Indian Journal of Computer Science and Technology
https://www.doi.org/10.59256/indjcst.20250401024
Volume 4, Issue1 (January-April 2025), PP: 161-165.
www.indjcst.com
ISSN No: 2583-5300

Implementation of Dynamic ARP Inspection for Enhanced Network Security

Pranav R Pillai[1], C. Vishnu Priya[2]
[1]M. sc., CFIS, Department of Computer Science and Engineering, Dr. MGR University, Chennai, Tamilnadu, India.
[2]Assistant Professor, Cyber Forensics and Information Security, IDE, University of Madras, Chepauk. Chennai, Tamilnadu, India.

To Cite this Article: *Pranav R Pillai[1], C. Vishnu Priya[2], "Implementation of Dynamic ARP Inspection for Enhanced Network Security", Indian Journal of Computer Science and Technology, Volume 04, Issue 01 (January-April 2025), PP: 161-165.*

Abstract: *This study focuses on enhancing network security by implementing dynamic ARP (Address Resolution Protocol) inspection to detect and mitigate ARP spoofing attacks. The solution is built using Flask, a lightweight web framework, and Scapy, a powerful library for packet manipulation. The system monitors network traffic in real time by analyzing ARP packets, identifying inconsistencies, and flagging potential spoofing attempts. It incorporates database integration through SQLite to manage allowed and blocked IP addresses, ensuring efficient attack response. The application also features a web interface that enables administrators to control monitoring operations, view alerts, and manage IP lists. Furthermore, the system integrates with external services like VirusTotal and ipinfo.io, providing detailed information about suspicious IPs. This comprehensive approach strengthens network security by detecting ARP spoofing attempts and preventing unauthorized access.*

Keywords: *ARP [Adress resolution protocol], Dynamic ARP Inspection [DAI], Scapy, IP Whitelisting and Monitoring, Spoofing Detection, Fire wall Rules*

I.INTRODUCTION

In modern networked environments, maintaining the integrity and security of communication is essential. Among the various types of cyber threats, Address Resolution Protocol (ARP) spoofing remains one of the most prevalent and disruptive. This deceptive strategy allows attackers to intercept, alter, or block network traffic between hosts, making it a serious threat in both wired and wireless networks.

To combat such vulnerabilities, the study introduces a real-time ARP spoofing detection and mitigation system built using Python. The application utilizes the Scapy library for low-level packet manipulation and the Flask framework to provide an accessible webbased user interface [1]. Through the continuous monitoring of ARP packets, the system identifies abnormal behaviors in MAC-toIP relationships. By comparing incoming ARP replies with known valid entries and verifying suspicious activity with active TCP handshake methods, the application is capable of detecting spoofing attempts with high accuracy.

Additionally, the system features a database-backed structure to manage trusted and blocked IP addresses, ensuring that responses to threats are both efficient and persistent. Firewall rules are automatically applied to isolate malicious devices, while trusted IPs are safeguarded against false detections [2]. The integration of external APIs like VirusTotal and ipinfo.io further enhances the system's capability by providing additional context about IP addresses involved in suspicious activity. Overall, the application aims to offer a lightweight, yet effective, approach to securing local networks from ARP spoofing attacks.

With the growing complexity of cyber threats, traditional security mechanisms like antivirus software and firewalls alone are no longer sufficient to protect against internal network attacks such as ARP spoofing. These attacks often occur within local area networks (LANs), bypassing perimeter defenses and targeting devices that rely on ARP for communication. This emphasizes the need for specialized tools capable of inspecting network-level traffic in real-time. The proposed system addresses this gap by offering a lightweight, Python-based solution that not only detects spoofing attempts but also takes automated preventive actions. Its modular design, combining network packet analysis, IP reputation services, and administrative control, ensures both scalability and practical usability across different network environments [3].

II.LITERATURE REVIEW

Mohanta, B. K., Sahu, S. K., & Panda, S et.al [4] had proposed a review on detection and prevention of ARP spoofing attacks.This paper offers a comprehensive overview of ARP spoofing attacks and discusses various detection mechanisms, such as passive packet monitoring, static ARP tables, and dynamic verification techniques. It emphasizes the importance of tracking MAC-IP bindings and detecting inconsistencies, which directly supports your system's use of IP_MAC_PAIRS and ARP_REQ_TABLE for spoof detection. The paper also highlights the limitations of traditional security tools against ARP-based attacks, reinforcing the need for specialized tools like yours.

Tripathi, S., & Chaurasia, B. K et.al [5] had proposed a ARP poisoning attack and its countermeasures: A survey. This survey outlines both the attack vectors and countermeasures for ARP poisoning in local networks. It introduces methods such as MAC

filtering, dynamic inspection, and active probing. Your system incorporates similar ideas by sending TCP SYN packets to suspected IPs and waiting for valid responses to confirm spoofing. This paper supports your project's proactive validation method and adds theoretical backing to your decision to use real-time traffic inspection.

Chatterjee, S., & Misra, S et.al [6] had proposed an intrusion detection system for ARP spoofing using active MAC address verification. The authors propose an intrusion detection system that leverages MAC verification against known IP addresses to identify spoofing attempts. It uses an active approach, similar to the TCP verification method in your system, where a handshake mechanism validates the authenticity of ARP replies. This directly aligns with your spoof_detection() function and enhances its reliability by reducing false positives.

Kar, P., & Bharati, R. et.al [7] had proposed detection and mitigation of ARP spoofing using active probing and MAC verification. This study presents a system that uses active probing and MAC address consistency checks to detect ARP spoofing. It maintains a known MAC-IP table, similar to your IP_MAC_PAIRS, and verifies packets using active scanning methods. The similarity in detection logic—validating suspicious replies using TCP SYN or ARP requests—makes this paper highly relevant to your approach.

Bhandari, R., & Lamba, A. et.al [8] had proposed enhancing LAN security by implementing ARP inspection using Python.The paper demonstrates how Python, particularly using Scapy, can be employed to build a simple ARP monitoring system. It validates the feasibility of your approach to use Python and Scapy for sniffing and parsing ARP packets. The authors focus on developing a lightweight solution that can be implemented in small to medium-sized LANs, which mirrors the architecture and scalability of your uploaded system.

Dhanalakshmi, S., & Kannammal, E. et.al [9] had proposed Prevention of ARP spoofing attacks using virtual agents.This paper explores a rule-based defense mechanism using intelligent agents to automate the detection and prevention of ARP spoofing. While your system does not use agents, it achieves similar automation by implementing firewall rules through subprocess commands. The concept of reducing manual intervention and dynamically applying mitigation actions aligns closely with your use of api_block_ip() and netsh command integration.

Kumar, R., & Manjusha, N et.al [10] had proposed Real-time detection and prevention of ARP spoofing attack using IDS.This work presents an intrusion detection system that monitors ARP traffic in real-time and alerts users upon detecting spoofing attempts. It supports your use of real-time alert generation, logging suspicious activity, and providing a dashboard for administrators to take action. The IDS model in the paper serves as a theoretical foundation for your application's architecture, especially the alerting and response mechanisms.

III.PROPOSED METHODOLOGY

The developed system adopts a proactive strategy to identify and counter ARP spoofing attacks in real-time. It leverages the Scapy library to monitor network traffic by capturing ARP packets and distinguishing between ARP requests and replies. When a request is sent, the system records the destination IP along with the time it was observed. Upon receiving a reply, it examines whether the MAC address matches the expected value for the corresponding IP address. Any deviation suggests suspicious behavior. To further confirm such anomalies, the system initiates a TCP SYN handshake with the suspect IP[11]. If a valid acknowledgment is not received, the system flags the event as a spoofing attempt and raises an alert.

To streamline threat management, the system incorporates a local database to classify and store IP addresses. Two primary tables are maintained: one for trusted IPs and another for potentially harmful ones. IPs identified as attackers are logged in the blocked list and are prevented from further interaction through automated firewall rules. In contrast, legitimate IPs can be whitelisted to avoid false alarms[12]. This classification helps ensure accurate detection and reduces unnecessary interference with normal network traffic. Administrators have the flexibility to update these lists through the system interface, enhancing control and adaptability.

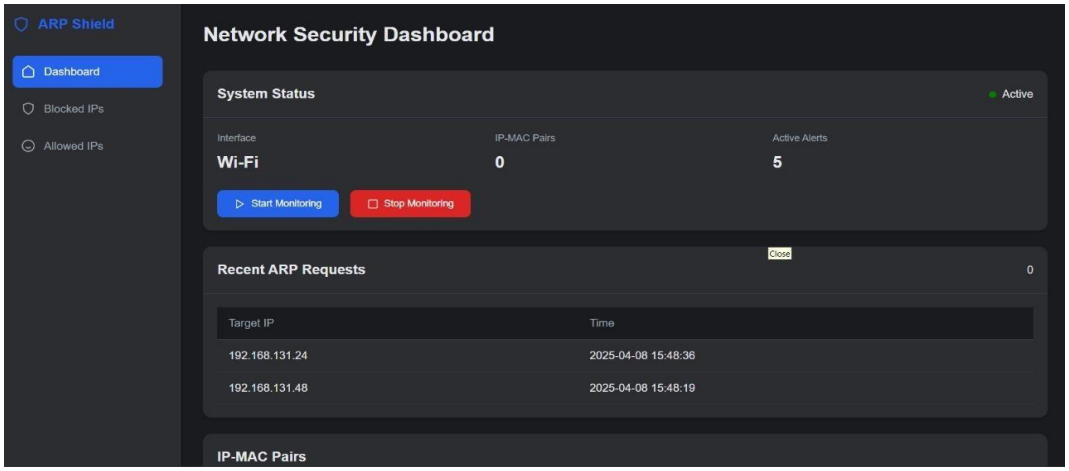

Fig 3.1: Monitoring target IP

An easy-to-use web interface built using Flask allows users to manage the monitoring operations efficiently. Administrators can start or stop the ARP monitoring process, review detection logs, and manage IP access permissions directly from the dashboard. The system also integrates external services like VirusTotal and ipinfo.io to retrieve in-depth details about suspicious IP addresses, such as geographic origin, ownership, and known threat records[13]. These external insights help in

making better security decisions. By combining live monitoring, intelligent detection techniques, database-driven IP control, and external threat intelligence, the system provides a comprehensive and robust solution against ARP spoofing attacks.

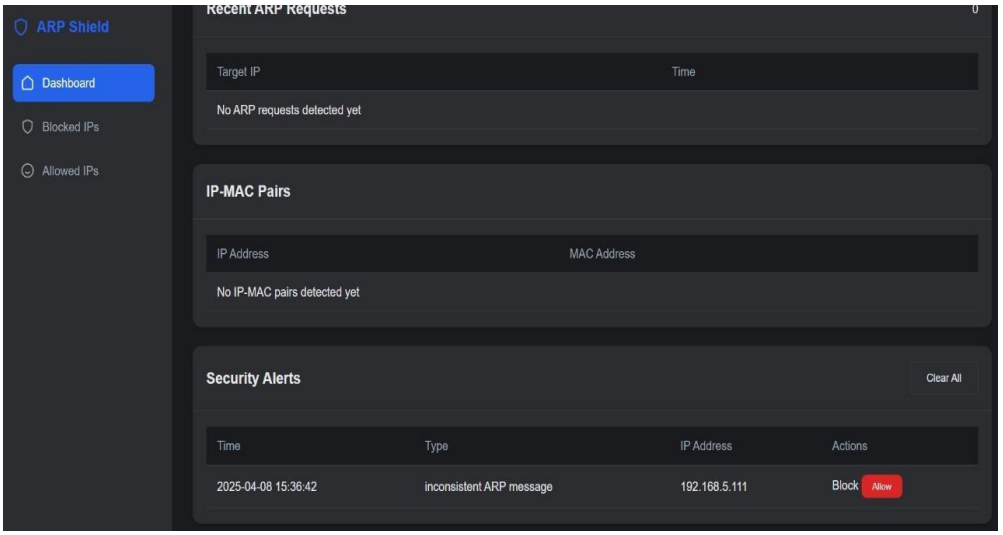

Fig 3.2: Detecting Target IP

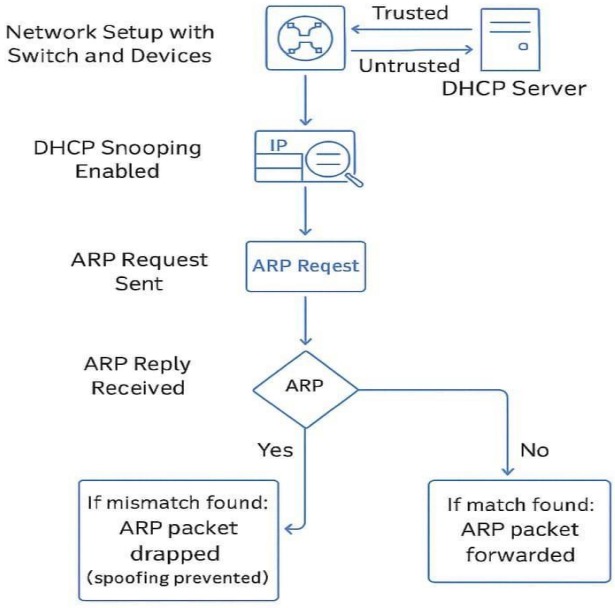

Fig 3.3: System Architecture

IV.FINDINGS

The implemented ARP spoofing detection system effectively monitors and analyzes ARP traffic within a local network, allowing real-time identification of malicious activity. By utilizing Scapy for packet sniffing and comparing ARP requests and replies, the system accurately detects inconsistencies in IP-MAC mappings, which are key indicators of spoofing attempts. The detection mechanism, supported by validation through TCP handshake responses, ensures that false positives are minimized and only verified threats trigger alerts [14].

Additionally, the system provides an efficient method for managing network security through IP classification. With the integration of a SQLite database, the application maintains organized lists of allowed and blocked IP addresses. This helps automate the response to attacks by immediately applying firewall rules to block suspicious IPs, while trusted devices are excluded from detection processes [15]. This structured approach ensures that network traffic remains secure without disrupting legitimate communication.

The inclusion of a web-based interface and third-party API integration significantly enhances the system's usability and reliability. The Flask-based dashboard allows administrators to control monitoring operations, view alerts, and inspect IPs with ease. External services like VirusTotal and ipinfo.io enrich the system by providing additional data about potential threats, aiding in better decisionmaking. Overall, the system demonstrates a practical and comprehensive solution for detecting, analyzing, and mitigating ARP spoofing attacks in real time.

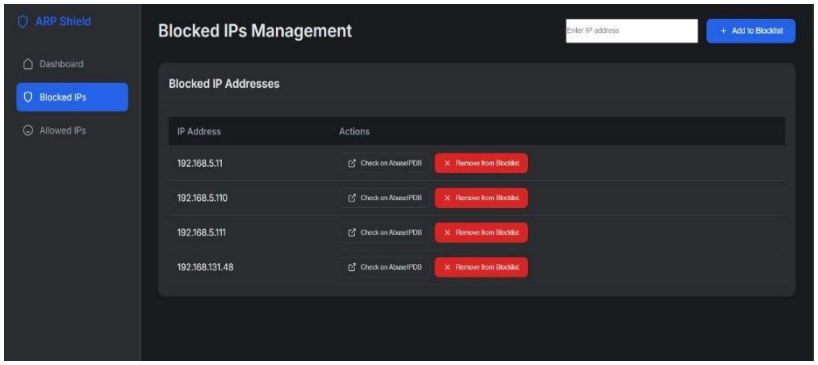

Fig 4.1: Blocked IP's

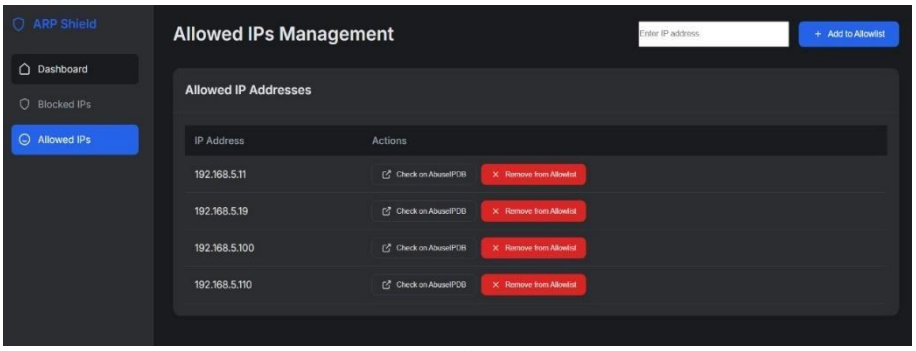

Fig 4.2: Allowed IP'S

V.CONCLUSION

In conclusion, the research outlined in this paper has successfully demonstrated the design and implementation of a lightweight, real-time ARP spoofing detection and prevention system. By utilizing the Scapy library for packet analysis and monitoring, the application efficiently captures and examines ARP traffic to identify irregularities in IP-MAC associations. The system's ability to cross-verify ARP responses using active TCP handshake validation adds a reliable layer of confirmation, reducing the risk of false detections and enhancing overall accuracy.The integration of a local database to categorize IP addresses as either trusted or malicious allows for streamlined and automated responses to identified threats. Suspicious IP addresses are not only flagged but are also actively blocked through firewall rules, while trusted devices are protected from unnecessary disruption. The administrative web interface built using Flask enhances usability by allowing users to initiate monitoring, review threat alerts, and manage IP access through an intuitive dashboard.

Moreover, external tools such as VirusTotal and ipinfo.io contribute to the system's robustness by providing contextual data about potentially harmful IP addresses. These additional insights support better decision-making and reinforce the system's defensive capabilities. Overall, the proposed solution effectively addresses ARP spoofing vulnerabilities in local networks and can serve as a foundational framework for future enhancements in network security tools.

References

1. *Scapy. (n.d.). Scapy Documentation. Retrieved from https://scapy.readthedocs.io/en/latest/*
2. *Flask. (n.d.). Flask Web Framework Documentation. Retrieved from https://flask.palletsprojects.com/*
3. *VirusTotal. (n.d.). VirusTotal API Documentation. Retrieved from https://developers.virustotal.com Mohanta, B. K., Sahu, S. K., & Panda, S. (2020)*
4. *A review on detection and prevention of ARP spoofing attacks. Journal of Cyber Security Technology, 4(1), 1–16. https://doi.org/10.1080/23742917.2020.1719141*
5. *Tripathi, S., & Chaurasia, B. K. (2022) ARP poisoning attack and its countermeasures: A survey. International Journal of Network Security, 24(2), 239–250. https://doi.org/10.6633/IJNS.202203.24(2).11*
6. *Chatterjee, S., & Misra, S. (2019).An intrusion detection system for ARP spoofing using active MAC address verification. Computer Standards & Interfaces, 64, 94–105. Https://doi.org/10.1016/j.csi.2018.11.002*
7. *Kar, P., & Bharati, R. (2019). Detection and mitigation of ARP spoofing using active probing and MAC verification. Procedia Computer Science, 167, 1194– 1203 https://doi.org/10.1016/j.procs.2020.03.426*
8. *Bhandari, R., & Lamba, A. (2018).Enhancing LAN security by implementing ARP inspection using Python. International Journal of Computer Applications, 180(27), 1–4.https://doi.org/10.5120/ijca2018917394*
9. *Dhanalakshmi, S., & Kannammal, E. (2016).Prevention of ARP spoofing attacks using virtual agents. Procedia Computer Science, 85, 13–20. https://doi.org/10.1016/j.procs.2016.05.174*
10. *Kumar, R., & Manjusha, N. (2017).Real-time detection and prevention of ARP spoofing attack using IDS. International Journal of Engineering & Technology, 9(3), 484–491.https://doi.org/10.14419/ijet.v9i3.19699.*

11. *Parulkar, G., & Choudhary, M. (2021). ARP spoof detection using packet analysis and monitoring techniques. International Journal of Advanced Research in Computer and Communication Engineering, 10(4), 45–50.https://ijarcce.com/wp-content/uploads/2021/04/IJARCCE.2021.10409.pdf*
12. *SQLAlchemy. (n.d.). SQLAlchemy Documentation. Retrieved from https://docs.sqlalchemy.org*
13. *Patil, A., & Gawande, R. (2020). Detection and prevention of ARP spoofing using Python and Scapy. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 6(2), 101–105.https://ijsrcseit.com/paper/2020/6/2/1204.pdf*
14. *Pooja, A., & Ranjan, S. (2019). Evaluating the effectiveness of real-time network monitoring for ARP spoofing attacks. International Journal of Computer Sciences and Engineering, 7(12), 65–70.https://www.ijcseonline.org/pdf_paper_view.php?paper_id=5247&14-IJCSE-07122.pdf*
15. *Jain, R., & Sharma, S. (2021). Real-time intrusion detection and network traffic analysis: A review. International Journal of Engineering Research & Technology (IJERT), 10(5), 200–205.https://www.ijert.org/research/real-time-intrusion-detection-and-network-traffic-analysis-IJERTV10IS050125.pdf*