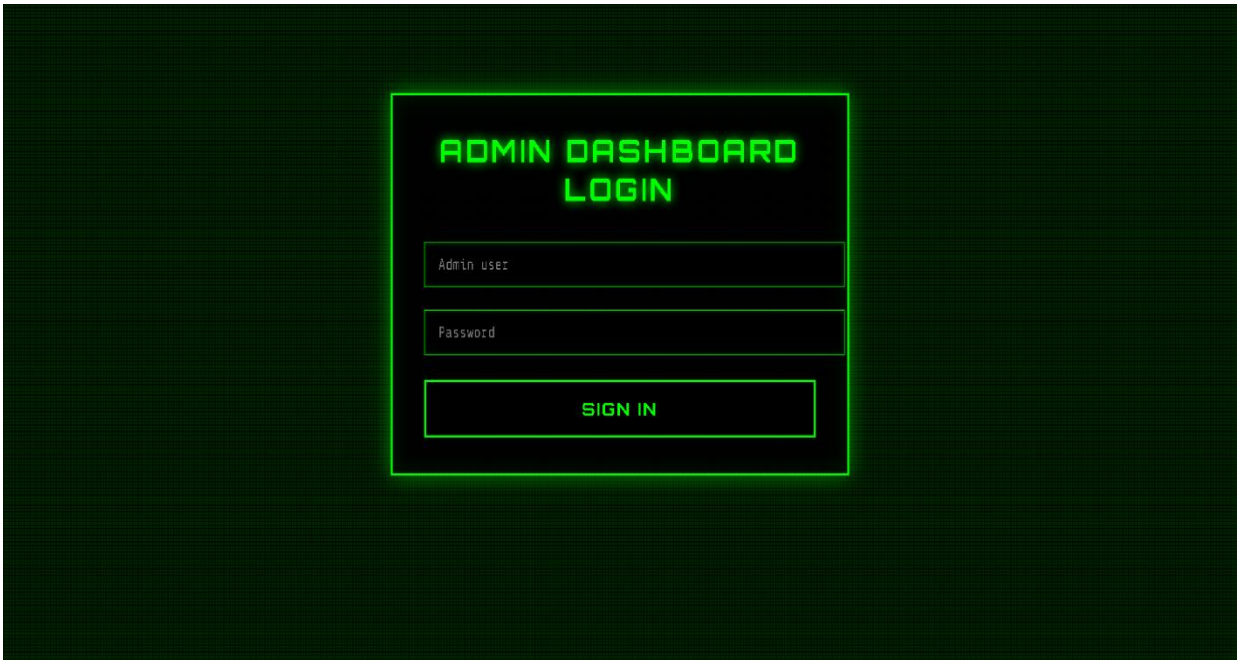
shows the Admin Portal Login interface designed as a decoy entry point in the Honeypot Security System.

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

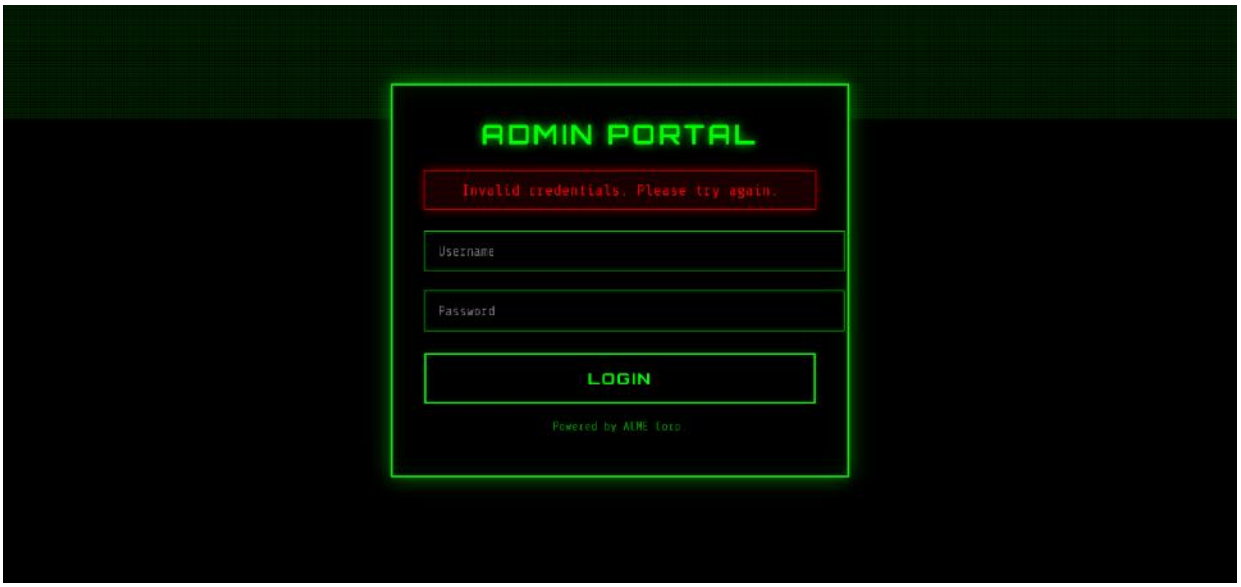
International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 1, January 2026

DOI: 10.17148/IJARCCE.2026.151129



illustrates the Admin Dashboard Portal Login page .



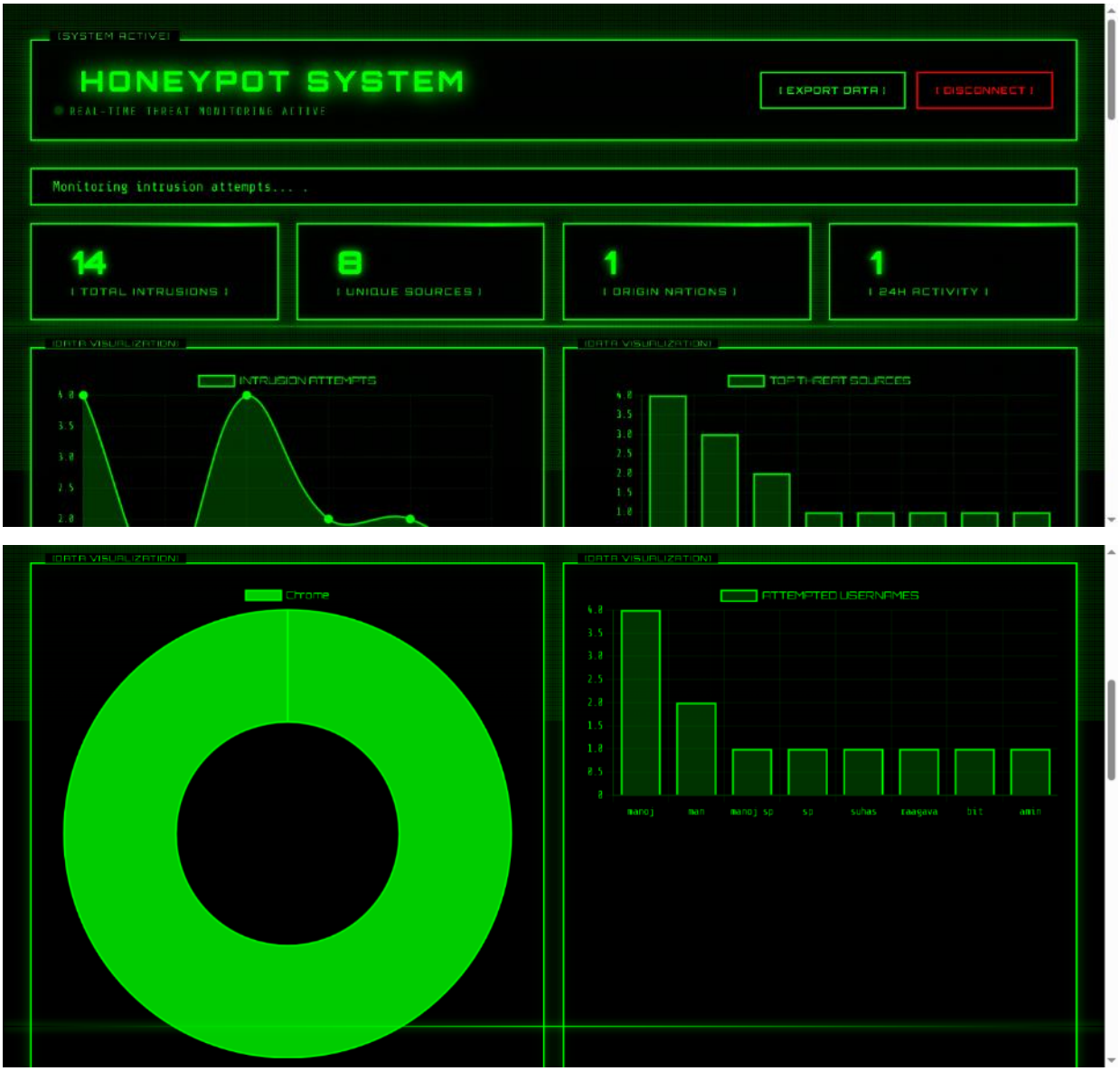
displays the Admin Portal Login interface when invalid credentials are submitted.

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 1, January 2026

DOI: 10.17148/IJARCCE.2026.151129



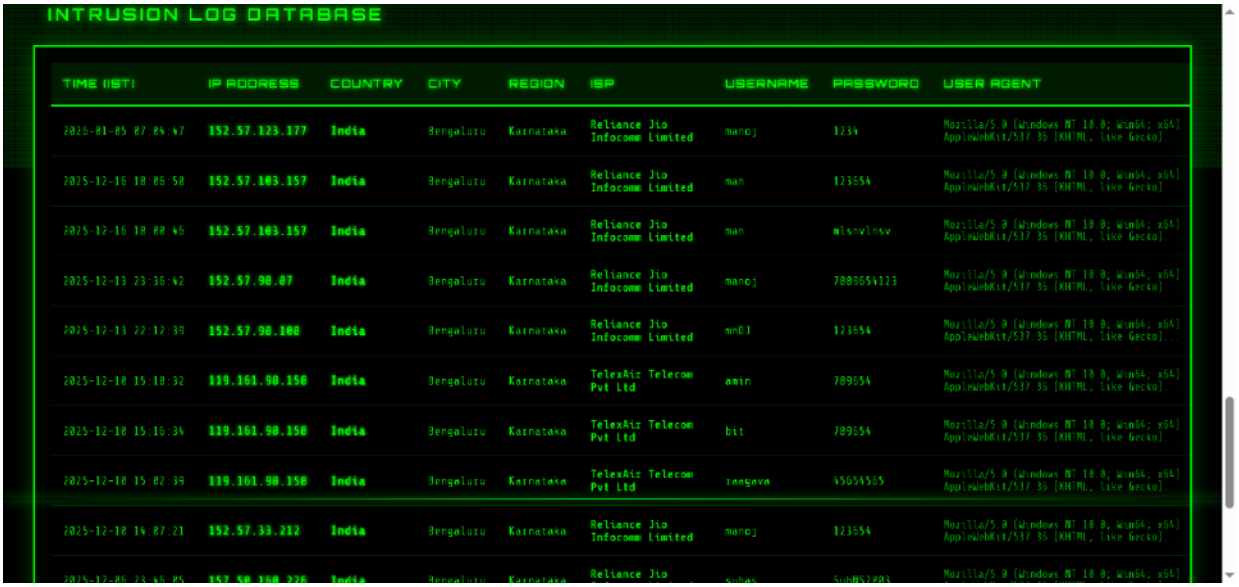
shows the graphical representation of intrusion attempts over time.

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 1, January 2026

DOI: 10.17148/IJARCCE.2026.151129



shows the Intrusion Log Database maintained by the honeypot system

VI. CONCLUSION

This work presents a practical implementation of a honeypot-based web security monitoring system tailored for internal academic evaluation. By combining controlled decoy interfaces with structured logging and a simple dashboard, the system enables observation of suspicious behavior without relying on automated intelligence. The approach supports learning objectives while maintaining system safety and clarity.

LIMITATIONS AND FUTURE WORK

The current implementation focuses on basic interaction logging and manual inspection. Future improvements may include alert mechanisms, enhanced filtering capabilities, and support for deploying multiple decoy instances. These extensions can further improve monitoring effectiveness while preserving system simplicity.

REFERENCES

1. Spitzner, L., *Honeypots: Tracking Hackers*, Addison-Wesley Professional, 2003.
2. Provos, N., and Holz, T., *Virtual Honeypots: From Botnet Tracking to Intrusion Detection*, Addison-Wesley, 2007.
3. Behl, A., *Cyberwar: The Next Threat to National Security and What to Do About It*, Oxford University Press, 2017.
4. Scarfone, K., and Mell, P., "Guide to Intrusion Detection and Prevention Systems (IDPS)," NIST Special Publication 800-94, National Institute of Standards and Technology, 2007.
5. Singh, S., and Silakari, S., "An Overview of Web Application Security," *International Journal of Computer Applications*, vol. 44, no. 12, pp. 1–6, 2012.
6. Alata, E., et al., "A Framework for the Design and Evaluation of Honeypots," *Proceedings of the IEEE International Conference on Computer Security*, pp. 1–8, 2006.
7. Garfinkel, T., Rosenblum, M., and Boneh, D., "Flexible OS Support and Applications for Trusted Computing," *USENIX Security Symposium*, 2003.
8. Kaur, J., and Kaur, K., "A Survey on Honeypot Based Intrusion Detection System," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 5, pp. 450–455, 2017.
9. OWASP Foundation, "OWASP Top 10 Web Application Security Risks," Available: https://owasp.org/www-project-top-ten/
10. Welling, L., and Thomson, L., *PHP and MySQL Web Development*, 5th Edition, Addison-Wesley, 2016.