Enhancing Network Security: ARP Spoofing Detection and Mitigation Using Ettercap and SSH-Based Defenses

Swati V[1], Penumarthi Hima Varshini[2], D Navya[3], and Vishwas H.N.[4]

Department of Computer Science and Engineering,
Amrita School of Computing, Bengaluru,
Amrita Vishwa Vidyapeetham, India
[1]bl.en.u4aie23031@bl.students.amrita.edu
[2]bl.en.u4aie23046@bl.students.amrita.edu
[3]bl.en.u4aie23050@bl.students.amrita.edu
[4]hn_vishwas@blr.amrita.edu

Abstract. ARP is a key protocol in IPv4 networks, mapping 32-bit IP addresses to MAC addresses for local communication, but its lack of authentication renders it vulnerable to ARP spoofing. The study presents a practical demonstration of ARP spoofing using Ettercap in a controlled environment, where victim ARP tables were successfully manipulated to redirect network traffic. The experimental results confirm the feasibility of such attacks by capturing sensitive data and highlighting the associated risks. To mitigate these vulnerabilities, an SSH-based defense strategy was implemented to ensure secure data transmission between devices. Our findings were proved by encrypted traffic logs which demonstrate that SSH effectively prevents unauthorized packet interception during file transfers. This study underscores the importance of incorporating encryption protocols to enhance network security.

Keywords: ARP, ARP Spoofing, ARP Poisoning, Network Security, LAN, Penetration Testing, Prevention

1 Introduction

The ARP (Address Resolution Protocol) is essential to IPv4 networks. It enables devices to resolve IP addresses to MAC addresses for data link layer communication. Although ARP is an essential protocol to the functionality of a network, it is insecure and susceptible to manipulation through ARP spoofing. An attacker can impersonate another device on the network, and this may lead to man-in-the-middle attacks, data interception, and unauthorized access to sensitive information.

ARP spoofing is a significant cyber threat since it does not rely on exploiting software vulnerabilities but rather manipulates the ARP protocol itself. The paper aims to understand what ARP spoofing is, demonstrate its implementation, and discuss effective mitigation strategies.

Various research work has been conducted to investigate detection and mitigation methods with conventional tools as well as advanced machine learning methods. Wireshark and Ettercap are leading software used in network traffic analysis. Wireshark can be used in the in-depth analysis of packets, while Ettercap can effectively identify ARP poisoning. Authors in [5] show the efficacy of these tools in network traffic analysis and ARP attack detection using pattern analysis. Their research, however, only addresses ARP poisoning attacks and not other network attacks, thus the need for a comprehensive approach.

Intrusion Detection Systems (IDS) are essential to identify unauthorized access and malicious activity. The research in [1] improves IDS performance by employing machine learning algorithms with explainable AI techniques. Eight machine learning models were tested on a military network dataset of 41 features and 22,544 records. Random Forest worked best, and LIME analysis provided interpretability to model decisions. These methods must be more flexible to respond to real-time threats.

Growing sophistication of cyber attacks necessitates more protection measures. Traditional security measures have been unable to keep pace with emerging attacks [2]. Deep learning models such as Convolutional Neural Networks (CNNs), Gated Recurrent Units (GRUs), Long Short-Term Memory (LSTM) networks, and Deep Neural Networks (DNNs) enhance IDS effectiveness. NSL-KDD dataset was used in [2] for model development and testing, where preprocessing techniques such as feature engineering and normalization were used for improving accuracy. Despite good performance from these models, they must be optimized for use in real-world applications.

Encryption methods are also used in network security. AES and Chaotic Map Algorithm are two widely used encryption processes [3]. AES has proven to be highly resilient and effective, while the Chaotic Map Algorithm provides dynamic encryption. Scalability and efficiency are two very crucial aspects in Big Data environments. Even with encryption, ARP spoofing is not always prevented as there should be further security layers.

Machine learning is now being used more and more to detect ARP spoofing. Authors in [14] use LSTM networks and decision tree classifiers on the Kitsune Network Attack Dataset to classify ARP spoofing attacks. Although their models were very accurate, the research is limited by the quality of datasets. Feature reduction techniques like decision trees can possibly enhance the performance of LSTM, and more extensive datasets should be used.

GAN-based models offer a second viable approach. The GAN-IF framework, initially designed to identify software piracy, can be used for ARP spoofing [4]. Separating legitimate from spoofed ARP traffic, GANs offer a valuable anomaly detection capability. Isolation Forest improves the adaptability factor by identifying anomalous MAC-IP mappings. This approach outperforms traditional IDS with dynamic adaptation to novel attack channels.

Simulation-based methods are useful for controlled experiments. In the research work in [6], GNS3 is used to simulate ARP attacks through tools such as Nping, Arpspoof, and Ettercap. The attacks are identified using XArp software,

showing feasibility. But ARP-based threats might not be completely addressed, and stronger detection models are needed.

Network behavior profiling is another method to detect ARP spoofing. ARP-profiler detects anomalies through packet count, ARP reply ratio, and request-reply ratios [8]. Although promising, this solution needs to be optimized to fit various network environments. Likewise, the system in [9] identifies ARP, DHCP, and DoS attacks on Kali Linux platforms but does not have broad applicability in that system.

There are some studies that integrate attack detection with defense. In [7], there is an educational platform that simulates ARP spoofing attacks and offers countermeasures like static ARP tables and traffic encryption. However, it primarily deals with ARP spoofing, not other IoT vulnerabilities.

Existing work suggests greater incorporation of security features. The D-ARP technique employs signed ARP packets and correlation methods to identify spoofing [11]. Though efficient, its applicability to other types of attacks such as DDoS is not explored. Another method, ARP-PROBE, employs explainable AI methods like SHAP for enhanced interpretability in IoT networks [12].

A new method, EMR-ARP, introduces a voting system to verify ARP messages to avoid MITM attacks [13]. While good, its reliance on ARP protocol manipulation limits compatibility with installed hardware, rendering deployment challenging.

ARP spoofing-based MITM attacks continue to pose a serious threat on university networks. Current ML-based IDS are missing essential network metrics, and conventional methods are unable to address new threats. The research in [15] proposes a dynamic ARP spoofing detection method based on real-time data sets, tuning ML classifiers, especially CNNs, to achieve a 99.26% F1-score for real-time attack detection.

Despite improvements, certain key problems still exist with ARP spoofing detection:

1. Explainable AI: Methods such as SHAP enhance transparency of ML-based Intrusion Detection System..
2. Lightweight Detection: IoT contexts require efficient solutions that compromise between security and resource constraints.
3. Adaptive Security: Dynamic network configurations require adaptive defense systems.
4. Decentralized Architectures: Centralized security presents single points of failure, which must be detected distributively.
5. Client-Side Dependencies: Security mechanisms should limit the use of client-side implementations for greater usability.
6. Protocol Compatibility: New security controls must be compatible with previous systems.

To solve these problems, this paper suggests an optimized, real-time, and interpretable detection model to counter ARP spoofing to improve network security resilience via adaptive and transparent threat detection mechanisms.

2 Overview of ARP Spoofing

ARP spoofing, or ARP poisoning, is an attack that intercepts or redirects traffic on a LAN by exploiting the trust-based nature of ARP in IPv4. Since ARP lacks authentication, attackers can send forged messages to link their MAC address with a legitimate IP.

2.1 The ARP Protocol: A Brief Overview

The ARP protocol was designed to bridge between the IP layer and the data link layer in the OSI model. Devices that are on a local network use ARP to map their IP addresses to MAC addresses. When a device requires sending data to another, it checks its ARP cache to find out if it has a record of the recipient's IP address. If the entry is present, then it sends the data frame with its corresponding MAC address. However, if the entry does not exist, the device broadcasts an ARP request to the local network with the question, "Who has this IP address?" Then, the device having that IP address will reply back with its MAC address.

This process presumes that ARP messages are genuine and that the devices will trust the information in the ARP messages. This is exactly where the problem is in the case of ARP spoofing.

2.2 How ARP Spoofing Works

In ARP spoofing, an attacker sends forged ARP messages linking their MAC address to a legitimate IP (usually the gateway). Victim devices update their ARP tables, causing traffic to be misdirected to the attacker.

1. Attacker Preparation: The attacker must be on the same LAN as the victim.
2. Sending Fake ARP Messages: Fake ARP replies associate the attacker's MAC with a trusted IP.
3. Packet Interception: Traffic meant for the trusted IP is sent to the attacker.
4. Man-in-the-Middle Attack: The attacker can capture, modify, or inject data into the communication.

2.3 Implications of ARP Spoofing

ARP spoofing has several serious security implications, including:

1. Data Interception: The attacker intercepts sensitive information, which could be any form of login credentials, financial information, or private communications.
2. Man-in-the-Middle Attacks: The attacker mimics the victim and gateway, which allows them to manipulate the communication between these two entities by changing messages or redirecting traffic to destinations that might be malicious.

3. Denial of Service (DoS): In some instances, ARP spoofing leads to network disruption through normal flow interruption. For instance, the attacker can flood the network with false ARP messages such that the legitimate devices end up losing connectivity.
4. Session Hijacking: By obtaining session cookies or authentication tokens from intercepted traffic, the attacker can hijack active sessions and gain unauthorized access to online services.

2.4 Real-World Use of ARP Spoofing

ARP spoofing is one of the most common techniques used in cyberattacks, especially on networks that have not been equipped with adequate security measures. Once the attackers take control of the network traffic, they can do a variety of malicious things. They can steal credentials by intercepting login information or other sensitive data in transit, inject malicious code into otherwise safe web traffic to compromise systems, and conduct phishing attacks by manipulating DNS responses or redirecting victims to malicious websites. These activities allow hackers to utilize social engineering attacks and collect sensitive information from unsuspecting users.

A very popular example of exploiting ARP spoofing happens through public Wi-Fi networks when attackers used the method to listen in on people's communications and steal their information. This type of attack has been demonstrated on various penetration testing environments and continues to be a big security concern for networks.

2.5 ARP Spoofing: Attack Types and Challenges

Common ARP spoofing attacks include Man-in-the-Middle (MitM), where an attacker intercepts and modifies communications; Denial of Service (DoS), caused by flooding the network with false ARP replies; and Session Hijacking, where active sessions are intercepted to steal credentials or gain unauthorized access. Defending against these attacks is challenging due to ARP's lack of authentication, but mitigation techniques—such as static ARP entries, DAI, and encryption—can significantly reduce the risks.

2.6 Prevention and Mitigation Strategies

Multiple strategies exist to counter ARP spoofing, each with pros and cons. A layered defense offers the best protection.

1. Static ARP Entries: Manually mapping trusted IP-MAC pairs prevents spoofing by blocking unauthorized ARP replies. Effective for small or critical systems, but hard to scale and manage in large networks.
2. Dynamic ARP Inspection (DAI): Available in managed switches, DAI verifies ARP packets against trusted sources like DHCP snooping tables, rejecting mismatches. Ideal for dynamic environments but requires proper setup and hardware.

3. Network Segmentation and VLANs: Isolating network traffic limits spoofing to a single segment and blocks lateral movement. Effective but demands strict access control and configuration.
4. Encrypting Data: Protocols like HTTPS and VPNs encrypt traffic, ensuring confidentiality even if spoofed. Doesn't prevent spoofing, but protects data integrity. Adds overhead and may need extra infrastructure.
5. Layered Defense: Combine static entries for key devices, DAI for flexibility, VLANs for isolation, and encryption for security. Together, they offer both prevention and mitigation.

3 Methodology

The methodology section will describe step by step how ARP spoofing is performed in a controlled penetration testing environment. This section will describe the setup, execution, and monitoring of an ARP spoofing attack, followed by the analysis of the results and possible mitigations.

3.1 Setup and Tools

To execute ARP spoofing in a controlled manner, we set up a virtualized network environment. The choice of tools and platforms was made based on their relevance to penetration testing and network security research.

Choice of Tools:

1. Kali Linux: Chosen as the attacker machine because it is a penetration testing operating system that comes pre-installed with various security tools, including Ettercap. It provides a stable and secure environment for executing network attacks.
2. Ettercap: Selected as the ARP spoofing tool because it is an open-source network analysis tool that supports Man-in-the-Middle (MitM) attacks, passive and active sniffing, and ARP poisoning. It allows real-time traffic interception and manipulation, making it an ideal tool for our attack scenario.
3. VirtualBox/VMware: Used for setting up the network environment because virtualization provides an isolated testing space, ensuring no harm to actual networks. It allows controlled execution of attacks while preserving system integrity.
4. Metasploitable 2: A deliberately vulnerable Linux-based virtual machine chosen to simulate real-world attack scenarios. It contains multiple security flaws that make it a suitable target for penetration testing.
5. DVWA (Damn Vulnerable Web Application): A vulnerable web application that enables us to analyze how ARP spoofing affects web-based communications and credential transfers. It serves as a practical target to demonstrate real-world exploitation risks.

Fig.1. illustrates the experimental network setup used for ARP spoofing.

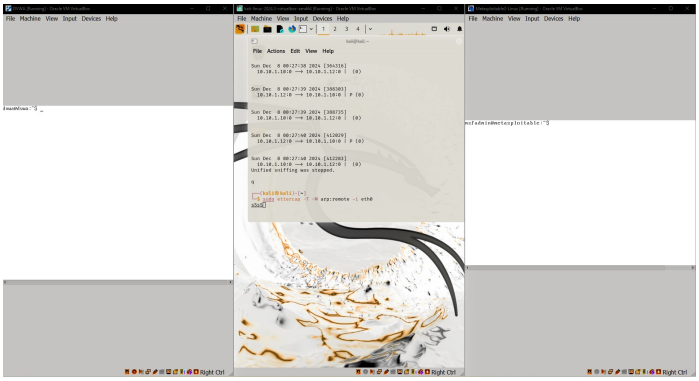

Fig. 1: Experimental setup in VirtualBox, illustrating ARP spoofing with Kali Linux, DVWA, and Metasploitable.

3.2 Step-by-Step Execution of ARP Spoofing

Step 1: Installation and Setup: Ettercap is pre-installed within Kali Linux, along with the network interfaces properly configured to ensure connectivity of the attacker machine (Kali Linux) with both target machines (Metasploitable and DVWA). The attacker machine receives a static IP address so that it is on the same subnet as the victim machines.

Step 2: Initiating ARP Spoofing To initiate the attack, the following Ettercap command is executed on Kali Linux:

```
ettercap -T -M arp:remote
/10.10.1.11// /10.10.1.12// -i eth0
```

Here, '-T' indicates that the mode is text mode; 'arp:remote' is the type of attack, and '/10.10.1.11//' and '/10.10.1.12//' are the IP addresses of the two target machines, respectively. The '-i eth0' option denotes the network interface used.

Fig.2. displays the outcome of an ARP poisoning attack. The attack is on two groups: Group 1, which is traffic from 10.10.1.11 to 10.10.1.10, and Group 2, which is traffic from 10.10.1.10 to 10.10.1.11. This manipulation of ARP tables makes it possible to reroute communication, thus creating a Man-in-the-Middle (MitM) attack scenario. It shows how ARP spoofing can intercept and redirect network traffic.

Step 3: Monitoring ARP Table Changes Once the attack is in motion, we can observe the effects on the victim machines by checking their ARP cache:

```
arp -a
```

At this point, the ARP tables on the target machines should reflect the MAC address of the attacker, meaning that the ARP poisoning was successful.

Fig.3. and Fig.4. together show the successful interaction between the Metasploitable 2 and DVWA machines. The Metasploitable 2 terminal shows its ARP

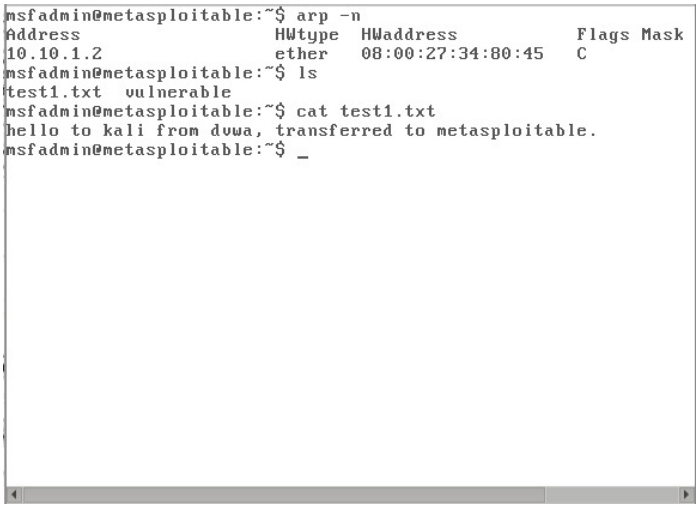

Fig. 2: Metasploitable terminal that shows ARP details and contents of the file transferred.

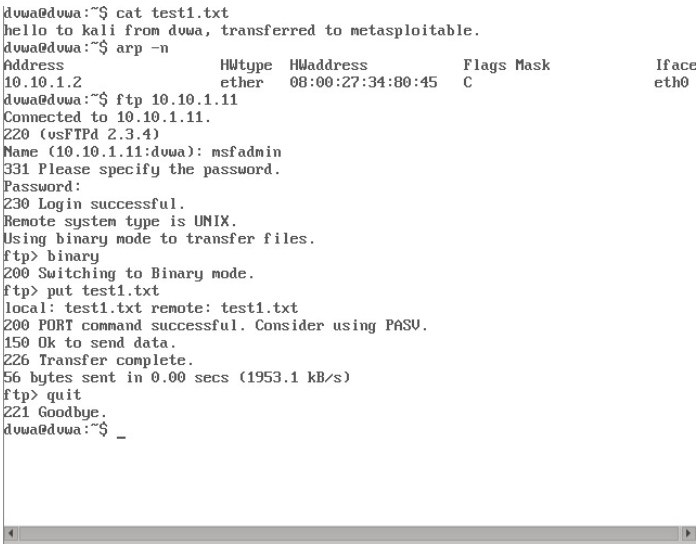

Fig. 3: DVWA terminal illustrating file transfer to Metasploitable using FTP.

table, confirming that the file test1.txt containing the message: "hello to kali from dvwa, transferred to metasploitable." Meanwhile, on the DVWA terminal, the content of the file is verified before transferring it to Metasploitable 2 via FTP in binary mode. The ARP table further confirms a network connection to 10.10.1.2, so inter-machine communication did indeed happen.

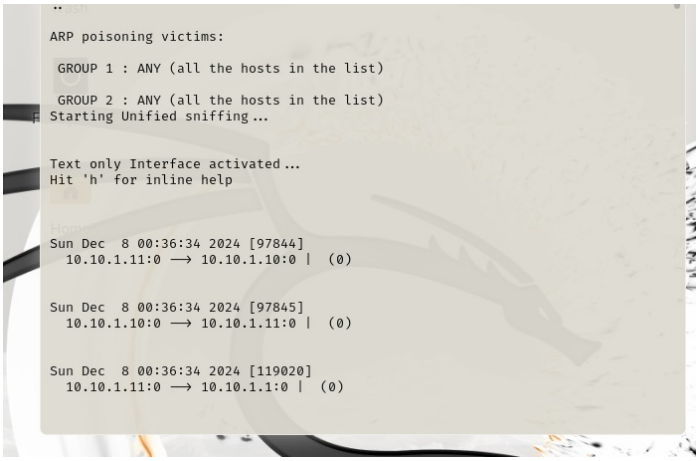

Fig. 4: ARP poisoning results showing manipulated network routes between hosts 10.10.1.11 and 10.10.1.10.

Step 4: Analyzing the Attack Results

The final step in the methodology is to capture and analyze intercepted traffic. This can be done using Wireshark and observing packets flowing between victim machines; the attacker can inspect and change these packets, capturing sensitive data or injecting malicious content.

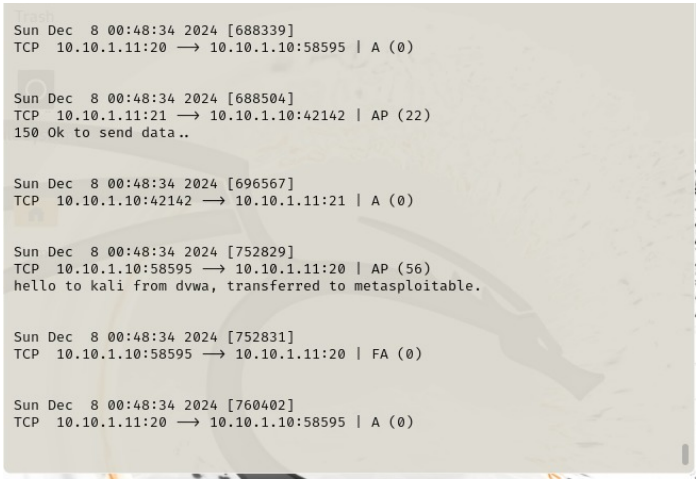

Fig. 5: Packet capture logs showing TCP communication and data transfer between hosts 10.10.1.11 and 10.10.1.10.

Fig. 5. presents the captured network activity logs which outline the TCP communication between the two hosts: 10.10.1.11 and 10.10.1.10. The logs outline processes of data transfer, which in this case involved a message exchange with text "hello to kali from dvwa, transferred to metasploitable". Other flags like "AP" indicate active data transfer processes, while "FA" denotes connection termination acknowledgments. This figure therefore shows a successful data exchange and flow between the two specified hosts.

4 Conclusion

ARP spoofing remains a significant network security threat, enabling attackers to intercept data and disrupt services. This study demonstrated the feasibility of ARP spoofing in a controlled environment and highlighted its potential to compromise communication integrity. Among mitigation strategies—such as static ARP entries, Dynamic ARP Inspection (DAI), network segmentation, and encryption—the focus was on SSH-based defenses. Experimental results confirm SSH effectively encrypts traffic, preserving data confidentiality even under active spoofing.

4.1 Future Work and Practical Implications

Future research will focus on evaluating mitigation techniques under varying network conditions, using metrics like latency and resource overhead. Machine learning-based anomaly detection, enhanced by explainable AI (e.g., SHAP, LIME), offers a promising direction, especially in scalable environments like cloud and containerized networks. Practically, this work aids enterprise security, IoT, and cloud systems by promoting layered defenses and encryption. It also supports cybersecurity education through ARP spoofing simulations, highlighting the need for adaptive, intelligent countermeasures in securing modern networks.

Bibliography

[1] N. Jain, et al., "An Optimized Intrusion Detection Model Using ML and Explainable AI," *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, IEEE, 2024.

[2] N. K. Sah, et al., "Comparative Deep Learning Approach for Intrusion Detection," *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, IEEE, 2024.

[3] K. P. Shah, et al., "Securing Images: Cryptographic Approach in Big Data Scenario," *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, IEEE, 2024.

[4] U. Kumaran, et al., "Adversarial Defense: A GAN-IF Based Cyber-security Model for Intrusion Detection in Software Piracy," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 14, no. 4, pp. 96–114, 2023.

[5] K. M. Majidha Fathima and N. Santhiyakumari, "A survey on network packet inspection and ARP poisoning using Wireshark and Ettercap," *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, pp. 1136–1141, 2021.

[6] T. Vakaliuk, Y. Trokoz, O. Pokotylo, V. Osadchyi, and V. Bolotina, "Emulation and Detection of ARP Attacks in GNS3 Environment: Modelling and Development of a Defense Strategy," 2024.

[7] R. Petrović, D. Simić, S. Stanković, and M. Perić, "Man-in-the-middle attack based on ARP spoofing in IoT educational platform," *2021 15th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS)*, pp. 307–310, 2021.

[8] P. Akhil and B. Antony Jose, "A Profiling Based Approach To Detect ARP Poisoning Attacks," *2021 International Conference on Green Energy, Computing and Sustainable Technology (GECOST)*, pp. 1–5, 2021.

[9] M. Dandotiya, A. S. Dandotiya, N. Dandotiya, and A. Sahu, "A Secure Detection Framework for ARP, DHCP, and DoS Attacks on Kali Linux," *International Journal of Research in Applied Science and Engineering Technology (IJRASET)*, vol. 10, no. 7, pp. 3044–3053, 2022.

[10] R. Gothwal, G. Dharmani, R. S. Reen, and E. G. AbdAllah, "Evaluation of Man-in-the-Middle Attacks and Countermeasures on Autonomous Vehicles," *2023 10th International Conference on Dependable Systems and Their Applications (DSA)*, pp. 502–509, 2023.

[11] S. M. Morsy and D. Nashat, "D-ARP: An Efficient Scheme to Detect and Prevent ARP Spoofing," *IEEE Access*, vol. 10, pp. 49142–49153, 2022.

[12] M. M. Alani, A. I. Awad, and E. Barka, "ARP-PROBE: An ARP Spoofing Detector for Internet of Things Networks Using Explainable Deep Learning," *Internet of Things*, vol. 23, pp. 100861, 2023.

[13] Y. Zhao, R. Guo, and P. Lv, "ARP Spoofing Analysis and Prevention," *2020 5th International Conference on Smart Grid and Electrical Automation (ICSGEA)*, pp. 572–575, 2020.

[14] M. Usmani, M. Anwar, K. Farooq, G. Ahmed, and S. Siddiqui, "Predicting ARP Spoofing with Machine Learning," *2022 International Conference on Emerging Trends in Smart Technologies (ICETST)*, pp. 1–6, 2022.

[15] A. Husain, H. Al-Raweshidy, and W. S. Awad, "ARP spoofing detection for IoT networks using neural networks," *Proceedings of the Industrial Revolution & Business Management: 11th Annual PwR Doctoral Symposium (PWRDS)*, 2020.