Honeypot-Based Network Security System for Attack Detection and Analysis

Prof. Nitin Jain [1], Chandana N R [2], Deeksha U[3], Dhanyatha M[4], Prakruthi S B [5]
[1]*Associate Professor, Department of Information Science and Engineering, Bahubali College of Engineering, Shravanabelagola*
[2,3,4,5] *UG Student, Department of Computer Science and Engineering Bahubali College of Engineering, Shravanabelagola*

***Abstract*—Cyber-attacks are increasing fast in both the size and sophistication level, and they have become to a critical threat for current network infrastructure. Legacy security tools, for example firewalls and intrusion detection systems have limitations in being able to deeply scrutinize attacker behaviour. Honeypots are an efficient form of deception based defence, but work by luring attackers into supervised systems. This thesis proposed a network system of honeypot based security, which capable to x real-time attacks and mimic them. The system includes several honeypot sensors, automatic file-based threat intel processing, secure data storage and a user-interactive Flask web interface. Experimental results show improved attack visibility, reduced false positives, and enhanced situational awareness.**

I. INTRODUCTION

The exponential growth of Internet-connected systems has significantly increased the attack surface for cyber adversaries. Attackers exploit vulnerabilities using techniques such as brute- force authentication, port scanning, malware injection, and denial-of-service attacks. Conventional defense mechanisms primarily

focus on prevention and detection but provide limited insight into attacker strategies. Honeypots are designed to expose decoy services intentionally to attract attackers and log their activities in a controlled environment. This research, in turn, discusses the development of a honeypot detection and analysis system that will monitor the activities of attackers, gather threat intelligence, and provide actionable insights through visualization.Unlike traditional defenses, honeypots assume compromise and are designed to study attacker behavior rather than merely block it.

II. RELATED WORK

Numerous researchers have studied honeypot technologies for network security. Low- interaction honeypots simulate limited services, while high-interaction honeypots provide real operating systems for deeper attacker engagement. Recent studies explore geographic deployment, behavioral analysis, and machine learning integration to improve threat intelligence accuracy.

III. SYSTEM ARCHITECTURE

The system architecture shows how a honeypot is integrated into a secure network to monitor and analyze malicious activities. The incoming traffic from the internet has to pass, first, through an external firewall that will filter unauthorized accesses into the router. The router then directs the traffic to different segments of the network: the internal network, the service network, and honeypots. An IDS/IPS is placed at the point of segregating detection and prevention of known intrusion attempts, while suspicious traffic is allowed to pass for observation. The internal firewall further secures the critical systems by controlling the access between internal components. Honeypots are deployed at strategic locations both outside and inside the network to attract attackers away from real servers. The Service Network hosts essential services such as DNS, SMTP, and HTTP required for the support of normal operations. This layered architecture provides strong security but can effectively allow monitoring and analysis of attackers.

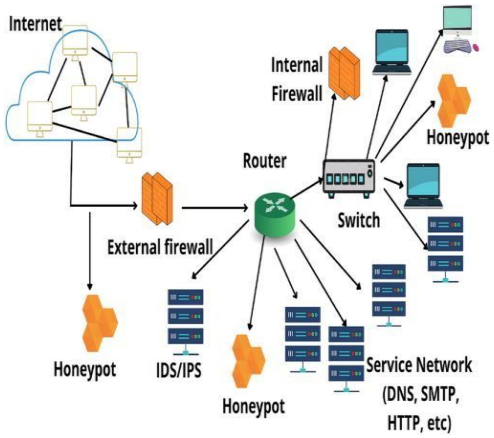

Figure 1: Honeypot-Based Layered Network Security Architecture

IV. METHODOLOGY

The methodology includes data collection, preprocessing, feature extraction, attack classification, and visualization. Extracted features include source IP, destination port, protocol, timestamp, attack type, and geographic information. Rule-based and statistical analysis techniques are applied to identify malicious behavior.

A. Data Collection

In honeypot system setup and deployment, there are multiple sensors that entice attackers and include HTTP honeypot, SSH honeypot, and port honeypot honeypot. All connection attempts and payloads are intercepted and recorded instantly.

This data points to actual malicious activities without endangering actual network resources. These services serve to lure attackers who might use unAuthorized access, Scans, or exploitation.

Each interaction recorded by the honeypot is automatically logged, which includes the IP address, port, protocol, and timestamp. Since there are no legitimate users interacting with honeypots, traffic obtained is classified as suspicious/malicious in nature.

B. Feature Extraction and Pre-processing

The raw attack logs are then processed to determine useful features such as IP address, port number, protocol type, timestamp, and type of attack. The data is cleaned to eliminate duplicate or invalid records.

Pre-processing of data is done in order to standardize the data and make it amenable for analysis. Invalid data points, duplicates, and incomplete data points are removed during pre-processing. The data points are made uniform during data normalization.

C. Attack Analysis and Classification

Analysts study the extracted data to recognize attack patterns like brute force attacks, port scanning, and exploitation.The extracted data is processed using rule-based methods and behavior analysis to categorize attack types.These approaches help to recognize attack types and understand attack intentions and techniques.Payload analysis can recognize exploitation attacks like SQL or command execution attacks.Rule-based approaches and behavior logic help to categorize attacks into relevant categories.These approaches help to understand attack intentions and techniques.

D. Data Storage

All processed attack events are recorded securely within an SQLite database.Data stored within the database is recorded in a structured format that is conducive for querying and analysis based on historical data.Long-term tracking and analysis based on the data is possible through the database.Tables within the database include attack events, dates and times, IP addresses, and attack types.Recording data within the database is useful for efficient querying and comparisons.The SQLite database is lightweight for faster performance with low system resources.

E. Visualization and Dashboard Generation

The collected information is analyzed through graphical representations on an online dashboard. The use of graphs like timeline, pie chart, and bar chart enables quick analysis of attack patterns. Graphs like line graphs enable analysis of attack patterns based on time. Pie graphs enable analysis of attack pattern distribution. Bar graphs enable analysis of attack patterns based on countries/ips. Graphical representations enable effective analysis of collected information. The online dashboard enables security administrators to get real-time situational awareness.

F. Monitoring and Reporting

The system keeps track of honeypot activity and updates the dashboard dynamically. Attack statistics and behaviors can be reported. This helps with decision-making for enhancing network security. The attack statistics and threat level are summarized in the report. Attack trends are also summarized. The attack level helps determine the effectiveness of implemented security. Enhancements can be made to network security based on knowledge acquisition.

V. IMPLEMENTATION DETAILS

It implements honeypot monitoring in a web- based application using Python and the Flask framework. Honeypot sensors, such as HTTP, SSH, and open ports, are deployed that will attract and capture malicious activities. All interaction by an attacker is logged and processed securely without having an impact on real systems. It provides user authentication and maintains restricted access to keep systems secure. Automatic analysis of log and threat files uploaded to extract IPs and details about various attacks was done. Data processing is stored in an SQLite database for efficient recovery and analysis. Interactive dashboards and charts present attack trends and statistics. The system deploys in an isolated environment and has been tested to ensure safe and reliable operation. A rule-based honeypot activity monitoring system applies rulesets on detected activities for finding patterns of brute force, port scanning, or exploitation attempts. If the rule condition is met, the system classifies the attack, logs it in the database, and displays it on the dashboard for further analysis.

1. System Architecture and Development

The honeypot monitoring system utilizes Python, since it is a language which is light, has a very large repertoire of libraries, and a strong community adoption as regards research into cybersecurity. Since there will be web communications with the users by handling user requests, authentication, data processing, and visualization, the Flask framework should be applied to build up a web application. The system's modular architecture allows separating data collection and analysis from the presentation layers by executing them on separate virtual machines, thus enhancing the maintainability and scalability of the proposed system.

2. Honeypot Sensor Configuration and Deployment

Various honeypot sensors, like HTTP, SSH, and open port services, are implemented to imitate vulnerable network services. All these sensors are configured to act like real systems so that attackers may get attracted. The deployment will be done in an isolated environment, not affecting production systems or sensitive data in case of malicious activity.

3. Secure Attack Data Collection

All interactions performed by attackers, such as login attempts, execution of commands, and port probing, are recorded in real time. The system guarantees that all information being logged is done in a secure manner that does not enable attackers to perform malicious tasks.

4. Authentication and Access Control Mechanism

For the protection of critical attack data, a user authentication system is incorporated with secure login details. It is impossible to gain access to the dashboard or log analysis details using the upload log feature with login credentials because only authorized persons have access to this system.

5. Log File Upload and Automated Processing

The system also facilitates a secure file upload mechanism for the logs and threats gathered from the honeypots. The files are then automatically processed to identify the core details like source IP, date, methods of attack, and number of attempts.

6. Database Design and Data Storage

Extracted attack data is stored in an SQLite database, chosen for its lightweight nature and ease of integration with Flask. The database schema is designed to efficiently store attack metadata, classifications, and timestamps. This structured storage allows fast querying and long- term attack trend analysis.

7. Rule-Based Attack Detection and Classification

In rule-based engines, there is a continuous analysis of the incoming honeypot data to check for known attacks. Rules are used to analyze the incoming data for brute force login attacks, port scan activity, and exploitation attacks. When a rule condition matches, an analysis of the attack with its storage in a database takes place.

8. Visualization and Dashboard Analytics

The system generates interactive dashboards and graphical charts to represent attack statistics visually. Metrics such as number of attacks, attack types, source IP distribution, and time- based trends are displayed. Visualization improves understanding of attack patterns and supports informed security decision-making.

9. Alerting and Monitoring Mechanism

When suspicious activity matches predefined attack rules, the system logs the event and highlights it on the dashboard. This allows security administrators to quickly identify ongoing threats. The monitoring mechanism supports continuous security awareness without manual supervision.

10. Testing, Evaluation, and Secure Deployment

The system is thoroughly tested using simulated attack scenarios to validate detection accuracy and reliability. Deployment is performed in a sandboxed or virtualized environment to ensure safety. This controlled setup confirms that the system operates effectively without introducing security risks.

VI. HONEYPOT SYSTEM USER INTERFACES

The system provides secure and user-friendly interfaces for managing honeypot activities. The login module ensures only authorized users can access the system.The file upload module allows users to efficiently submit log files for attack analysis.

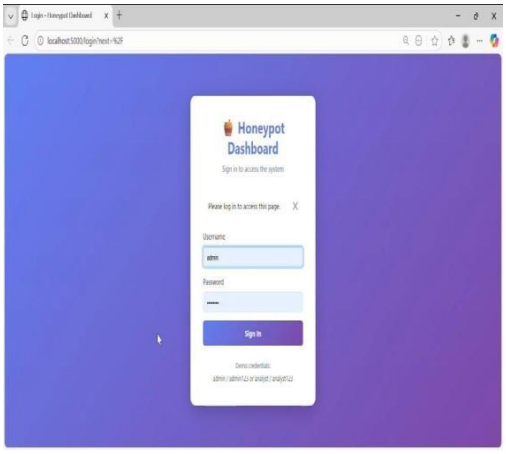

Figure 3: Honeypot Dashboard

Fig. 3: Login interface of the Honeypot Dashboard used for secure user authentication and access control to the system.It requires valid credentials to prevent unauthorized access to monitoring features.This ensures that only authorized users can view and manage honeypot data.

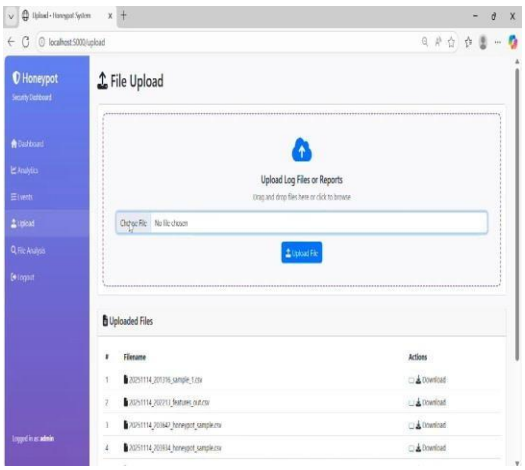

Figure 4: Upload Log Files or Report

Fig. 4: File upload interface of the Honeypot System used to upload log files and security reports for analysis.The module allows users to browse or drag-and-drop files securely into the system.Uploaded files are processed automatically to extract attack information and threat indicators.

VI. RESULTS AND DISCUSSION

More than a thousand attacks, ranging from brute force attacks to scanning, were recorded by this system. Visualization through interactive graphics made it possible to easily identify trends in attacks, attacker geographies, and targeted services.

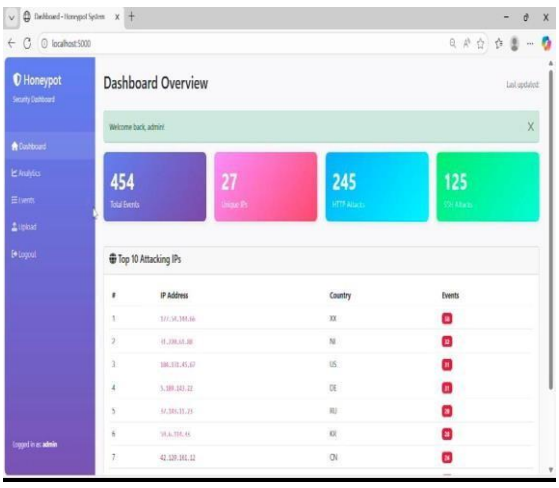

Figure 5: Attack Events and Attacking IPs

Fig.5: Attack events are instances where unauthorized users or automated scripts attempt to compromise the honeypot system.These events are captured in real-time by the honeypot and logged

for analysis. Each event includes details such as the attacker's IP address, time of attack, method used, and targeted services.

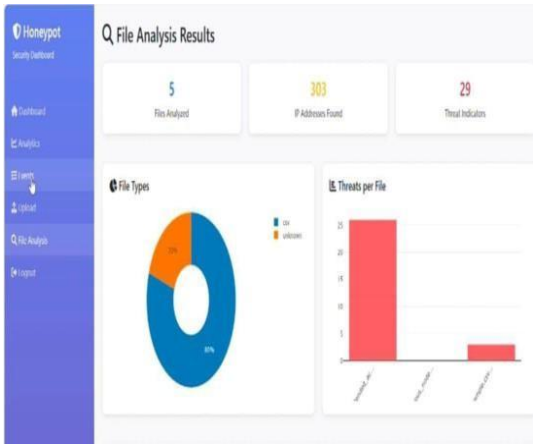

Figure 6: File Analysis Results

VII. ADVANTAGES

- It provides early attack detection, low false positives, and intelligence on threats. It may be detected by hackers and is a risk when properly isolated. This system provides early detections for attacks before affecting the production system.Improves upon false positives from conventional intrusion detection systems.
- Reduces false positives compared to traditional intrusion detection systems.
- Enhances threat intelligence by collecting detailed attacker behavior and methods.

Limitations of Honeypots in Network Security

- Skilled attackers may detect and avoid interacting with honeypots.
- Improperly isolated honeypots can become a security risk to the network.

VIII. CONCLUSION AND FUTURE WORK

Conclusion

The significance of honeypot systems is clearly brought out from this work as honeypot systems are established to be very effective monitoring tools for observing, detecting, and analyzing cyber-attacks. By functioning in an encapsulated atmosphere, honeypot systems securely record genuine attacker actions without risking production systems. This helps security managers measure attack methodologies such as brute-force attacks, scanning for ports, and attack strategies. The honeypot monitoring system designed in this work is very effective and works to enhance early attack warning systems for detecting attack actions at very early stages. The designed honeypot-based monitoring system helps to eliminate false alerts to a great extent because it uses rule-based systems that detect attacks based on suspicious action-patterns and not on normal network processes. The designed honeypot monitoring system is also helpful for improving threat intelligence as it helps to record attack actions with respect to source IP addresses.

Future Work

There are further improvements such as implementing real-time alert systems, like email alerts, to notify security administrators on serious attacks instantly. Other improvements, such as implementing more sophisticated logging and report systems, will enable security experts to stay informed about changing attack environments. All these improvements are set to improve the efficiency, flexibility, and intelligence of security systems using honeypots.

REFERENCES

[1] Yaser Alosefer and Omer Rana, "Honeyware: a web-based low interaction client honeypot", Third IEEE International Conference on Software Testing, Verification, and Validation Workshops (ICSTW), pp. 410 – 417, 2010.

[2] Xiaoyan Sun, Yang Wang, Jie Ren, Yuefei Zhu and Shengli Liu, "Collecting Internet Malware Based on Client-side Honeypot", 9th IEEE International Conference for Young Computer Scientists (ICVCS 2008), pp. 1493 – 1498, 2008.

[3] C. H. Nick Jap, P. Blanchfield, and K. S. Daniel Su, "The use of honeypot approach in softwarebased application protection for shareware programs", IEEE International Conference on Computing & Informatics, (ICOCI '06), pp. 1-7, 2006.

[4] Jian Bao and Chang-peng Ji, and Mo Gao, "Research on network security of defense based on Honeypot", IEEE International Conference on Computer Application and System Modeling (ICCASM), vol. 10, pp. V10-299 - V10-302, 2010.

[5] Anjali Sardana, R. C. Joshi, "Honeypot Based Routing to Mitigate DDoS Attacks on Servers at ISP Level", IEEE International Symposiums

on Information Processing (ISIP), pp. 505-509, 2008.

[6] Guanlin Chen, Hui Yao "Research of Wireless Intrusion Prevention Systems based on Plan Recognition and Honeypot", IEEE International Conference on Wireless Communications & Signal Processing (WCSP), pp. 1-5, 2009.

[7] Chao-Hsi Yeh and Chung-Huang Yang, "Design and Implementation of Honeypot Systems Based on Open-Source Software", IEEE International Conference on Intelligence and Security Informatics (ISI), 265-266, 2008.

[8] Babak Khosravifar, Maziar Gomrokchi, Jamal Bentahar, "A Multi-Agent-based Approach to Improve Intrusion Detection Systems False Alarm Ratio by Using Honeypot", IEEE International Conference on Advanced Information Networking and Applications Workshops, pp. 97 – 102, 2009.

[9] Wei Li-feng, Wang Xiao-bin, "Research on Honeypot Information Fusion Based on Game Theory", Second IEEE International Conference on Computer Research and Development, pp. 803 – 806, 2010.

[10] Xinliang Wang, Fang Liu, LuYing,"Research for Scan Detection Algorithm of High-Speed Links Based on Honeypot", 2nd IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC), pp. 66-70, 2010.