Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

Mitigating Address Resolution Protocol-Based Man-In-The-Middle Attacks in Local Area Networks

Charles Taminyan Siekang
Department of Cyber Security and Computer Engineering Technology, University of Technology and Applied Sciences (UTAS), Navrongo, Ghana.

Paula Aninyie Wumnaya
Department of Cyber Security and Computer Engineering Technology, University of Technology and Applied Sciences (UTAS), Navrongo, Ghana.

***Abstract*—In this modern era, computer devices communicate with each other via a network. This could be a wired or wireless network, but wireless network is more popular than the wired one since it can be accessed anywhere within a given range. Due to this, network security has become a major concern. Man-in-the-Middle (MITM) attacks is one of the most prevalent cyber-attacks threatening the security of local area networks (LANs). The MITM attacker can intercept, modify, change, or replace target victims' communication traffic without their awareness. Existing network-based mitigation solutions such as intrusion detection systems (IDS), static Address Resolution Protocol (ARP) tables, cryptographic and host-based mechanisms, often require specialized hardware or administrative control making them unsuitable for client-side device protection in a LAN network. This paper proposed and implemented a non-cryptographic, lightweight client-side system for Linux-based operating systems to ensure their protection. The program continuously monitors the ARP cache for anomalies and responds to potential spoofing attempts in real-time by issuing alerts and blocking the attacker from making unauthorized changes to the ARP cache. To evaluate its effectiveness, an ARP spoofing attack was simulated in a controlled virtual platform. The results demonstrate that the proposed system offers a tenable solution that should be prioritized to defend against ARP-based MITM attacks in LANs, while creating user awareness of possible spoofing attacks.**

Keywords*—ARP Spoofing; Man-in-the-Middle Attacks; Local Area Networks; Client-Side Security; Network Security; ARP Mitigation, Address Resolution Protocol (ARP), ARP Cache Poisoning, Intrusion Detection System, Lightweight Security Mechanism, and Linux-Based Security Tool.

I. INTRODUCTION

The rise in technological advancements in recent times has led to an increase in cybercrimes or attacks, posing significant threats to network security, and other digital infrastructure. Cyber-attack is the deliberate attempt to gain access to a system, network or any digital device without authorization, to cause harm or damage [1]. There are various kinds of cyber-attacks including, malware, denial of service (DoS), Man-in-the-Middle (MITM), social engineering attacks [1]. Among these attacks, MITM attack presents a critical security concern. An MITM attack is a kind of attack where a malicious third party secretly takes control of the communication channel between two or more endpoints, allowing the attacker to intercept, modify, change, or replace target victims' communication traffic without their knowledge [2][1].

MITM attacks can be performed in various ways, including Internet Protocol (IP) spoofing, Domain Name System (DNS) spoofing, Address Resolution Protocol (ARP) spoofing, Hypertext Transfer Protocol Secure (HTTPS) spoofing, and Secure Socket Layer/Transport Layer Security (SSL/TLS) hijacking [2], [3], [4]. In Local Area Networks (LANs), ARP spoofing is widely used to place the attacker between the victim and the default gateway since ARP is a stateless protocol, with inherent vulnerabilities such as weak encryption, misconfigured devices, and lack of robust authentication mechanisms [5], [4], [6]. Network traffic interceptions and manipulation occur via sending forged ARP packets to poison the victim's ARP table and redirect network traffic through the attacker's system [7].

Such attacks can cause data breaches, financial losses, and damage to reputation to affected institutions [2]. Even though several network-based solutions such as intrusion detection systems (IDS), static ARP tables, cryptographic and host-based mechanisms have been proposed [2][8], a lot of them require specialized hardware or administrative control over the entire network. This paper proposes a non-cryptographic mitigation approach using a client-side system that can be deployed on individual operating systems to continuously verify the integrity of the gateway IP-MAC pairings to detect and prevent ARP spoofing attacks.

II. RELATED WORKS

Most researchers, scholars and cyber security experts globally, have proposed various mitigation strategies to prevent cyber-attacks, with ARP-based MITM attacks being a notable concern in LANs. The increase in ARP-related vulnerabilities is due to lack of authentication, and ARP being a stateless protocol [9].

In [2], passive detection techniques were used to monitor ARP traffic for anomalies, focusing on IP-MAC parings. The study suggests the use of Arpwatch to monitor Ethernet traffic and alert administrators to changes in IP-MAC pairings, but it relies on administrators to determine whether an attack has occurred or not. The use of ARP-Guard and ARP-Defender, which use sensor-based architecture to monitor network traffic, helps in ARP spoofing detection. However, it may generate false positives and requires manual intervention for alerts.

In [3], active detection techniques were used to inject ARP messages into the network intentionally to detect

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

inconsistencies. Ettercap was configured to send known ARP requests and examine the responses to identify spoof ARP entries. The limitation is that it requires manual intervention and may not work well in larger networks. Another method is routing trace analysis, which detects ARP attacks by observing changes in the Time-To-Live (TTL) values in packets to infer deviations in network routing paths [9].

The authors in [10] suggest the use of static ARP entries to mitigate ARP poisoning attacks in LAN by changing dynamic IP-MAC mapping to a static one. Low scalability and much workload for network administrators in larger networks are the drawbacks of this strategy. They further proposed the use of automated ARP static entries system [9].

As proposed in [2], Secure ARP (S-ARP) is a cryptographic approach which uses public-key cryptography to authenticate ARP replies, thereby allowing only valid users to update ARP tables. However, this approach may lead to a single point failure since it relies on a central Authoritative Key Distributor (AKD). If the AKD fails, the whole system may stop working [8].

According to [2][10], hardware-based solutions employ a Dynamic ARP Inspection (DAI), a security mechanism implemented in switches. DAI ensures that only valid ARP requests and responses are forwarded. This helps prevent ARP spoofing attacks by monitoring ARP packets to ensure only valid IP-MAC pairings can be forwarded to a particular host. However, this approach may not be effective if the ARP spoofing occurs among the wireless nodes connected via the same Access Point (AP) [2].

The authors in [9] suggest the use of a central server ARP model, a server-based solution to collect IP-MAC mapping in LAN and store a table of valid host on the network. The stored data is used to track IP-MAC mappings that are not present in the table and flag as possible spoofing attacks. The limitation is that, if the central server fails, the system may not be able to detect attacks.

The authors in [2], suggest the use of host-based solutions to protect individual host or client devices on the network. Anticap was configured to prevent ARP spoofing attacks but works on Unix-based operating systems. It denies ARP updates where the MAC address differs from the IP-MAC pairings in the ARP cache. Also, client-side Bash script is a mitigation mechanism which monitors and stores the default gateway IP and MAC address to a file [8]. It then compares the gateway information with the pre-saved valid data, notifying the user if a mismatch occurs as a possible spoofing attack. However, the system is not automated, it requires manual intervention.

Despite the various mitigation strategies put in place to stop ARP-based MITM attacks in LANs, some gaps still exist. Most tools currently implemented suffer from platform dependency issues, require significant user intervention, and demand considerable technical expertise to operate. Also, some detection and prevention mechanisms consume a lot of computing resources and lacks automation [11][12]. To address these gaps, this paper proposes a Linux-based, automated and lightweight client/user-end system that can be deployed in user devices to prevent ARP-based MITM attacks in LAN environments. The system continuously monitors ARP cache in real time, detect spoofing attempts by comparing the default gateway's MAC address against trusted data that contains the gateway IP and MAC addresses, and automatically block malicious entries from updates to the ARP cache. Moreover, the system is designed such that it runs in the background with less resources required, including user-friendly alert messages of possible spoofing attacks and attack logs for further investigation and analysis.

III. METHODOLOGY

A. Threat Model

The threat model assumes that both the victim and attacker devices are connected to the same wireless router in a Wireless Local Area Network (WLAN). The router serves as an access point and a default gateway, allowing both hosts to operate within the same subnet and broadcast domain. It is assumed that the attacker has access to the wireless credentials since he/she is lawfully authorized to be on the network. This demonstrates real-world implementation of wireless networks.

The attacker can send spoofed ARP packets and messages to execute an ARP spoofing attack using Ettercap, allowing the attacker to place his/herself between the victim and the router. Once the attack succeeds, the attacker can sniff, detect and alter network traffic passing between the victim and the gateway. However, the attacker has no administrative control over the wireless router or any other network infrastructure device.

It is assumed that the network works without Dynamic ARP Inspection (DAI) or any centralized ARP validation policy, making it vulnerable to ARP spoofing attacks. Also, the attacker is not initially on-path but achieves that via ARP spoofing. Figure 1 illustrates the proposed threat model of an ARP spoofing attack in a WLAN.

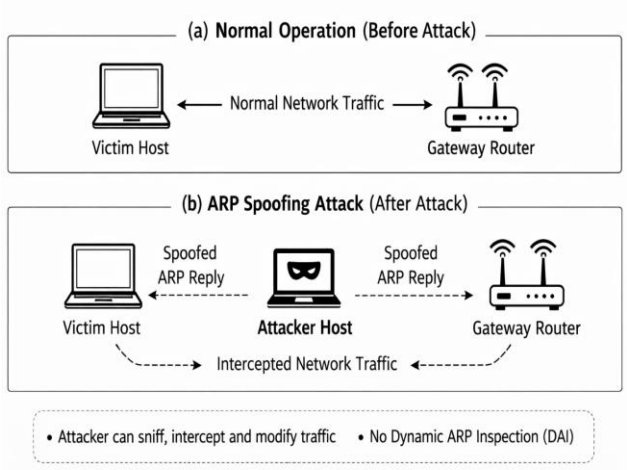

Fig. 1. Threat model of a normal WLAN communication and an ARP spoofing attack.

B. System Design

The system was designed to monitor ARP cache, detect unauthorized gateway MAC modifications, and automatically mitigate detected spoofing events. The architecture of the ARP spoofing mitigator is built as a user-end defensive mechanism that operates autonomously without any reliance on a centralized network infrastructure or a centralized administrator. The automated client-side system monitors the ARP cache of the victim's machine repeatedly, enabling it to detect and block any suspicious changes to the gateway's ARP

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

cache. The system is developed based on a modular structure using python as the core programming language. These modules are initialization, monitoring, validation, alert, mitigation and logging modules. This architecture enables the client-side system to detect and respond to ARP-based MITM attacks in LANs. Figure 2 represents a flowchart of the proposed system, adopted and modified from the model presented in [8].

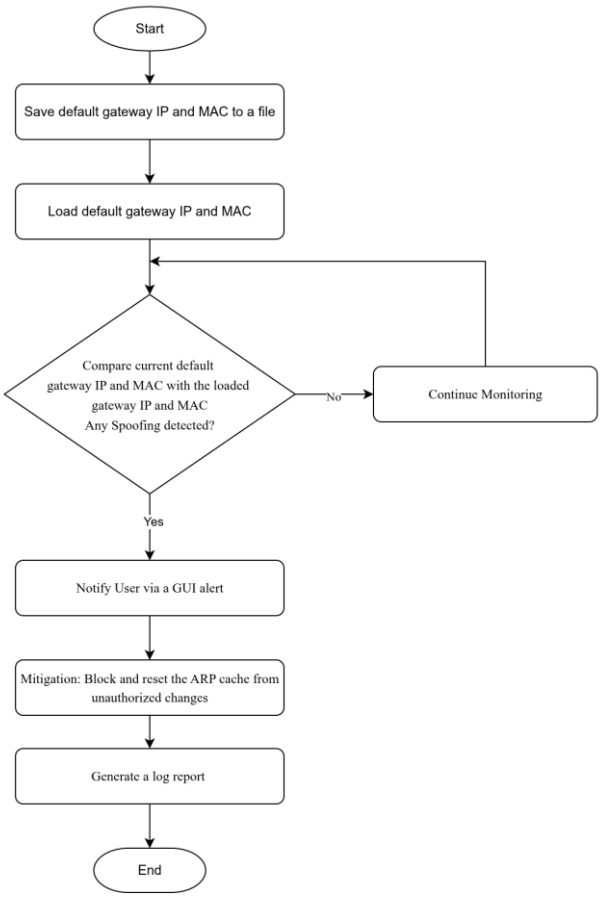

Fig. 2. Flowchart of the proposed system

C. Detection Algorithm

Input:
G_IP: Default gateway IP address;
G_MAC_ref: Trusted MAC address of the gateway
Parameters:
T = 3 seconds (polling interval);
N = 3 consecutive mismatches (alert threshold)
Procedure:
Initialize mismatch counter C ← 0
while monitoring is active **do**
  Wait for T seconds
  Obtain current MAC address G_MAC_cur associated with G_IP
  if G_MAC_cur ≠ G_MAC ref **then**
    C ← C + 1 {Anomaly detected}
  else
    C ← 0 {Reset counter}
  end if
  if C ≥ N **then**
    Check for legitimate gateway change
    if legitimate change confirmed **then**
      Update baseline: G_MAC_ref ← G_MAC_cur
      Reset C ← 0
    else
      Classify event as ARP spoofing
      Insert MAC-based firewall rule to block G_MAC_cur
      Generate user alert
      Log attack event
      Reset C ← 0
    end if
  end if
end while
Output: Detection alert and mitigation via MAC-based firewall rule

From the algorithm above, an event is considered an anomaly if there is a deviation between the current MAC address of the default gateway and the trusted MAC address of the default gateway that is saved during system initialization. Mathematically, an anomaly is expressed as;
G_MAC_cur ≠ G_MAC_ref, where G_MAC_cur = current MAC address and G_MAC_ref = trusted MAC address of the default gateway.
To differentiate between a transient ARP churn and an attack, the system only raises an alert of a possible ARP spoofing attack after the anomaly condition persist for three consecutive monitoring intervals.
Also, Legitimate modification of the gateway MAC address could happen due to hardware replacement, renewal of DHCP, high-availability failover (e.g., VRRP/HSRP) or transition of a wireless network. To avoid false-positive hitches when such events occur, a controlled re-baselining mechanism is adopted. In such cases no anomaly is detected, and the new gateway MAC is accepted as legitimate. This exception guarantees that the system does not update its baseline when suspicious activity is occurring, to avoid training on an attacker-controlled state. Therefore, a new gateway MAC address is considered legitimate if and only if the above stated conditions are met.

D. Simulation

The Oracle VirtualBox is used for simulation since it acts as a platform for performing simulated tests without affecting real-time systems. The setup comprised of two Linux machines (attacker and victim) and a wireless router, each assigned a defined role. The client-side system was installed and configured on the client or victim device to enable it to monitor the ARP cache for any spoofing attempts. The Attacker's machine running on the VirtualBox was used to launch the attack using Ettercap, targeting the client device. Lastly, the router acts as the default gateway and access point which allows for communication within WLAN. All devices within the WLAN use the same subnet to enable real-time replications of LAN conditions.

Ettercap, a network penetration testing tool was used to execute the attack. One of the simplest and easiest tools to conduct an ARP poisoning on Linux based system is Ettercap [8] [3]. To launch such an attack using Ettercap, the attacker must be connected to the same network as the victim to discover all active hosts. Once that is done, targets are selected, and the attack is initiated. Figure 3 shows the attack simulation process using Ettercap [8].

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

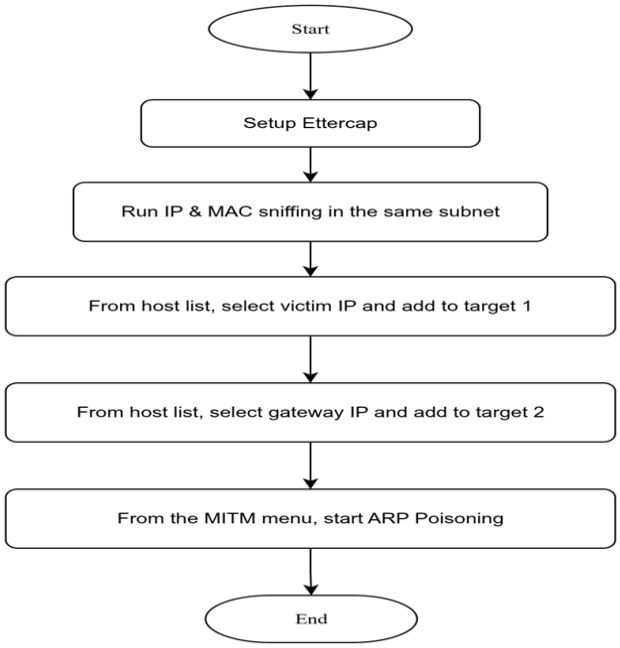

Fig. 3. Attack simulation process using Ettercap

IV. RESULTS AND DISCUSSION

The results of the ARP spoofing attack simulation, detection and mitigation are presented. The simulation was conducted using the Oracle VirtualBox where attacks were executed using Ettercap while the client-side mitigation tool was configured on the victim's machine. The effectiveness of the system was evaluated based on these principal functionalities: monitoring, detection, mitigation and logging of spoofed events.

The monitoring functionality of the system continuously checks the ARP cache and the default gateway traffic for anomalies. This allows the system to detect ARP spoofing attacks on the user's device. The client-side mitigation system was configured on the user's device to monitor ARP spoofing events. It can be configured to run automatically from system's startup or manually launched using the terminal command, sudo python3 arpmonitor.py. Figure 4 shows the monitoring interface of the client-side mitigation system.

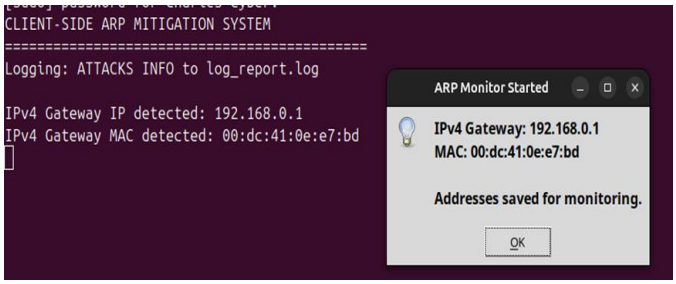

Fig. 4. Monitoring interface of the client-side mitigation system

The Detection functionality was tested based on the system's ability to detect and notify the user of a possible ARP spoofing attack. The client-side mitigation system was installed and configured on the victim's device, allowing it to monitor the default gateway for possible spoofing events. The system continuously monitored the ARP cache and compared the MAC address of the gateway with the one saved during the system's initialization. Once a mismatch is detected three consecutive times, a popup alert notifies the user of a possible spoofing attack. The alert message shows the target IP, valid MAC and the spoofed MAC addresses while advising the user to report the event to the network administrator. Figure 5 shows a terminal attack detection interface.

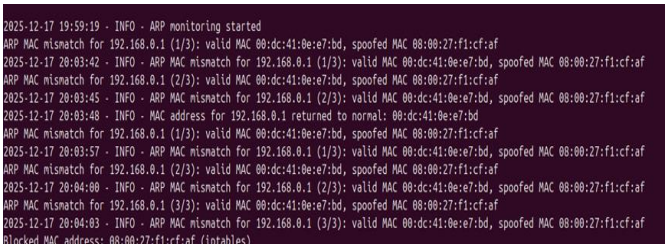

Fig. 5. Terminal attack detection interface

Mitigation happens after spoofing is detected. Once an event is flagged as a spoofing attack, the next step is to prevent further modifications of the ARP cache by restoring original IP-MAC pairings of the default gateway previously saved during system initialization. To further ensure proper mitigation, a firewall rule is dynamically added to counter the attack using iptables packet filtering system. The command "iptables -IINPUT – mmac - mac – source < attacker_mac > -jDROP" is used to drop packets originating from the attacker's MAC address, thereby by blocking malicious network traffic.

The ability of the system to automatically block the attackers' MAC address without any intervention by the user, confirms the functionality of the mitigation module. Figure 6 illustrates the GUI attack detection and mitigation interface. This result confirms the proper functioning of the detection, alert and mitigation modules.

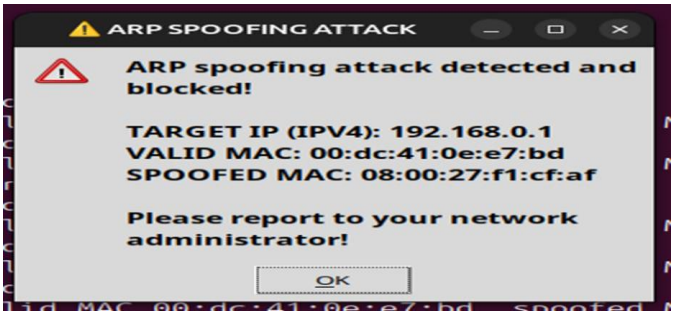

Fig. 6. GUI attack detection and mitigation interface

The logging functionality of the system was tested based on its ability to generate a log report of all spoofed events named, log_report.log. After every detected and mitigated event, a new entry is logged to the log_report.log file. Each log event records the time and date the attack occurred, the target IP address, the spoofed MAC address, valid MAC address and the block status. If the mitigation is successful, the block status indicates "True" else "False" will be displayed. The essence of the logging feature is not only centered on report generation, but its importance for post-attack analysis. It helps system and network administrators to understand the dynamics of spoofing attacks. It also helps with system auditing and forensic purposes. Figure 7 is an illustration of logged events of spoofing attacks.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

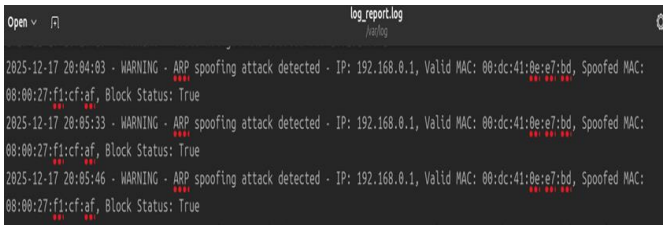

Fig. 7. Logged events of spoofing attacks

TABLE I. RESULTS OF QUANTITATIVE PERFORMANCE METRICS.

Metric	Value
Polling interval	3 seconds
Alert Threshold	3 consecutive mismatches
Detection latency	6 – 9 seconds
False positives	0
CPU usage	< 2%
Memory consumption	< 40MB
Number of trials	15

CONCLUSION

The system's effectiveness was demonstrated based on its ability to detect unauthorized MAC address changes, automatically mitigating spoofing attacks and generating a log file of spoofed events for post-attack analysis. Despite the success of the system in achieving its objectives, some limitations were revealed including platform dependency issues, limited scope of monitoring, single host protection and its reactive detection nature. Nonetheless, the study successfully delivered an effective host-based solution for ensuring client-side security in a LAN environment and provides a strong foundation for future improvements.

The developed system supports only Linux-based systems. Future research solutions should focus on cross-platform compatibility like Windows or macOS for easy deployability [4] [12].

The system currently detects spoofing events by monitoring only the default gateway, limiting its ability to detect other kinds of attacks targeting essential network devices like servers. Improving the monitoring scope would enable the system to ensure more comprehensive protection [2][3].

The system operates reactively, which only detects after the ARP cache is poisoned. Proactive detection strategies such as anomaly-based detection can be incorporated to help predict and prevent attacks before they cause any damage [13][11].

The system currently offers protection to individual hosts running it, making it unsuitable for enterprise networks. An enhanced version of the system could be developed to curb this limitation.

ACKNOWLEDGMENT

Firstly, I would like to thank the Almighty God for guiding and protecting me throughout my study. I am most grateful also to my co-author, Paula Aninyie Wumnaya, for her contribution and guidance throughout this study. To my family, friends, and everyone who aided me in any way, God richly bless you.

REFERENCES

[1] R. Gulshan and S. S. Chauhan, "A survey on cyber security threats," in Proceedings of the 2021 International Conference on Technological Advancements and Innovations (ICTAI), IEEE, 2021. https://doi.org/10.1109/ICTAI53825.2021.9673184

[2] M. Conti, N. Dragoni, and V. Lesyk, "A survey of man-in-the-middle attacks," IEEE Communications Surveys & Tutorials, vol. 18, no. 3, pp. 2027–2051, 2016. https://doi.org/10.1109/COMST.2016.2548426

[3] T. Kiravuo, M. Särelä, and J. Manner, "A survey of Ethernet LAN security," IEEE Communications Surveys & Tutorials, vol. 15, no. 3, pp. 1477–1491, 2013. https://doi.org/10.1109/SURV.2012.121112.00190

[4] W. Stallings, Network Security Essentials: Applications and Standards, 6th ed. Boston, MA, USA: Pearson, 2017. https://api.pageplace.de/preview/DT0400.9781292154916_A37747529/preview-9781292154916_A37747529.pdf

[5] D. Dolev and A. C. Yao, "On the security of public key protocols," IEEE Transactions on Information Theory, vol. 29, no. 2, pp. 198–208, 1983. https://doi.org/10.1109/TIT.1983.1056650

[6] A. S. Tanenbaum and D. J. Wetherall, Computer Networks, 5th ed. Boston, MA, USA: Pearson Education, 2011. https://www.inf.ufsc.br/~bosco.sobral/ensino/ine5645/Computer-Networks---A-Tanenbaum---5th-edition.pdf

[7] A. Mallik, "Man-in-the-middle-attack: Understanding in simple words," Cyberspace: Journal Pendidikan Teknologi Informasi, vol. 2, no. 2, pp. 109–134, 2018. https://www.researchgate.net/publication/335385872_MAN-IN-THE-MIDDLE-ATTACK_UNDERSTANDING_IN_SIMPLE_WORDS

[8] A. A. M. Amin and M. S. Mahamud, "An alternative approach of mitigating ARP-based man-in-the-middle attack using client-side bash script," in Proceedings of the 2019 6th International Conference on Electrical and Electronics Engineering (ICEEE), IEEE, 2019. https://doi.org/10.1109/ICEEE2019.2019.00029

[9] S. Hijazi and M. S. Obaidat, "A new detection and prevention system for ARP attacks using static entry," IEEE Systems Journal, vol. 13, no. 3, pp. 2732–2738, 2019. https://doi.org/10.1109/JSYST.2018.2880229

[10] J. S. Meghana, T. Subashri, and K. R. Vimal, "A survey on ARP cache poisoning and techniques for detection and mitigation," in Proceedings of the 2017 4th International Conference on Signal Processing, Communications and Networking (ICSCN), IEEE, 2017. https://doi.org/10.1109/ICSCN.2017.8085417

[11] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 303–336, 2014. https://doi.org/10.1109/SURV.2013.052213.00046

[12] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in the Internet of Things," Journal of Network and Computer Applications, vol. 84, pp. 25–37, 2017. https://doi.org/10.1016/j.jnca.2017.02.009

[13] S. Axelsson, "Intrusion detection systems: A survey and taxonomy," Technical Report, Chalmers University of Technology, 2000. https://www.researchgate.net/publication/2597023_Intrusion_Detection_Systems_A_Survey_and_Taxonomy