

C3RBERUS – AI Based Incident Response System for Cybersecurity

¹Prof. Ambika S, ²Rumman Ahamed Khan, ³Syed Aqeeb Ahamad, ⁴Sandeep Raj S.N, ⁵Tarun Kumar S

¹Assistant Professor, Dept. of CSE, SJM Institute of Technology, Chitradurga, India

²Seventh semester Student, Dept. of CSE, SJM Institute of Technology, Chitradurga, India

³Seventh semester Student, Dept. of CSE, SJM Institute of Technology, Chitradurga, India

⁴Seventh semester Student, Dept. of CSE, SJM Institute of Technology, Chitradurga, India

⁵Seventh semester Student, Dept. of CSE, SJM Institute of Technology, Chitradurga, India

Abstract: The increasing number of cyber threats targeting Linux-based servers highlights the need for timely threat detection and automated incident response. This paper presents **C3RBERUS (Cybersecurity 3-layered Response-Based Enhanced Resilient Unified Shield)**, an AI-based incident response system that integrates SSH brute-force detection, an AI-powered honeypot, and ransomware detection into a unified security framework. The system monitors authentication logs, blocks suspicious IP addresses, provides a controlled environment for analyzing unauthorized activities, and detects potential ransomware behavior using canary files and file monitoring. A centralized dashboard provides real-time visibility into security events, alerts, and system status. The proposed framework combines threat detection, deception, monitoring, and automated response to provide a practical approach to cybersecurity incident management in Linux-based environments.

IndexTerms - Cybersecurity, Incident Response, SSH Brute-Force, Honeypot, Ransomware Detection, Canary Files, Automated Response, Linux Security.

I. INTRODUCTION

The rapid adoption of cloud computing and internet-connected infrastructure has increased the exposure of Linux-based servers to cyber threats such as SSH brute-force attacks, unauthorized access, and ransomware. Traditional security mechanisms such as firewalls and rule-based intrusion detection systems mainly depend on predefined rules and may have limited adaptability to evolving attack patterns. Artificial Intelligence (AI) and automation provide opportunities for improving threat monitoring and response by analyzing suspicious activities and supporting faster security actions. To address these challenges, this paper presents C3RBERUS (Cybersecurity 3-layered Response-Based Enhanced Resilient Unified Shield), an AI-based incident response system for Linux-based environments. The proposed system integrates three major security layers: SSH brute-force protection, an AI-powered honeypot for analyzing suspicious users, and proactive ransomware detection using file monitoring and canary files. It also provides automated response mechanisms and a centralized dashboard for monitoring security events, logs, blocked IP addresses, and system status. By combining threat detection, deception, monitoring, and automated response in a unified framework, C3RBERUS aims to provide a practical approach to managing cybersecurity incidents in Linux-based systems.

II. PROBLEM STATEMENT

Linux-based servers are frequently exposed to cyber threats such as SSH brute-force attacks, unauthorized access, and ransomware. Existing security mechanisms often operate independently and rely on predefined rules, which can limit their ability to detect evolving attack behavior and provide timely responses. There is a need for a unified cybersecurity system that can continuously monitor server activity, detect suspicious behavior, analyses potential threats, and automatically initiate appropriate response actions. Therefore, the proposed C3RBERUS system aims to integrate SSH brute-force detection, an AI-powered honeypot, ransomware monitoring, automated response, and centralized security monitoring into a single framework for Linux-based environments.

III. RELATED WORK / LITERATURE REVIEW

Research on SSH brute-force detection has explored both machine learning and deep learning approaches. Hynek et al. (2020) investigated machine learning-based detection using network traffic logs and reported effective detection of SSH brute-force attacks, although the approach was limited to network-level analysis and lacked real-time adaptability. Wanjuan et al. (2021) used a Convolutional Neural Network (CNN) with SSH authentication logs and reported approximately 94% detection accuracy, while also identifying the need for large training datasets and higher computational resources.

Ransomware detection has also been studied using machine learning and behavioural analysis. Ahmed et al. (2022) proposed a deep learning and ensemble-based approach for ransomware detection, reporting high detection rates with low false positives, but the approach involved greater model complexity and resource requirements. Al-Rimy et al. (2023) investigated behavioural analysis for early ransomware detection and identified limitations related to dataset availability and response delay.

Cyber deception and honeypot technologies have been investigated as methods for studying attacker behaviour. Javadpar et al. (2024) examined honeypot-based cyber deception and reported improved attacker engagement and data collection, while noting

that skilled attackers may identify honeypot environments. Morić et al. (2025) explored AI-based honeypots capable of creating more adaptive and realistic environments, but the approach introduced higher computational requirements and dependency on AI models.

Recent research has also examined the integration of multiple cybersecurity techniques. Satpute et al. (2025) proposed an approach combining AI, intrusion detection systems, and honeypots, with reported improvements in attack detection but challenges related to integration and scalability. Hassan and Ibrahim (2023) investigated AI-driven cybersecurity systems for automated incident response and threat analysis, while identifying reduced interpretability as a limitation of AI-based approaches. Al-Mohannadi et al. (2025) investigated AI-based optimization for cyber-threat detection and highlighted the need for continuous model updates and tuning.

Overall, the reviewed studies demonstrate the use of machine learning, deep learning, behavioural analysis, honeypots, and AI-driven automation for addressing individual cybersecurity challenges. However, the project literature survey identifies a need for integrating these capabilities into a unified framework. C3RBERUS addresses this area by combining SSH brute-force detection, an AI-powered honeypot, ransomware monitoring, automated response, and centralized security monitoring within a single Linux-based system.

IV. PROPOSED METHODOLOGY

The proposed C3RBERUS methodology follows a sequential process for identifying security incidents and initiating appropriate responses. The system first collects security data from Linux authentication logs, monitored files, and other relevant system events. The collected information is then analyzed to identify suspicious behavior, such as repeated SSH login failures, unauthorized user activity, or abnormal changes to monitored files. When suspicious activity is identified, the corresponding event is classified according to the type of threat and the appropriate response mechanism is selected. For SSH brute-force activity, the system can block the source IP address using firewall rules, while suspicious users can be directed to the honeypot for controlled observation. In the case of suspected ransomware activity, changes to canary or monitored files can trigger an alert and predefined response actions. The generated events and response actions are subsequently logged and forwarded to the centralized monitoring interface, allowing administrators to review incidents and system status. This methodology enables C3RBERUS to follow a continuous cycle of monitoring, detection, analysis, response, and reporting within a Linux-based environment.

V. SYSTEM ARCHITECTURE

The C3RBERUS system is organized as an integrated security framework in which different components communicate to detect threats, perform response actions, and provide centralized monitoring. The architecture consists of the following major components:

1. Linux Server: Acts as the primary environment where SSH services, monitored files, security tools, and C3RBERUS components operate.
2. Security Modules: The system includes SSH brute-force monitoring and blocking, an AI-powered honeypot, and ransomware/file monitoring components. These modules generate security events based on detected activities.
3. Backend and Processing Layer: The Python-based security components process logs, monitor system activity, perform detection, and execute automated response actions. Node.js and Express provide the backend services required for communication with the monitoring interface.
4. Database: MongoDB is used for storing relevant security events, logs, alerts, and other information required by the system.
5. Centralized Dashboard: The dashboard provides administrators with a consolidated view of security events, including blocked IP addresses, honeypot activity, ransomware alerts, logs, and system status.
6. AI Integration: Large Language Model APIs are used as part of the AI-powered honeypot to generate contextual responses during suspicious-user interaction.

VI. IMPLEMENTATION

The C3RBERUS system is implemented and deployed on an AWS EC2 Linux environment, which serves as the target server for security monitoring and incident response. The implementation uses Python and JavaScript, with Node.js and Express for backend services, MongoDB for storing security logs and events, and LLM APIs for the AI-powered honeypot. The security components use Linux tools such as journalctl for monitoring SSH authentication logs and iptables for automated IP blocking.

1. SSH Attack Detection and Prevention: The system continuously monitors SSH authentication activity on the EC2 instance using journalctl. Failed login attempts are tracked based on their source IP addresses, and when the configured threshold is exceeded, the suspicious IP is automatically blocked using iptables.

2. **AI-Powered Honeypot:** Suspicious or unverified users are redirected to a simulated Linux environment. The honeypot uses an LLM API to generate realistic responses to user commands, while interactions are recorded for behavioural analysis.

3. **Ransomware Detection:** The system uses hidden canary files and file monitoring to identify unauthorized modifications. When a monitored canary file is modified, a potential ransomware event is detected and the configured response action is triggered. The event is also logged for further analysis.

4. **Dashboard and System Integration:** The Node.js and Express backend integrates the security modules and communicates with the monitoring dashboard. MongoDB stores relevant security events, while the dashboard displays information such as blocked IP addresses, honeypot activity, ransomware alerts, logs, and system status.

Overall, the implementation combines the individual security modules within the AWS EC2-based Linux environment to provide continuous monitoring, automated threat response, logging, and centralized visualization.

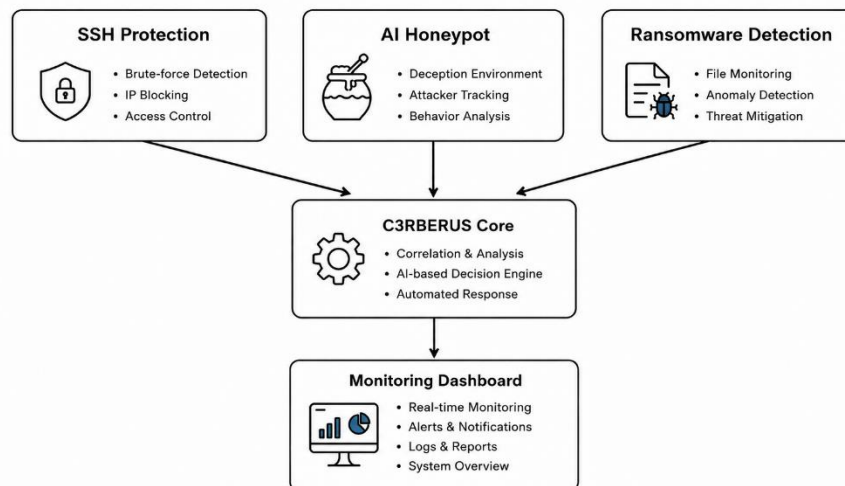


Fig. 1: System Architecture

VII. EXPERIMENTAL SETUP

The C3RBERUS system is evaluated in an AWS EC2-based Linux environment, which provides the server environment for deploying and testing the security modules. The experimental setup uses Ubuntu Linux, with Python for the security components and JavaScript with Node.js and Express for backend and dashboard integration. MongoDB is used for storing security events and logs, while journalctl and iptables are used for SSH log monitoring and IP blocking. The system also requires internet connectivity for SSH access, dashboard communication, and LLM API integration.

The evaluation involves testing the major security functions of C3RBERUS independently and as an integrated system. SSH brute-force detection is tested using repeated failed authentication attempts to verify detection and IP blocking. The AI-powered honeypot is tested by generating suspicious-user interactions and observing whether commands and responses are recorded correctly. The ransomware detection module is evaluated by modifying monitored canary files to verify alert generation and the configured response mechanism. The monitoring dashboard is tested to confirm that security events from the different modules are logged and displayed correctly. These tests are intended to evaluate the system's detection, response, logging, and monitoring capabilities.

Experimental Environment:

- Cloud Platform: AWS EC2
- Operating System: Ubuntu Linux
- Programming: Python, JavaScript
- Backend: Node.js, Express
- Database: MongoDB
- Security Tools: journalctl, iptables
- AI Integration: LLM API
- Dashboard: Web-based centralized monitoring interface

VIII. RESULTS AND DISCUSSION

The C3RBERUS system was tested in a cloud-based Linux environment to evaluate SSH brute-force detection, AI-powered honeypot interaction, ransomware detection, centralized monitoring, and email notifications. The tests confirmed that the major modules functioned as intended.

1. SSH Brute-Force Detection: Repeated failed SSH login attempts were detected, and the source IP was automatically blocked using iptables.
2. AI-Powered Honeypot: Unverified users were redirected to the honeypot, where their commands were logged and AI-generated responses were provided.
3. IP Verification and OTP: Unverified IPs were required to complete verification. Incorrect OTP attempts were recorded, while successfully verified IPs were added to the trusted list.
4. Ransomware Detection: Modification of a monitored file was detected and triggered the configured emergency response.
5. Email Notifications: The system sent security notifications for events such as honeypot activation, incorrect OTP or password attempts, SSH attacks, IP blocking, and ransomware detection.
6. Centralized Monitoring: Security events were stored and displayed through the dashboard, including blocked IPs, honeypot activity, ransomware incidents, and authentication events.
7. Overall Performance: Testing showed consistent event logging, dashboard updates, and stable operation across the tested security scenarios.

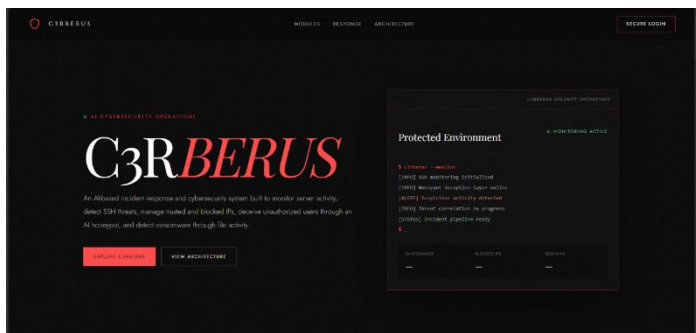


Fig. 2: Landing Page

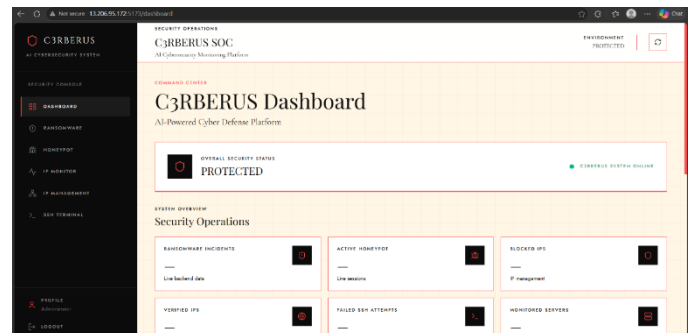


Fig. 3: Dashboard

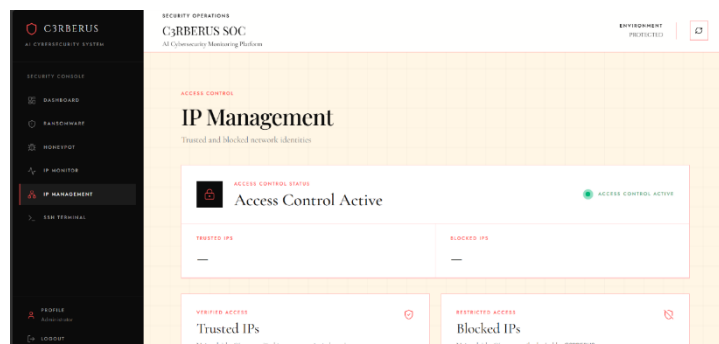


Fig. 4: IP Management

X. CONCLUSION AND FUTURE WORK

Conclusion

C3RBERUS provides a multi-layered incident response framework for protecting Linux-based cloud servers. The system combines SSH brute-force detection, IP verification, an AI-powered honeypot, ransomware detection, automated response, centralized monitoring, and email-based security notifications. Testing demonstrated that the implemented modules could detect and respond to the defined security scenarios while maintaining centralized logs and alerts.

Future Work

Future development can focus on extending C3RBERUS with **adaptive machine-learning models**, support for **multiple servers and distributed cloud environments**, and more advanced ransomware detection using behavioural and memory-level analysis. Additional notification channels such as SMS, mobile alerts, and SIEM integration can also be incorporated. Threat-intelligence integration, improved dashboard analytics, role-based access control, and more automated remediation workflows can further enhance the system.

XI. REFERENCES

- [1] A. Ahmad et al., “AI-Based Intrusion Detection System – Comprehensive Review,” *Journal of Cybersecurity and Artificial Intelligence*, 2023.
- [2] M. Alshammari et al., “Explainable AI in Intrusion Detection Systems,” *International Journal of Information Security*, 2025.
- [3] R. Bhardwaj et al., “AI-Powered Network Intrusion Detection Evaluation,” *Cybersecurity Review Journal*, 2024.
- [4] P. Singh et al., “SSH Brute-Force Attacks Analysis in the Wild,” *Cloud Security Research Journal*, 2024.
- [5] J. Wanjau et al., “Deep Learning for SSH Attack Detection,” *International Journal of Network Security*, 2021.
- [6] S. Javadpour and H. Ja’fari, “Survey on Cyber Deception Techniques for Honeypots,” *Journal of Cyber Defence Technologies*, 2024.
- [7] M. Ferrag et al., “Comprehensive LLM in Cybersecurity Review,” *Artificial Intelligence in Security Journal*, 2024.
- [8] D. Williams et al., “Cybersecurity with Deep Learning,” *Advanced Computing and Security Journal*, 2024.

Copyright & License:

© Authors retain the copyright of this article. This work is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.