International Journal of Scientific Research & Engineering Trends
Volume 11, Issue 1, Jan-Feb-2025, ISSN (Online): 2395-566X

Detecting Unauthenticated Access Using Honeypot Sentinel

Varshini J, Asvica J, Bharathi A K, Deepika P, Dharshana S, Yazhini K
Department of Computer Science and Engineering (Cyber Security)
Sri Shakthi Institute of Engineering and Technology Coimbatore, India

Abstract- Unauthorized access remains a critical threat to network security, as attackers can exploit vulnerable systems to obtain sensitive data or disrupt services. This paper introduces Honeypot Sentinel, a proactive intrusion detection tool designed to flag unauthorized access attempts by monitoring and verifying usernames and IP addresses. Honeypot Sentinel uses a MongoDB database for logging, enabling the system to record details of unauthorized access attempts efficiently. Upon detecting any access attempts outside the predetermined criteria, Honeypot Sentinel triggers alerts, allowing system administrators to promptly address potential threats. This approach provides network security teams with real-time data, helping them respond effectively to unauthorized access incidents.

Index Terms- Honeypot, Unauthorized Access Detection, Network Security, MongoDB, Intrusion Detection.

I. INTRODUCTION

With the ever-increasing sophistication of cyber threats, organizations face unprecedented challenges in securing their digital infrastructures. Attackers, both external and internal, employ advanced techniques to bypass traditional defenses, leaving networks vulnerable to exploitation.

Honeypots, as discussed in [1] and [2], offer an innovative approach by serving as decoy systems that lure attackers and capture their activities. By observing and analyzing intrusions, honeypots provide valuable insights into the tactics and techniques of cyber adversaries. Honeypot Sentinel builds on this concept by integrating honeypot technology with robust authentication and logging mechanisms to create a comprehensive intrusion detection and analysis solution.

Honeypot Sentinel leverages MongoDB as its primary storage engine, adhering to the best practices for scalable and flexible data management outlined in [3]. This allows the system to handle large volumes of security logs and metadata, including IP addresses, timestamps, and abuse scores. Unlike traditional security systems that only block unauthorized access, Honeypot Sentinel records and analyzes all access attempts, providing real-time alerts to administrators about suspicious activities. The approach aligns with the recommendations in [4] and [6], emphasizing the critical role of proactive monitoring and detailed logging in modern network security frameworks.

The system's architecture incorporates advanced cryptographic protocols and secure authentication mechanisms to ensure data integrity and confidentiality, consistent with the principles outlined in [9] and [10]. By capturing and analyzing attack vectors, Honeypot Sentinel identifies potential vulnerabilities and emerging threats, offering actionable intelligence to network administrators.

Furthermore, the integration of machine learning techniques, as highlighted in [14], enhances the system's ability to detect anomalies and predict malicious behavior, enabling organizations to stay ahead of evolving cyber threats.

By combining honeypot technology with real-time monitoring, logging, and advanced analytics, Honeypot Sentinel addresses the pressing cybersecurity needs of modern organizations. Its design reflects the guidelines outlined in [15], [16], and [20], ensuring comprehensive protection against a wide range of cyber threats. As networks become more complex and attackers more sophisticated, solutions like Honeypot Sentinel provide a critical layer of defense, equipping organizations with the tools needed to detect, analyze, and mitigate threats effectively. This integrated approach not only improves immediate threat response but also informs long-term strategies for bolstering network security.

Objective

- Detect Unauthorized Access Attempts: Identify access attempts that deviate from the accepted username and IP address criteria.
- Log Unauthorized Access: Store detailed records of unauthorized attempts for future analysis and reference.

© 2025 IJSRET

- Real-Time Alerts: Provide administrators with instant notifications for prompt response to potential security breaches.
- Enhanced Visibility: Allow network administrators to view a comprehensive history of access attempts, authorized and unauthorized.
- Integrate with Existing Security Measures: Complement existing security frameworks and tools to strengthen overall network protection.

II. LITERATURE SURVEY

Spitzner [1] explores the foundational concepts of honeypots, detailing their application in tracking and analyzing malicious activities. This seminal work highlights the potential of honeypots to reveal attacker strategies and tactics. Provos and Holz [2] expanded the domain with their study on virtual honeypots, which facilitate scalable monitoring of network intrusions and botnets. Their contribution emphasizes the role of deception technology in enhancing proactive defense mechanisms.

MongoDB, Inc. [3] provides comprehensive guidelines on implementing logging and monitoring in database systems. These strategies ensure operational performance and enable the real-time detection of anomalies and security incidents. Similarly, Chuvakin et al. [17] discuss the principles of log management, emphasizing efficient collection, analysis, and retention of logs to meet security and compliance requirements.

Scarfone and Mell [4] provide an in-depth guide to intrusion detection and prevention systems, categorizing tools based on their capabilities and effectiveness. Their framework remains a key resource for understanding the dynamics of IDPS implementation. Northcutt and Novak [5] offer complementary insights into network intrusion detection systems, focusing on detection mechanisms and practical deployment.

Schneier [9] and Ferguson and Schneier
[10] present foundational knowledge on cryptography, emphasizing the design of secure protocols and algorithms to ensure data confidentiality and integrity. These works are critical for understanding the application of encryption in modern cybersecurity.

Stallings [6] and Subramanian [19] provide detailed overviews of network security fundamentals, including firewalls, secure communication protocols, and threat mitigation techniques. Their work highlights evolving strategies to secure network infrastructures. Kurose and Ross [16] supplement these insights with a top- down approach to networking, focusing on vulnerabilities in layered architectures.

Bhattacharyya and Kalita [11] explore the application of machine learning in detecting network anomalies. Their study demonstrates how advanced algorithms can identify patterns indicative of security breaches, showcasing the synergy between AI and cybersecurity.

Kindervag [18] introduces the Zero Trust model, a paradigm that assumes no inherent trust within networks. This approach prioritizes identity verification and access control, enabling organizations to secure environments effectively against advanced threats.

Lambert [20] discusses digital forensics for network, internet, and cloud environments, providing practical insights into evidence collection and incident analysis.

Garfinkel and Spafford [14] complement this with practical approaches to securing UNIX systems, focusing on system hardening and secure configurations.

NIST [15] outlines a comprehensive cybersecurity framework that offers guidelines for risk management and structured response strategies. This framework enables organizations to establish resilient defenses against evolving threats.

III. METHODOLOGY

1. Objective
The proposed system is designed to analyze and log IP address details using APIs such as IPinfo and AbuseIPDB while implementing rate-limiting and MongoDB for secure, scalable data management.

2. SystemSetup
Flask, Requests, and MongoDB were installed and configured. API keys for IPinfo and AbuseIPDB were securely stored. MongoDB Atlas was integrated for persistent storage.

3. Implementation
- **IP Analysis:** Data retrieval from IPinfo (geolocation) and AbuseIPDB (abuse score).
- **Data Logging:** Transitioned from CSV- based logging to MongoDB collections for efficient querying and storage.
- **Rate-Limiting:** Enforced 2 requests per minute per IP to prevent abuse.

Authentication: Implemented token-based authentication with credentials stored in the users collection.

© 2025 IJSRET

4. Testing and Deployment
The system was tested for API accuracy, rate-limiting enforcement, and MongoDB operations. It was deployed on Heroku with MongoDB Atlas for cloud scalability.

This methodology ensures reliable IP analysis, data security, and compliance with modern web application standards.

IV. EXISTING SYSTEM

Many network security solutions rely on traditional intrusion detection systems (IDS) that either permit or deny access based on password authentication. However, these systems lack honeypot capabilities, which can log suspicious activity and aid forensic analysis.

Common intrusion detection solutions include:
- **Traditional IDS:** Detect unauthorized access patterns but often lack detailed logging or honeypot functionality.
- **Firewall-Based Detection:** Blocks unauthorized IP addresses but does not provide visibility into access attempts that might have bypassed these controls.
- **Multi-Factor Authentication (MFA):** Enhances security by requiring multiple authentication steps but lacks logging and monitoring of access attempts that do not meet pre-set criteria.

Disadvantages
- **Limited Logging Capabilities:** Many systems do not log unauthorized access attempts, limiting post-incident analysis.
- **High Alert Fatigue:** IDS often generate excessive alerts, which can overwhelm security teams and lead to ignored warnings.
- **Poor Scalability:** Traditional databases may not handle large volumes of log data effectively.
- **No User Insight:** Most systems do not analyze the intent behind access attempts, missing potential threat patterns.

V. PROPOSED SYSTEM

Honeypot Sentinel is designed to address these shortcomings by providing a honeypot-based solution integrated with MongoDB for secure data storage and retrieval.

Key features of the proposed system include:
- **Honeypot**-Driven Access Detection: Detects unauthorized access attempts by validating usernames and IPs against a secure database.
- **Real-Time Alerting:** Alerts administrators upon detection of unauthorized access attempts.
- **Scalable Logging via MongoDB:** Enables efficient storage and retrieval of access logs, providing scalability for high-traffic environments.
- **User-Friendly Interface:** Allows administrators to monitor activity logs and configure alert settings easily.
- **Cross-Platform Integration:** Supports integration with other network security solutions to enhance the overall defense strategy.

VI. SYSTEM REQUIREMENTS

Hardware
- Device with Intel Core i5 processor or equivalent.
- Minimum of 2 GB RAM.
- 100 MB of free storage space for data and application files.
- Stable internet connection.

Software
- Python for scripting and automation.
- MongoDB for database management.

Module Description
Authentication Module
- **Objective:** To ensure secure access to the API.
- **Description:** Implements token-based authentication using a MongoDB users collection to validate credentials and restrict unauthorized access.

IP Analysis Module
Objective: To retrieve and analyze IP address details.
Description:
- Uses the IPinfo API to gather geolocation details, including country and region.
- Fetches abuse confidence scores from the AbuseIPDB API to assess the threat level of the IP address.

Rate-Limiting Module
- **Objective:** To prevent abuse of the API by restricting request frequency.
- **Description:** Implements a middleware- based mechanism that limits requests to 2 per minute per IP address by tracking request timestamps.

Data Logging Module
- **Objective:** To maintain a persistent record of IP analysis data.
- **Description:** Logs IP address, geolocation data, abuse confidence score, and timestamp into MongoDB for efficient storage and auditing.

Testing and Deployment Module
Objective: To validate functionality and deploy the system in a production environment.
Description:

© 2025 IJSRET

- Tests API responses, rate-limiting logic, and MongoDB integration.
- Deploys the Flask application on Heroku with cloud-based MongoDB Atlas for scalable database storage.

VII. CONCLUSION

Honeypot Sentinel represents a cutting-edge solution for network security, designed to address the escalating risks of unauthorized access in modern organizations. Its primary focus is real- time detection and logging of unauthorized access attempts, employing an innovative honeypot-based approach to trap and monitor potential intruders. This sophisticated technology places decoy systems or "honeypots" within the network, which are deliberately left open to unauthorized users. When attackers interact with these honeypots, Honeypot Sentinel captures their actions, gathering valuable data on attempted breaches, techniques used, and potential points of vulnerability within the network.

The platform's strength lies in its integration with MongoDB, a highly scalable NoSQL database that can manage large volumes of complex data in real time. MongoDB's architecture supports vast, rapidly growing logs that include essential details such as IP addresses, geographic locations, timestamps, entry points, and any suspicious activity observed. This information is instantly available for analysis, making it easier for administrators to assess patterns, identify frequent intrusion attempts, and trace the source of attacks. MongoDB's flexibility allows Honeypot Sentinel to adapt as the network scales, ensuring consistent, uninterrupted logging and analysis regardless of data volume.

Beyond simply blocking access, Honeypot Sentinel adds a proactive layer to network defense. Instead of passively denying unauthorized users, it captures and records interactions, creating an invaluable pool of data for forensic analysis. This forensic data enables administrators to conduct thorough investigations post-incident, revealing insights into the strategies and behavior of attackers. By understanding these patterns, organizations can not only address immediate threats but also strengthen long-term security strategies. These logs can reveal vulnerabilities that might otherwise go undetected, giving security teams the opportunity to reinforce specific areas of their network infrastructure.

Moreover, Honeypot Sentinel's real-time alerts empower administrators to act swiftly. When suspicious activity or an attempted breach is detected, the system immediately notifies security personnel, enabling rapid containment and response before the attacker can progress further into the network. This immediate detection- response cycle is crucial in limiting the potential damage of an unauthorized access attempt, whether from an external attacker or an insider with malicious intent.

For organizations seeking a more robust, proactive approach to security, adopting Honeypot Sentinel provides several critical advantages. Not only does it prevent unauthorized access, but it also facilitates a comprehensive view of network security by combining detection, monitoring, and forensic data collection. This empowers organizations to respond proactively to threats, informed by historical data and analysis. By bolstering their network defenses in this way, organizations strengthen their overall security posture, reducing risks and enhancing their ability to anticipate and respond to emerging threats effectively.

REFERENCES

1. Spitzner, L. Honeypots: Tracking Hackers. Addison-Wesley, 2002.
2. Provos, N., & Holz, T. Virtual Honeypots: From Botnet Tracking to Intrusion Detection. Addison-Wesley Professional, 2007.
3. MongoDB, Inc. MongoDB Documentation: Logging and Monitoring. Retrieved fromhttps://www.mongodb.com/docs/manual/administration/monitoring/, 2023.
4. Scarfone, K., & Mell, P. Guide to Intrusion Detection and Prevention Systems (IDPS): Recommendations of the National Institute of Standards and Technology. NIST Special Publication 800-94, 2007.
5. Northcutt, S., & Novak, J. Network Intrusion Detection. New Riders Publishing, 2001.
6. Stallings, W. Network Security Essentials: Applications and Standards. Pearson, 2018.
7. Bishop, M. Introduction to Computer Security. Addison-Wesley, 2005.
8. Bruneau, J. IP Address-based Intrusion Detection in Network Security. ACM Computing Surveys, 2020.
9. Schneier, B. Applied Cryptography: Protocols, Algorithms, and Source Code in C. Wiley, 2015.
10. Ferguson, N., & Schneier, B. Practical Cryptography. Wiley, 2003.
11. Bhattacharyya, D. K., & Kalita, J. K. Network Anomaly Detection: A Machine Learning Perspective. CRC Press, 2013.
12. Strebe, M. Network Security Foundations. Wiley, 2006.
13. Ullman, J. D., & Aho, A. V. Foundations of Computer Security. MIT Press, 2009.
14. Garfinkel, S., & Spafford, G. Practical UNIX and Internet Security. O'Reilly Media, 2003.
15. NIST. National Cybersecurity Framework Manual. NIST Press, 2017.
16. Kurose, J., & Ross, K. Computer Networking: A Top-Down Approach. Pearson, 2016.
17. Chuvakin, A., Schmidt, K., & Phillips, C. Logging and Log Management: The Authoritative Guide to

© 2025 IJSRET

Understanding the Concepts Surrounding Logging and Log Management. Syngress, 2013.
18. Kindervag, J. Zero Trust Networks: Building Secure Systems in Untrusted Networks. O'Reilly Media, 2017.
19. Subramanian, V. Basics of Network Security. Springer, 2020.
20. Lambert, R. Digital Forensics for Network, Internet, and Cloud Computing: A Forensic Evidence Guide for Moving Targets and Data. Syngress, 2020.

© 2025 IJSRET