International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 03 | Mar 2025 www.irjet.net p-ISSN: 2395-0072

# Securing Network using Honeypot & Detecting Malicious IP Addresses

**Aakash Gojare[1], Prof. Rajesh Bansode[2]**

[1]*Student, Thakur College of Engineering & Technology, Maharashtra, India*
[2]*Professor, Thakur College of Engineering & Technology, Maharashtra, India*

---

**Abstract -** *In today's digital landscape, securing networks against cyber threats is of paramount importance. The project titled "Securing Network Using Honeypot and Detecting Malicious IP Addresses" presents a comprehensive solution to network security challenges. This Flask-based application integrates multiple features to ensure real-time monitoring, threat detection, and response, making it an essential tool for modern cybersecurity.*

*The application incorporates functionality to retrieve the server's local IP address, resolve domain names to their corresponding IPs, and fetch geolocation details of IP addresses. By integrating with the honeydb.io API, it leverages honeypot intelligence to check whether a given IP is flagged as malicious. The Scapy library is employed for monitoring network traffic and detecting suspicious activity based on configurable parameters such as packet thresholds. Upon identifying malicious IPs, they are dynamically blocked using Windows netsh commands, and administrators receive real-time email alerts via the yagmail library.*

*Designed for ease of use, the application allows users to define monitoring parameters, block durations, and trusted IPs through a simple interface. Key technologies utilized include Flask for backend development, Scapy for traffic analysis, yagmail for email notifications, and netsh for IP blocking. Hosted on GitHub and Render, and developed using Anaconda, the project addresses both current and emerging cybersecurity challenges, offering a robust framework for securing networks.*

***Key Words***: Honeypot, Cybersecurity, Flask Application, Malicious IP Detection, Network Security, Real-time Monitoring, Threat.

## 1. INTRODUCTION

The exponential growth of digital technologies has significantly increased the complexity and scale of network infrastructures. With this growth comes a surge in cyber threats, making network security a top priority for organizations worldwide. Cyberattacks such as malicious IP activities pose severe risks, ranging from data breaches to operational disruptions. Traditional security measures often fall short in providing real-time detection and mitigation of such threats, necessitating innovative and comprehensive solutions.

This project, "Securing Network Using Honeypot and Detecting Malicious IP Addresses," addresses these challenges by integrating advanced network monitoring and threat prevention mechanisms. By leveraging honeypot intelligence from the honeydb.io API, the application identifies malicious IP addresses and prevents potential attacks. The inclusion of real-time network traffic analysis through the Scapy library ensures early detection of abnormal activity, which is indicative of malicious behavior. Malicious IPs are dynamically blocked using Windows netsh commands, minimizing the attack's impact and securing the network.

The project's user-friendly design empowers administrators to customize monitoring parameters and receive timely alerts through email notifications. Static HTML interfaces provide an intuitive and lightweight user experience, making the tool accessible for users with varying levels of technical expertise. Key technologies such as Flask for backend development, Render for deployment, and Anaconda for development environment management further enhance the project's reliability and scalability.

Through its proactive and multifaceted approach, this project offers a significant advancement in network security. It addresses the limitations of traditional methods, ensuring network integrity and reducing the risk of cyberattacks. The integration of honeypot intelligence and malicious IP detection sets a new standard in cybersecurity, making this project a vital resource for organizations aiming to safeguard their digital assets.

## 2. RELATED WORK

M. Kharrazi et al., "A Survey of Honeypot Techniques in Cybersecurity,"[1] This paper provides a comprehensive overview of honeypot techniques, including low-interaction, high-interaction, and hybrid honeypots. It highlights their differences in complexity, interaction levels, and use cases. Advantages, such as cost-effectiveness and detailed threat analysis, and limitations, including resource demands and detection risks, are discussed. The authors emphasize selecting honeypot types based on organizational security needs.

H. Haslum and T. Fray, "The Use of Honeypots for Intrusion Detection in Industrial Control Systems,"[2] This study examines honeypot deployment in Industrial Control Systems (ICS), addressing challenges such as mimicking ICS protocols and ensuring minimal disruption. Case studies demonstrate honeypots' effectiveness in detecting malware and

unauthorized access, emphasizing their role in enhancing ICS security when carefully designed and managed.

J. Levine et al., "Advanced Honeypot Architecture for Network Threat Detection,"[3] This paper introduces a multi-layered honeypot architecture for enhanced threat detection. The architecture employs low-interaction honeypots for initial probes and high-interaction honeypots for detailed analysis. Evaluation results show improved detection rates and insights into attack methodologies, making the architecture adaptable to various network environments.

M. S. Hossain et al., "A Comparative Study of Honeypot Tools for Malware Analysis,"[4] This paper compares honeypot tools, including Dionaea, Honeyd, and Kippo, based on deployment ease, malware capture effectiveness, and interaction levels. Each tool's strengths, such as Dionaea's wide malware capture and Honeyd's low resource usage, are highlighted. The authors recommend selecting tools based on specific use cases.

P. Spathoulas and S. K. Katsikas, "Dynamic Honeypot Deployment for Autonomous Security,"[5] This research proposes a dynamic honeypot framework for autonomous security management. Using machine learning, honeypots are deployed based on real-time network conditions. Experimental results demonstrate improved threat detection and reduced overhead compared to static setups, highlighting the framework's adaptability and resilience.

K. G. Kyriakopoulos et al., "High-Interaction Honeypots: Design and Implementation Challenges,"[6] This paper discusses the challenges of high-interaction honeypots, including realistic system behavior, data volume management, and security maintenance. The authors provide guidelines for overcoming technical and operational difficulties, emphasizing the importance of robust design and maintenance for effective deployment.

L. Spitzner, "Honeypots and Honeynets: Analysis and Case Studies,"[7] This paper explores honeypots and honeynets, supported by real-world case studies. Honeynets, consisting of multiple networked honeypots, offer insights into coordinated attacks. Case studies demonstrate their effectiveness in detecting botnets and APTs, underscoring their value in threat intelligence and network security.

D. M. E. Inácio et al., "Effective Use of Honeypots for Cybersecurity Training,"[8] This paper highlights the use of honeypots in cybersecurity training. By simulating real-world attack scenarios, honeypots provide hands-on learning experiences. Results from training sessions show improved understanding of attack techniques and defensive strategies, demonstrating honeypots' value in practical education.

A. C. Clark et al., "HoneyBot: A Novel Approach to Botnet Detection Using Honeypots,"[9] This paper introduces HoneyBot, a honeypot system designed for botnet detection. Combining low and high-interaction honeypots with machine learning, HoneyBot effectively identifies botnet behaviors. Evaluation results demonstrate its efficacy in detecting command-and-control communications and propagation techniques.

S. L. Garfinkel, "Legal and Ethical Issues in Honeypot Deployment," [10] This paper addresses legal and ethical concerns in honeypot deployment, such as entrapment, privacy violations, and data collection. The author provides guidelines for responsible deployment, emphasizing transparency, consent, and legal compliance to avoid potential pitfalls.

## 3. PROPOSED METHODOLOGY

The proposed system integrates advanced network monitoring and threat mitigation features to enhance cybersecurity. The application's architecture consists of the following components:

1. Local IP Retrieval and Domain Resolution: The application retrieves the server's local IP address and resolves domain names to IPs, providing essential network details.

2. Honeypot Intelligence Integration: Using the honeydb.io API, the system checks IP addresses against a database of flagged malicious IPs.

3. Network Traffic Monitoring: The Scapy library monitors real-time network traffic, analyzing packets to detect suspicious activity based on customizable thresholds.

4. Dynamic IP Blocking: Identified malicious IPs are blocked dynamically using Windows netsh commands to prevent further network access.

5. Real-Time Alerts: The yagmail library enables email notifications, ensuring administrators are informed of security events promptly.

6. User Interface: Static HTML files provide a lightweight and intuitive interface for configuring monitoring parameters, block durations, and trusted IPs.

7. Deployment and Scalability: The application is hosted on Render and version-controlled on GitHub, ensuring accessibility and continuous updates.

## 4. WORKING

The system begins by initializing and retrieving essential network details, such as the server's local IP address and resolving domain names to IPs. This foundational step ensures that the application is aware of its environment and ready to monitor incoming and outgoing traffic. Using the Scapy library, the application continuously analyzes real-time network traffic, focusing on packet data to detect

irregularities or anomalies that may signal potential threats, such as suspicious IP activities.

When an IP address is detected, it is cross-verified with honeypot intelligence from the honeydb.io API. This integration allows the application to check if the IP has been flagged as malicious in a global database. If an IP is identified as malicious or suspicious, the application takes immediate action by dynamically blocking the IP using Windows netsh commands. This process ensures that the IP cannot access or disrupt the network further.

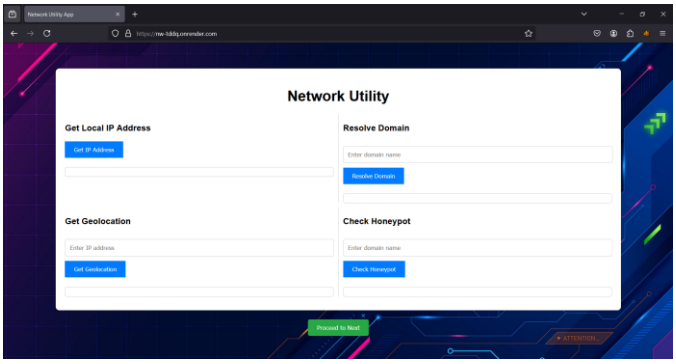


**Fig -1**: Network Utility

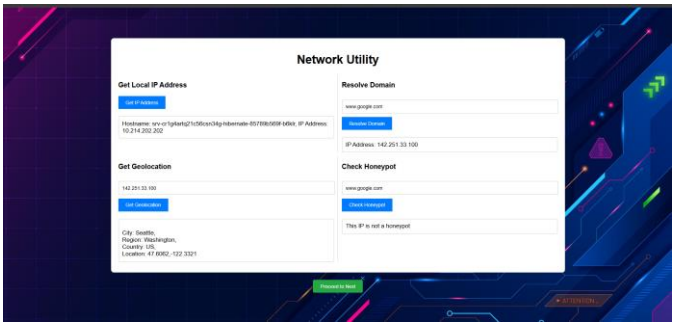


**Fig -2**: Network Utility Detected Output

To keep administrators informed, the application generates real-time email alerts using the yagmail library. These alerts contain detailed information about the detected threat and the actions taken, enabling prompt follow-up. Additionally, the application's interface allows administrators to customize monitoring parameters, such as setting packet thresholds, defining block durations, and managing trusted IPs. The user interface, designed with static HTML files, provides an intuitive and lightweight experience, making the application accessible even to users with limited technical expertise.

The application is hosted on Render for deployment and version-controlled on GitHub, ensuring it is both scalable and reliable. Regular updates can be seamlessly integrated, making the tool adaptable to evolving cybersecurity needs. Overall, the system's working mechanism ensures a robust, real-time, and user-friendly approach to detecting and mitigating network threats.

## 5. CONCLUSION

In conclusion, the project "Securing Network Using Honeypot and Detecting Malicious IP Addresses" offers a comprehensive and adaptable solution to modern cybersecurity challenges. By integrating honeypot intelligence, real-time threat detection, and malicious IP identification, the Flask-based application ensures robust network protection. Key functionalities such as geolocation tracking, malicious IP detection through honeydb.io, and dynamic IP blocking using netsh empower organizations to respond effectively to evolving cyber threats. The use of Scapy for traffic analysis and real-time email alerts further enhances its proactive security framework. Designed for simplicity and user accessibility, the application provides configurable monitoring and an intuitive interface. Built with technologies like Flask, Scapy, and yagmail, and hosted on platforms such as GitHub and Render, this project demonstrates a forward-thinking approach to securing digital infrastructures against emerging cyber risks.

Looking ahead, the project has significant potential for future enhancements. These include integrating cloud-based honeypot systems for improved threat intelligence, supporting cross-platform IP blocking mechanisms, and incorporating machine learning for anomaly detection and predictive analytics. Additionally, the development of real-time visualization dashboards for monitoring network activity and expanding support for multiple operating systems and diverse deployment environments will further enhance its versatility and effectiveness in addressing advanced cybersecurity needs.

## ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my guide, Prof. Rajesh Bansode, for his valuable guidance and support throughout this project. I also extend my thanks to my institution Thakur College of Engineering & Technology and faculty for providing resources and encouragement. Special thanks to my peers and family for their motivation and understanding.

## REFERENCES

[1] M. Kharrazi, et al., "A Survey of Honeypot Techniques in Cybersecurity," Journal of Cybersecurity and Privacy, vol. 2, no. 1, pp. 15–30, Jan. 2021.

[2] H. Haslum and T. Fray, "The Use of Honeypots for Intrusion Detection in Industrial Control Systems," IEEE Transactions on Industrial Informatics, vol. 15, no. 7, pp. 3883–3892, Jul. 2019.

[3] J. Levine, et al., "Advanced Honeypot Architecture for Network Threat Detection," IEEE Communications Magazine, vol. 58, no. 5, pp. 42–49, May 2020.

[4] M. S. Hossain, et al., "A Comparative Study of Honeypot Tools for Malware Analysis," International Journal of Network Security, vol. 22, no. 2, pp. 190–200, Mar. 2020.

[5] P. Spathoulas and S. K. Katsikas, "Dynamic Honeypot Deployment for Autonomous Security," IEEE Systems Journal, vol. 14, no. 2, pp. 2053–2062, Jun. 2020.

[6] K. G. Kyriakopoulos, et al., "High-Interaction Honeypots: Design and Implementation Challenges," IEEE Transactions on Information Forensics and Security, vol. 14, no. 6, pp. 1506–1517, Jun. 2019.

[7] L. Spitzner, "Honeypots and Honeynets: Analysis and Case Studies," IEEE Security & Privacy, vol. 17, no. 5, pp. 50–57, Sep. 2019.

[8] D. M. E. Inácio, et al., "Effective Use of Honeypots for Cybersecurity Training," IEEE Transactions on Education, vol. 62, no. 3, pp. 208–215, Aug. 2019.

[9] A. C. Clark, et al., "HoneyBot: A Novel Approach to Botnet Detection Using Honeypots," IEEE Access, vol. 8, pp. 12400–12411, Jan. 2020.

[10] S. L. Garfinkel, "Legal and Ethical Issues in Honeypot Deployment," IEEE Security & Privacy, vol. 16, no. 6, pp. 79–85, Nov. 2018.