UNEC Journal of Computer Science and Digital Technologies
Volume 2, № 1, 2026, pp. 26-33
https://doi.org/10.30546/UNECCSDT.2026.001.243

Section: Cybersecurity & Privacy

Design of an Isolated and Low-Cost Raspberry Pi-Based IoT Network Infrastructure Against ARP Spoofing and Man-in-the-Middle Attacks

Hüseyin Polat[1*], Saadin Oyucu[2], Emre Kaya[3], Mahammad Huseynli[4]
[1,2]Gazi University, Faculty of Technology, Ankara, Türkiye
3Gazi University, Graduate School of Natural and Applied Sciences, Ankara, Türkiye
[3]Azerbaijan State University of Economics, Baku, Azerbaijan
[1]0000-0003-4128-2625, polath@gazi.edu.tr,
[2]0000-0003-3880-3039, saadinoyucu@gazi.edu.tr
[3]0009-0001-4459-7174, emre.kaya@fibabanka.com.tr
[4]0009-0000-8777-601X, mahammad.huseynli@unec.edu.az

Abstract

This study focuses on the design and implementation of a low-cost, isolated network infrastructure to protect IoT devices in home and small-scale business environments from ARP spoofing and Man-in-the-Middle (MitM) attacks. The system is located on a Raspberry Pi and establishes an IoT network in a logically isolated manner, preventing devices on the primary home or enterprise network from directly accessing the IoT devices. This network separation enhances security by reducing the attack surface and limiting horizontal attacks. Network traffic in the isolated environment is monitored with minimal latency. ARP packets are captured, parsed, and analyzed using a specially developed algorithm. Using the data analyzed, the system generates a risk score to assess the probability of ARP spoofing or MitM attacks. Alerts are triggered when pre-defined threshold conditions are met. In this study, the alerts are used for monitoring and evaluation purposes as a first version; however, the architecture is designed to enable integration with firewalls or automated email or SMS mechanisms. The system has been tested in various attack scenarios. Tests have shown that during attack conditions, risk scores exceed threshold values and alerts are triggered. Furthermore, a web interface has been improved, providing ease of use for the end-user. Overall, the proposed solution demonstrates that it provides an affordable, practical, and effective security mechanism compared to its counterparts.

Keywords: IoT Security, ARP Spoofing, Isolated Network Architecture, Raspberry Pi, Network Traffic Analysis

Received:	*Revised:*	*Accepted:*	*Published:*
20/05/2026	*10/06/2026*	*11/06/2026*	*17/06/2026*

1. Introduction

The rapid growth of the Internet of Things (IoT) ecosystem has significantly transformed interactions between the physical and digital worlds. IoT devices, widely used in areas ranging from smart homes to industrial automation, have become essential due to their capabilities in data collection, processing, and communication [1, 2]. However, this rapid proliferation has also exposed substantial security vulnerabilities. Because most IoT devices operate with limited processing power and energy resources, they are generally incapable of

supporting traditional security mechanisms [1, 3]. This limitation makes them highly susceptible to network-based threats [4, 2].

In home and small-scale business environments, a common installation mistake is placing IoT devices on the same network as users' primary devices. This design expands the attack surface, especially in networks that allow guest access or lack comprehensive security controls. Unrestricted access to IoT devices by any connected system opens the door to hacking activities. In this context, attacks exploiting inherent security vulnerabilities in the Address Resolution Protocol (ARP) pose a serious threat, especially since the attack requires low technical expertise [5, 6].

ARP spoofing attacks exploit the lack of authentication in the IP-MAC address mapping process in local area networks [2]. By sending forged ARP responses, attackers can manipulate traffic flows, flood the network with forged ARP packets to block communication, or impersonate the default gateway or another device on the network [4, 7]. These attacks form the basis of Man-in-the-Middle (MitM) attacks, which allow the attacker to interrupt, modify, or block network communication [2, 8]. Given the increasing number of Internet of Things (IoT) devices controlling both sensitive data and physical processes, the impact of such attacks extends beyond data breaches to potential physical hazards.

Although commercially available firewalls, intrusion detection systems, and software-defined networks constitute efficient protection, these tools remain out of the reach of both ordinary users and small companies because of their high price tags and difficult installation procedures that require infrastructure facilities characteristic of large enterprises [9, 10]. Consequently, the emergence of cost-effective and easily usable systems is crucial to the further implementation of IoT security measures.

In light of this requirement, the project will implement a cost-effective architecture based on Raspberry Pi, which is intended to safeguard IoT devices from ARP spoofing attacks and MitM attacks. The architecture will control the attack surface by isolating the IoT devices from the local network and monitoring ARP traffic in the isolated part of the network in real time. A multi-metric scoring algorithm will assign points to IP-MAC pairs, packet rates, and MAC address changes to raise alerts when certain thresholds are surpassed. The architecture is also expected to be flexible enough to allow scalability and usability with existing firewalls or automated response systems. A web interface has been developed to improve the usability of this architecture. It can be noted that Internet of Things (IoT) devices have a relatively lower level of security compared to the traditional computing systems since their hardware and interfaces provide limited resources, thus restricting any user interaction. The use of such devices on a home/business network means that both legitimate and illegitimate devices can establish communication with each other, thus increasing the attack surface. The proposed threat model includes network level threats such as passive and active attacks. ARP protocol, which natively resolves IP addresses to MAC addresses without any authentication, is susceptible to Man-in-the-Middle (MITM) attacks, wherein an attacker may spoof or tamper with the ARP response and masquerade as a gateway or victim device. With MITM attacks, attackers can snoop on, alter, or hijack network traffic, thus making detection of threats in IoT extremely tricky and enabling attackers to linger on for a long time. The unisolated environment wherein IoT and user devices are in the same network segment, especially when guest connectivity is involved, poses the highest level of risks. In order to counter these weaknesses, this research recommends building logically segregated Raspberry Pi network infrastructure for IoT, which is primarily concerned with anomalous behavior monitoring.

2. Literature Review

The rapid expansion of IoT networks has significantly increased data collection and processing capabilities while simultaneously enlarging the cyberattack surface [3, 4]. Due to limited memory, computational power, and energy capacity, IoT devices often cannot support traditional, resource-intensive cryptographic mechanisms, making them vulnerable to network-

based threats such as ARP spoofing and Man-in-the-Middle (MitM) attacks [1, 2, 10]. Edge Computing mitigates latency by processing data closer to its source and can enhance security by shifting certain protection mechanisms to the network edge, reducing dependence on centralized infrastructures [1, 3, 11]. ARP is vulnerable to spoofing attacks in which malicious attackers masquerade as gateway nodes to compromise the traffic [5, 6, 7]. Man-in-the-middle (MitM) attacks can lead to significant data compromises, particularly in high-risk scenarios such as university environments [2, 12, 13]. Economical solutions using the Raspberry Pi device make it easier to deploy firewalls and intrusion prevention systems to monitor and control traffic in small-scale environments [8, 9]. Instantaneous detection approaches, including comparison of static router MAC address tables and ARP tables, can ensure rapid responses in less than one second [5]. The fusion of machine learning algorithms, VLANs, SDN control frameworks, and lightweight hardware systems enhances security in IoT networks [4, 7, 8, 10, 12, 13].

3. Proposed System Architecture

The designed system consists of a relatively inexpensive IoT architecture implemented with isolation and intended for use in home or small businesses environments (Figure 1). The main aim of the architecture is an increase in security as a result of logical separation of IoT devices from the general network utilized for performing regular tasks. According to the developed architecture, the central node based on the Raspberry Pi technology creates the isolated IoT segment and monitors it consistently.

The Raspberry Pi 5 with 8 GB RAM serves as the core platform, offering sufficient processing capacity and low energy consumption. Acting as both a gateway and a monitoring unit, the device manages internet access for IoT devices while simultaneously collecting and analyzing network traffic. This integrated approach eliminates the need for enterprise-grade hardware, providing a cost-effective and compact security solution suitable for non-expert users.

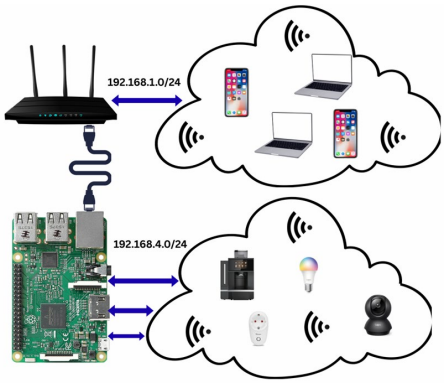

Figure 1. Proposed isolated IoT network architecture

IoT devices are placed in an isolated network segment, which is separate from personal computers, mobile devices, and guest access networks. The internet connectivity for the Raspberry Pi is provided through Ethernet and then distributed among IoT devices in a controlled way. Despite this, IoT devices remain accessible to the Internet, but are not allowed for direct communication with other network devices. The isolation helps reduce exposure to threats within the internal network, especially in those with guest access capabilities.

The network traffic within the isolated IoT segment is continuously monitored using software components running on the Raspberry Pi. Network packages are captured without affecting their transmission and then analyzed at the protocol level. To save time and improve performance, only those traffic types that are interesting for our analysis, namely ARP and RARP traffic, are studied.

Captured ARP packets are analyzed in detail, as ARP plays a critical role in IP–MAC address mapping and is central to detecting ARP spoofing. Attributes such as source and

destination addresses, timestamps, and packet types are extracted and structured for further analysis and scoring.

Risk assessment of the analyzed ARP information uses a special scoring algorithm to check for potential anomalies on certain time periods. Risk scores obtained from this process are then evaluated in comparison with preset thresholds, and alerts are generated if needed. At present, although alerts perform only an informational function, the system can be modified to implement firewalls based on alerts.

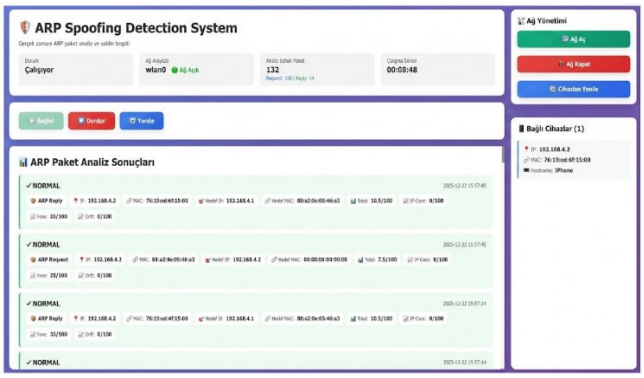

Figure 2. The proposed system's web-based interface

To increase user-friendliness, a web interface is designed to help users monitor the status of the network, connected devices, traffic statistics, and received alerts. This is implemented as an additional layer of support for successful system management regardless of any technical skills (Figure 2).

4. Attack Detection Approach

The following is a multi-tiered, real-time detection methodology for detecting ARP spoofing attacks via thorough traffic analysis. Rather than using one criterion for determining the threat level, each captured ARP packet is analyzed independently through three different approaches to obtain an integrated score. The whole detection process can be divided into four steps. Initially, ARP packets are captured with timestamps to form a historical data set with IP-MAC pairs, frequency of ARP messages, and changes in MAC addresses. In the next step, the system is able to learn from normal traffic behavior at each IP address for dynamic threshold settings. In the third step, there are three scoring algorithms working in parallel for each individual packet. The detection logic is structured around three primary analysis categories.

IP–MAC Consistency Analysis: In normal conditions, an IP address maps to a single MAC address. ARP spoofing disrupts this pattern by associating a single IP address with multiple MAC addresses. The system monitors the number, frequency, and timing of mapping changes. Oscillation patterns where an IP alternates between MAC addresses are considered strong attack indicators. Additionally, a single MAC address mapped to multiple IP addresses is flagged as suspicious.

ARP Packet Rate Analysis: The attacker attempts to flood the network with ARPs reply packets in order to affect the contents of the ARP table. This tool measures the rate of ARP packet sending per IP for different time windows (1 minute, 5 minutes, and 15 minutes) to identify abrupt changes and continuous abnormalities. It compares the current behavior with the learned normal one. The ratio between ARP reply and ARP requests is analyzed since an abnormal reply of traffic might suggest an attack.

MAC Address Change Analysis: Frequent, rapid MAC address changes are uncommon in legitimate networks but are typical in spoofing attacks. A time-weighted mechanism evaluates MAC changes within the last hour, giving greater importance to recent events. Unique MAC counts and oscillation behaviors are also considered.

Each category produces a score between 0 and 100. The final risk score is calculated using weighted averaging: IP–MAC consistency (50%), ARP packet rate (30%), and MAC change behavior (20%). Risk levels are defined as Normal (<20), Low Risk (20–50), High Risk (50–75), and Attack (≥75). Additionally, if any single category exceeds 50, the system directly escalates to "Attack." This dual-layer threshold logic enhances both sensitivity and specificity while enabling rapid detection of significant anomalies.

5. Experimental Environment, Test Scenarios and Observations

The current section is dedicated to the validation process of the proposed system in an actual home and business setting. The first goal was to evaluate the ability of the developed method to detect ARP spoofing attack within controlled but real-life conditions.

5.1 Setup of the Test Environment

The experimental infrastructure was built using a Raspberry Pi 5 (8 GB RAM) and a Tapo C211 home security camera. The Raspberry Pi established an isolated IoT network and provided internet connectivity exclusively through this segment. IoT devices connected only to the isolated network and had no direct communication with the main home or institutional network. This configuration reflects a common real-world "guest network – IoT network separation" model. The Raspberry Pi was connected to the external network via Ethernet and broadcast a separate wireless network for IoT devices. All ARP traffic within the isolated network was monitored and analyzed in near real-time. During testing, natural traffic was generated through real IoT devices to observe system behavior under typical usage conditions.

5.2 Non-Attack Scenarios (Normal Network Behavior)

In cases of no threat attacks, only the authorized devices within the Internet of Things will be operating in the network. In such circumstances, risk scores have stayed low and stable throughout the experiment. IP-MAC mappings are accurate; ARP packets are at expected levels, while changes in MAC addresses are uncommon. The system allows temporary abnormalities but classifies them as low-risk events; the highest risk score achieved was 12.1 (Figure 3).

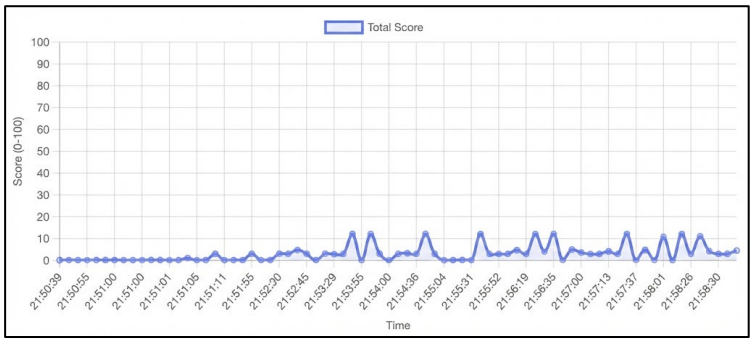

Figure 3. System output total score table in normal network flow

5.3 Attack Scenarios

Controlled ARP spoofing attacks were then executed within the isolated network. The attacker device attempted to manipulate the ARP tables of other devices. Upon initiation of the attack, significant anomalies were detected, particularly in IP–MAC consistency analysis. Rapid changes in IP–MAC mappings and abnormal increases in ARP reply packets led to a rapid escalation in risk scores. As the attack persisted, total risk scores progressively increased, and the system generated "High Risk" and "Attack" level alerts (Figure 4).

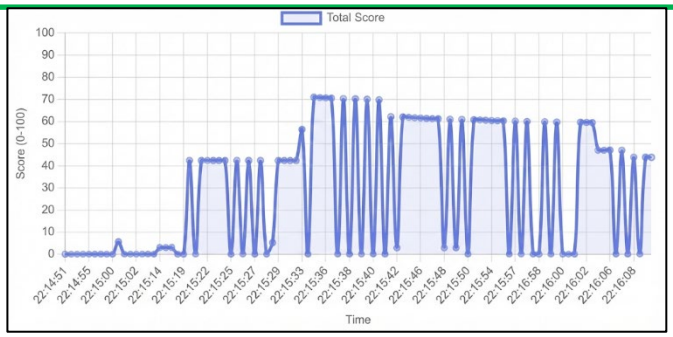

Figure 4. System output of total score table during attack

5.4 Evaluation Criteria and Analysis of Generated Alerts

The evaluation process was concerned with the assessment of criteria based on observation such as alert consistency during attacks, reduction of scores following attacks, and consistency during normal operations. The alerts were set based on accumulated anomalies but not individual packets, thus reducing error due to fluctuations. Moreover, the system responded quickly when high anomalies occurred in each category of analysis. (Figure 5).

Timestamp	IP Address	Total Score	IP-Consistency	Rate Anomaly	Drift Detection	Status
22.12.2025 22:16:35	192.168.4.20	62.43	67.35	59.2	55.0	Attack
22.12.2025 22:16:35	192.168.4.1	0.99	0.0	3.29	0.0	Normal
22.12.2025 22:16:34	192.168.4.20	62.61	67.69	59.2	55.0	Attack
22.12.2025 22:16:34	192.168.4.1	1.14	0.0	3.8	0.0	Normal
22.12.2025 22:16:33	192.168.4.20	62.79	68.05	59.2	55.0	Attack
22.12.2025 22:16:33	192.168.4.1	1.37	0.0	4.56	0.0	Normal
22.12.2025 22:16:32	192.168.4.20	62.97	68.42	59.2	55.0	Attack
22.12.2025 22:16:31	192.168.4.20	63.16	68.79	59.2	55.0	Attack

Figure 5. Attack alerts

5.5 System Performance and Latency

The system operated near real-time on Raspberry Pi 5, with no noticeable latency. Packet capture, analysis, and risk score updates did not disrupt network performance or IoT device functionality. The web-based interface effectively displayed network status, connected devices, and alerts, confirming the system's practicality and usability for home users and small businesses.

6. Discussion and Future Work

The Raspberry Pi-based isolated IoT network architecture proposed in this study provides a low-cost and practical security solution for home and small-scale business environments against ARP spoofing and Man-in-the-Middle attacks. Experimental findings show that the system maintains low risk scores under normal traffic conditions while significantly increasing scores during attack scenarios. By logically separating IoT devices from the main network without requiring VLAN configuration or additional enterprise hardware, the architecture effectively narrows the attack surface. The Raspberry Pi 5 platform successfully performed near real-time ARP analysis without negatively impacting network performance.

The multi-layered detection approach enhances reliability by jointly evaluating IP–MAC consistency, ARP packet rate, and MAC address changes rather than relying on a single metric. Its rule-based design eliminates the need for training data, enabling rapid deployment and adaptability in resource-constrained environments. Although the dynamic learning mechanism reduces false positives, it may slightly increase detection time. The web-based management interface improves usability by presenting visual risk indicators and alerts, making the system accessible to non-expert users. However, the current implementation focuses solely on ARP-

layer analysis; DNS spoofing, application-layer threats, and the inspection of encrypted traffic remain outside its scope. Additionally, reliance on a single Raspberry Pi node introduces a potential single point of failure.

In terms of scalability, the proposed system's current architecture targets home and small-scale business environments and can effectively support a limited number of IoT devices. Considering the processing power and memory capacity of the Raspberry Pi 5, an increase in latency and resource usage in the ARP packet analysis process is expected as the number of devices that can be monitored simultaneously increases. Theoretically, it is predicted that the Raspberry Pi 5, with its 8 GB RAM and quad-core processor, can seamlessly support dozens of IoT devices; however, load tests with a large number of devices are required to definitively determine this limit. In this study, only a limited number of real IoT devices could be used in the test environment; scalability evaluation in a large-scale environment could not be performed due to practical limitations. This limitation is considered a research area that is planned to be addressed with more comprehensive testing infrastructures in the future.

In terms of cost comparison, the proposed system offers a significant cost advantage compared to commercial IoT security gateways. Enterprise-class security devices such as Cisco Meraki MX or Fortinet FortiGate require a budget of hundreds to thousands of dollars, including hardware costs and annual license fees. In contrast, the Raspberry Pi 5 hardware used in the proposed system can be obtained for approximately $80-100 USD, with no ongoing license or subscription costs. This cost difference makes the proposed approach an attractive alternative, especially for home users and small businesses that cannot budget for enterprise solutions. While it is acknowledged that the comprehensive feature set and technical support offered by commercial solutions cannot be fully met by this system, the proposed architecture is considered to offer an effective and accessible solution in terms of basic security needs.

Overall, the study demonstrates that low-cost hardware, such as Raspberry Pi, can be effectively used in security-focused network applications. Future work will include automated response mechanisms, hybrid machine learning-based detection models, broader protocol analysis (e.g., DNS, DHCP, MQTT), scalability evaluation in dense IoT environments, distributed multi-node architectures, long-term real-world testing, user behavior-based alerting, and energy efficiency optimization.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. T. R. Hadiningrum, R. A. D. Talasari, K. F. Ilham, and R. M. Ijtihadie, 23, “Survey on Risks Cyber Security in Edge Computing for The Internet of Things Understanding Cyber Attacks Threats and Mitigation”, JUTI, vol. 23 no. 1, pp. 29–50, Feb. 2025, doi: 10.12962/j24068535.v23i1.a1210.
2. H. Fereidouni, O. Fadeitcheva, and M. Zalai, “IoT and Man-in-the-Middle Attacks,” SECURITY AND PRIVACY, vol. 8, no. 2, Mar. 2025, doi: 10.1002/spy2.70016.
3. K. U. Aditya, P. N. Kamath, Y. Poral, B. D. Mallika and V. Acharya, "Framework for Early Cyber Attack Detection Using ML Models Deployed on Fog Devices," 2024 12th International Symposium on Digital Forensics and Security (ISDFS), San Antonio, TX, USA, 2024, pp. 1-6, doi: 10.1109/ISDFS60797.2024.10527351.
4. Ali, Anas & Husain, Mubashar & Hans, Peter. (2025). Intelligent ARP Spoofing Detection using Multi-layered Machine Learning (ML) Techniques for IoT Networks. 10.48550/arXiv.2507.21087.
5. Imad, Hiba & Abdulridha Hussain, Mohammed. (2022). Defending a wireless LAN against ARP spoofing attacks using a Raspberry Pi. Basrah Researches Sciences. 48. 123-135. 10.56714/bjrs.48.2.12.

6. S. Selvarajan, M. Mohan and B. R. Chandavarkar, "Techniques To Secure Address Resolution Protocol," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1-7, doi: 10.1109/ICCCNT49239.2020.9225413.
7. S. Sun, X. Fu, B. Luo and X. Du, "Detecting and Mitigating ARP Attacks in SDN-Based Cloud Environment," IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, ON, Canada, 2020, pp. 659-664, doi: 10.1109/INFOCOMWKSHPS50562.2020.9162965.
8. Karahan, O., & Kaya, B. (2020). Raspberry Pi Firewall and Intrusion Detection System. Journal of Intelligent Systems: Theory and Applications, 3(2), 21-24. https://doi.org/10.38016/jista.653486
9. Adrian, Ronald & Mandasari, R. & Alam, Sahirul. (2025). Design and Implementation of a Machine Learning-Based Adaptive IDS on Raspberry Pi for Smart Home Network Security. Jurnal Sains dan Teknologi Industri. http://dx.doi.org/10.24014/sitekin.v22i2.33485
10. Hussain, Shafiq. (2025). Securing IoT Devices and Edge Computing with Hybrid Mesh Firewalls. https://doi.org/10.13140/RG.2.2.27030.05441.
11. Amin, Rashid & Hussain, Mudassar & Alhameed, Mohammed & Raza, Syed & Jeribi, Fathe & Tahir, Ali. (2020). Edge-Computing with Graph Computation: A Novel Mechanism to Handle Network Intrusion and Address Spoofing in SDN. Computers, Materials & Continua. https://doi.org/10.32604/cmc.2020.011758
12. Girdler, Tom & Vassilakis, Vassilios. (2021). Implementing an intrusion detection and prevention system using Software-Defined Networking: Defending against ARP spoofing attacks and Blacklisted MAC Addresses. Computers & Electrical Engineering. 90. 106990. 10.1016/j.compeleceng.2021.106990.
13. Sharmistha Majumder, Mrinal Kanti Deb Barma, and Ashim Saha. 2024. ARP spoofing detection using machine learning classifiers: an experimental study. Knowl. Inf. Syst. 67, 1 (Jan 2025), 727–766. https://doi.org/10.1007/s10115-024-02219-y