*Received: 20 November 2022* | *Revised: 16 January 2023* | *Accepted: 20 January 2023*
*DOI: 10.37917/ijeee.19.2.2* *Vol. 19 | Issue 2 | December 2023*

*Open Access*

Iraqi Journal for Electrical and Electronic Engineering
*Original Article*

IJEEE University of Basrah College of Engineering

# An Effective Approach to Detect and Prevent ARP Spoofing Attacks on WLAN

**Hiba Imad Nasser*, Mohammed Abdulridha Hussain**
Department of Computer Science - University of Basrah, Coolege of Education for Pure Science, Basrah, Iraq

Correspondance
*Hiba Imad Nasser
Department of Computer Science,
College of Education for Pure Science
University of Basrah, Basrah, Iraq
Email: eduppg.hiba.amad@uobasrah.edu.iq

**Abstract**
***Address Resolution Protocol (ARP) is used to resolve a host's MAC address, given its IP address. ARP is stateless, as there is no authentication when exchanging a MAC address between the hosts. Hacking tactics using ARP spoofing are constantly being abused differently; many previous studies have prevented such attacks. However, prevention requires modification of the underlying network protocol or additional expensive equipment, so applying these methods to the existing network can be challenging. In this paper, we examine the limitations of previous research in preventing ARP spoofing. In addition, we propose a defence mechanism that does not require network protocol changes or expensive equipment. Before sending or receiving a packet to or from any device on the network, our method checks the MAC and IP addresses to ensure they are correct. It protects users from ARP spoofing. The findings demonstrate that the proposed method is secure, efficient, and very efficient against various threat scenarios. It also makes authentication safe and easy and ensures data and users' privacy, integrity, and anonymity through strong encryption techniques.***

**Keywords**
**Network Security, ARP, ARP Spoofing, MITM.**

## I. Introduction

The current network environment is rapidly evolving, where ordinary users can easily access the information they need and receive services[1]. However, relying on such networks also causes related security issues. Various problems exist in network-based security incidents. Representative attack methods include ”man in the middle” (MITM), ”denial of service” (DoS), ”sniffing,” ”spoofing,” ”poisoning,” and ”session hijacking,” among many others. These methods cause organizations and people to lose their lives[2].

A wireless Local Area Network (WLAN) is a type of Local Area Network (LAN) network that connects users via radio frequency (RF) and infrared (IR) media[3]. Some business entities and even agencies have preferred to use WLAN technology because it is very easy to use. Still, only a few pay attention to data communication security on the wireless network[4],[5]. A ”Man in the Middle” (MITM) attack is one type of attack on a local area network (LAN) or WLAN with open access. This method allows attackers to sniff data frames, modify traffic, or even stop traffic (ARP poisoning). The basic concept of ARP poisoning, or spoofing, is to use the ARP cache to provide a fake identity, server address, or gateway to network users so that the attacker can modify network traffic[6]. If left unchecked, such attacks can disrupt network traffic so that they can disconnect the internet on devices connected to the network[7].

In addition, the ARP spoofing attack technology can perform a spoofing attack in a switching (local) environment. However, most countries believe they are safe from these attacks in a switching environment[8]. Automated and advanced tools are produced and distributed without permission. The lack of awareness of the current problem increases the possibility of security incidents. ARP (Address Resolution Protocol)have been known to have flaws since 1982, but ARP spoofing is still being used to cause damage today, nearly 40 years later[9].

Although many studies of these security threats have been conducted long, expensive equipment is required, so it is installed and operated only in a particular organization, static table creation and management [10]. The current network system's implementation was either impossible or limited. Therefore, this paper describes the process of an ARP poisoning attack and then suggests a way to defend against it. That is, an ARP poisoning attack changes the ARP cache table information by repeatedly sending an abnormal ARP response packet to the target of the attack.

During this process, The ARP request packet is being monitored [11]. An ARP poisoning attack is found and stopped by analyzing and validating the information in the request packet before sending responses[12].

*Problems statement*

Layer 2 connections are the most susceptible. Layer 2 protocols are risky since they don't ensure a reliable IP-to-MAC connection. Layer 3 devices, like routers, link multiple subnets to allow nodes end-to-end Internet access; hence, their security is a primary consideration when putting up a network.ARP translates IP to MAC addresses. ARP may be used for spying and spoofing, but the protocol was created with security features to avoid such attacks. This section shows ARP's vulnerabilities[13].

• Stateless: ARP is a stateless protocol; thus, it sends reply packets even if it hasn't done any ARP inquiries and doesn't verify packet validity. This allows the attacker to transmit bogus ARP reply packets to the victim using his MAC address.

• No authentication: ARP depends on a secure LAN; therefore, it's impossible to determine whose host delivered the ARP message. ARP spoof.

• ARP cache table update: The ARP table will rapidly update its cache of IP-to-MAC mappings when an ARP request or reply is received; however, the correctness of this information is not validated. Reduces network traffic. Changing your IP address will clear the cache.

Contributions to our proposed scheme to prevent ARP spoofing attacks:.

[1] Without using encryption, which could slow ARP, the attack could still be found, and alerts were sent to the administrator about the attack scenario on time.

[2] Using packet filtering in conjunction with network traffic monitoring to prevent ARP spoofing attacks without negatively impacting the speed at which ARP request and reply exchanges occur.

[3] The suggested solution doesn't require any extra networks or hardware. It also doesn't change the protocol. As a result, it's cheap and doesn't overburden the system or network.

[4] Consider all ARP assaults; periodically check the ARP cache table, and maintain an encrypted blacklist of all potential attackers so our proposal resists DOS, impersonation, reply, and inside attacks.

[5] A technique of customizing the device's dynamic tables and entries without respect for the router's capabilities.

This paper details a basic client-side ARP spoofing detection and prevention approach. The algorithm filters packets before issuing ARP requests or replies. Furthermore, it offers a defensive method that does not require network protocol upgrades or costly equipment. Verifying MAC and IP addresses before sending or receiving packets prevents ARP spoofing. Moreover, it enables safe and rapid authentication and maintains data and users' confidentiality, integrity, and anonymity via efficient encryption algorithms. Its efficacy seems to be equivalent to that of other methods. The sections are given in the following order: Section 2 examines the relevant literature. The background should be clarified in Section 3. Section 4 discusses the approach used. In Section 5, the results and testing of the proposed approach are discussed. Section 6: Security Performance and Analysis Section 7 ends with a summary and conclusion.

## II. RELATED WORK

Prabhadevi. B [14]GNS3, the Ettercap, and Wincap packet analyzers are also described in this work. This architecture uses IP-MAC table comparisons between Ethernet and ARP headers, and incorrect entries are added to a spoofed database. Every 10 minutes, the gateway gets messages to empty the cache, warn about cache poisoning, and add fake information to the fake list. New hosts need an updated ARP table. This method uses phoney data to identify attacks. Still, since the list's storage locations are unknown, the approach is subject to attack and wastes time because it doesn't function with a real network.

Hijazi et al.[15] The studies explored many ARP difficulties and suggested ways for spotting and blocking these attacks. The recommended cure includes a static ARP table entry technique, type comparison, IP-Mac-based input detection, and preventive measures. For system security, the IP-Mac address in ARP must match. So you can work, remove the false IP-Mac. The proxy server uses the patch file to address and correct ARP security problems. This strategy has been proven effective; it requires no additional resources and provides essential solutions on a small network. This technique can add static ARP entries but not dynamic ones.

Majumdar et al.[16] The Python programming language was used to develop a tool for ARP poisoning and spoofing.

For attack detection, a Python script utilizing the Scapy module is used. After the contents of the request packet have been verified, an alert is sent if the original Mac does not match the reply Mac, and a static item is stored in a cache table as a security precaution. The proposed method does not permit dynamic entries. Rather than creating a script for spoofing, the existing available tools may be utilized to undertake Kali Linux assaults and packet analysis.

Alsukkar et al. [17]The researchers suggest a few methods, one of which is using an application that alerts the user of an ARP spoofing attack by displaying the attacker's IP-Mac address to identify and defend against MITM assaults. Each network node's genuine Mac address and the results of a ping ICMP message are sent to the router for additional security. There has been a reversion to the previous ARP table configurations. The application is loaded on the administrator's and the users' computers. Additionally, the network configuration may attack and corrupt the ARP table. Both programmes proved valuable and workable, and they were both developed in Python for use only on the Linux platform. The proposed investigation into alternative OSes, however, has been ineffective. Sending packets to all devices per second causes a lot of network traffic.

Mahendra [18]The researcher's method improves the ARP table static input process by reducing the time-consuming manual entry methods and automatically evaluating the correctness of data entered into the static table. The strategy worked well by using the static record feature of some operating systems, such as Linux and Windows, to create a semi-static table for the cache. In a virtual network, packets are sent, and the validity of the responses is verified before they are added to the table using an ARPing tool. As a result, the suggested solution does not ensure the safety of all network users. Instead, it must give a white list of trustworthy IP-MAC addresses that may be used to evaluate the ARPing tool's ability to add to the static record.

Rupal et al.[19]This article explains how an authentication tool may also be used to identify and avoid ARP poisoning in a dynamic IP configuration. The primary ARP cache stores the authentication information for IP-MAC pairings, which is subsequently saved in a text file in the secondary ARP cache. One server sets up IP DHCP, another uses MySQL and a database to authenticate users, and the third server monitors for and prevents cache poisoning attacks. To detect and prevent attacks, all devices receive an ICMP broadcast request and reply message. Due to its failure to transmit a response, the IP-Mac is deleted from the secondary and primary ARP caches. On the other hand, broadcasting requests make the system less useful because the network gets clogged up, the server has to be managed by authorized people, and only reliable storage facilities can be used.

## III. Background

### *A.* Address Resolution Protocol

The Address Resolution Protocol (ARP) is one of the major protocols in the TCP/IP suite[20]. ARP aims to map an IPv4 address to a physical address. Network applications at the application layer use IPv4 addresses to communicate with other devices. But at the data link layer, the address is a MAC address permanently burned into the network card. ARP is used to determine the MAC address of a device on your LAN as well as the corresponding IPv4 address with which a network application is attempting to communicate[21].

In contrast, static mapping requires the creation of a database that connects a logical address with a physical address[22]. This table is saved on each network computer. Each device that knows the IP address but not the physical address of another device may look it up in the database. The static mapping table must be routinely updated when devices' MAC addresses change. This may impact network performance[23]. Dynamic mapping implements a distinct situation. When a computer knows the logical address of another device, it may use a protocol to determine the device's physical address. Two protocols, ARP and Reverse(RARP), have been created to accomplish dynamic mapping (RARP)[24]. ARP converts logical addresses to physical addresses, whereas RARP converts physical addresses to logical addresses. Since the ARP protocol has been the focal point of this thesis, we shall confine our discussion to that protocol[24].

### *B.* ARP Spoofing Attack

One kind of MITM attack is called ARP spoofing, and it involves an attacker sending fake ARP packets to a local area network[25]. The attack aims to have the victim's network interface card (MAC) address matched to the IP address of another computer (such as the default gateway), redirecting all communication intended for the other machine to the attacker. If an attacker can fake an ARP, they may change network traffic or completely block it. Many additional attacks, including DOS attacks, MITM attacks, and session hijacking attacks, take advantage of this vulnerability[26].

ARP spoofing is an attack method that takes advantage of the fact that the integrity check of the ARP message is not guaranteed and sends a fake ARP response packet to the target to trick the MAC address and stop normal operation[27].

ARP poisoning repeatedly sends abnormal ARP response packets to the target. Fig.1 and Fig.2 assume that the host's IP address (1) is '0.0.0.1' and the MAC address is AA.AA', and the host's IP address (2) is '0.0.0.2' is 'CC.CC'. Fig.1 shows the normal communication flow before the ARP spoofing attack occurs. When attempting to communicate from Host (1) to Host (2), a MAC address is required, and an ARP request message is sent in the Broadcasting method to locate a host with the IP address '0.0.0.2'. Host(2) with IP address '0.0.0.2' transmits an ARP response message to the host (1) to inform it of its MAC address information. After that, using the ARP cache table, hosts (1) and (2) perform a typical communication process, and Fig.2 shows the communication flow after an ARP spoofing attack occurs.Fig.1 shows an ARP request message to find the Host's MAC address. At this time, the attacker deceives host (1) that the MAC address of host (2) is 'CC.CC' and host (2) that the MAC address of the host (1) is 'CC.CC'. As a result, hosts (1) and (2) appear to communicate generally with each other, but in fact, they communicate with the attacker, resulting in information leakage to the attacker.

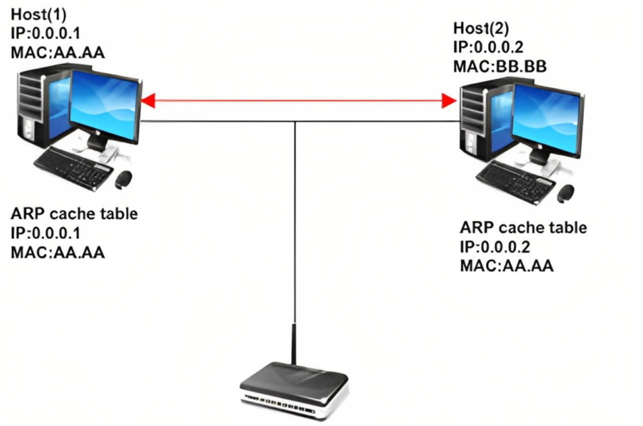


Fig. 1. Pre-ARP Spoofing Communication

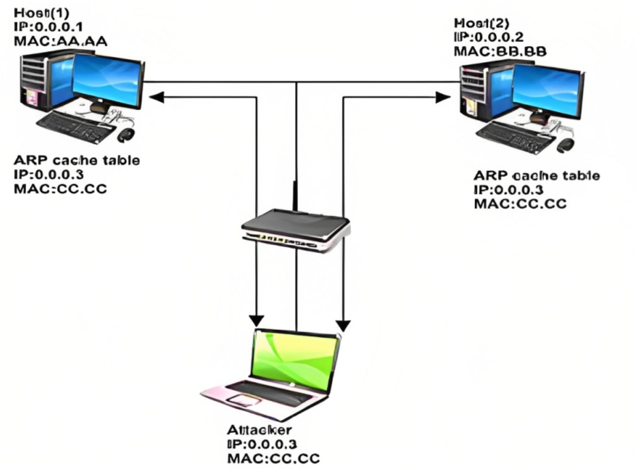


Fig. 2. ARP Spoofing communication

## IV. Proposal approach

### A. Defense Scenario and the algorithm

This algorithm protects all devices in the network from ARP cache poisoning. An attacker trying to perform an ARP cache-forcing attack will spoof the source IP address in the ARP packet with the victim's IP address and the source MAC address with the attacker's MAC address. If the victim receives an IP address via an ARP request or response, the IP address assigned to the victim's machine's MAC address will be in the ARP table. When the addresses in the ARP table do not match the addresses in the spoofed ARP packet, one model is called to detect the attack and another to prevent and respond. Algorithm1and Fig.3 show the suggested approach.

**Algorithm 1: proposal algorithm**

```
Begin:
Input: ARP request packet
Output: ARP reply packet.
Before sending ARP Reply Frame, the dest. host will
check the MAC address and HDD No.of the source host in blackList.
If: (MacSource and HDDserial in a blacklist), then
    Display the alert message "this device is blocked" and drop a packet
Else:
    If: ( the packet is an ARP packet ), then
        If: ARP reply (op=2) or ARP request (op=1), then
            get the original mac of the sender from ARPtable and response mac from the ARP Reply packet
            If: (NOT match), then
                detection(responseip,originalmac)
                response(blackList,responsemac, HDDserial)
            Else:
                ARP Reply/Request Frame will be sent to the source host.
            End if
        End if
    End if
End if
(pause 1 second)ARP new entry
End:
```

### B. Detection model

This model gets the IP-MAC, the target address: If the firewall has not blocked the attacker's previously, then we may reassign its actual addresses to the target device (as well as the gateway), send an alert to the victim, put a stop to the assault, and return the network to its normal functioning state. This is done by blocking the attacker from the firewall and then transmitting the original information (the actual IP and MAC of the "host IP") to the "target IP." the detection Algorithm 2 And Fig.4 Show how the model works.

**Algorithm 2: detection algorithm**

```
Begin:
Input: response ip, originalmac.
```

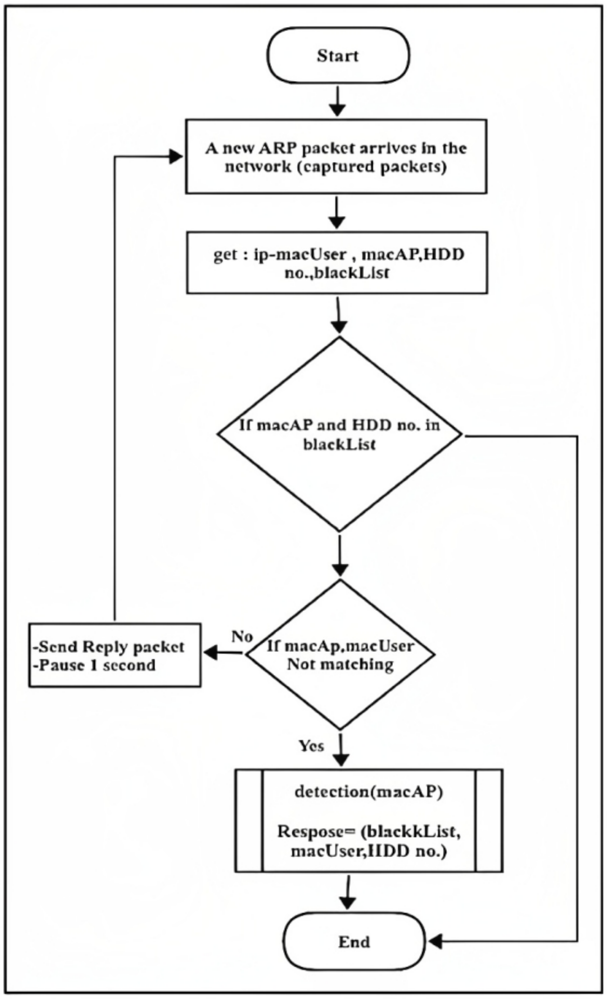

Start
A new ARP packet arrives in the network (captured packets)
get : ip-macUser , macAP,HDD no.,blackList
If macAP and HDD no. in blackList
No
-Send Reply packet
-Pause 1 second
If macAp,macUser Not matching
Yes
detection(macAP)
Respose= (blackkList, macUser,IIDD no.)
End

Fig. 3. the proposal flowchart

```
Output: alarm message.
restore the original Mac for the ARP cache table and router
show alert message in dest. Host 'under attack.'
check the response ip address of the source host in its firewall.
if (response ip in a firewall), then
    show alert message this IP is blocked
Else:
    blocking this attacker's IP from all protocols in dest. Host from the firewall
End if
End:
```

### *C.* Response model

The response model is used if the main program verifies that the IP-MAC packet input information does not match the IP-MAC in the ARP table. An if statement checks whether the attacker's IP-MAC address is on the blacklist after decrypting the blacklist. A notice saying "The MACspoof address has been banned" will appear if it is. If it doesn't, the attacker's device will be attacked in reverse to cut it off from the service, the file will be re-encrypted after the update, and the device will be blacklisted. Below, the response algorithm in Algorithm 3 and the flowchart in Fig.5 explain their function.

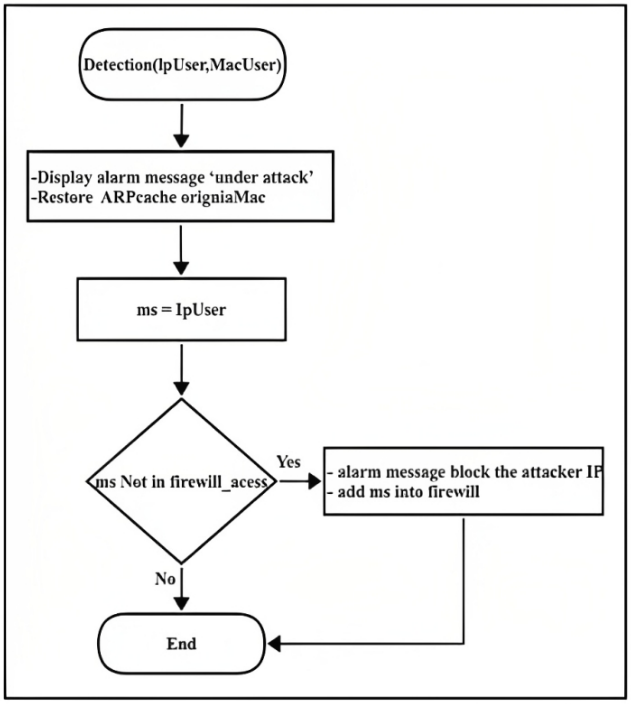

Detection(IpUser,MacUser)
-Display alarm message 'under attack'
-Restore ARPcache origniaMac
ms = IpUser
ms Not in firewill_acess
Yes
- alarm message block the attacker IP
- add ms into firewill
No
End

Fig. 4. the detection flowchart

**Algorithm 3: response algorithm**

```
Begin:
Input: (blacklist, original_mac, HDDserial)
Output: Encrypt (blacklist)
If (original_mac and HDDno. not in blacklist), then
    Add original_mac and HDDno. to blacklist
    Display alert message "added the device to blacklist."
    Decrypt (blacklist)
Else:
    Display alert message "the macSpoofed is already in a blacklist."
End if
do a reverse attack on the attacker host to cut service
A flood attack on an attacker host
End:
```

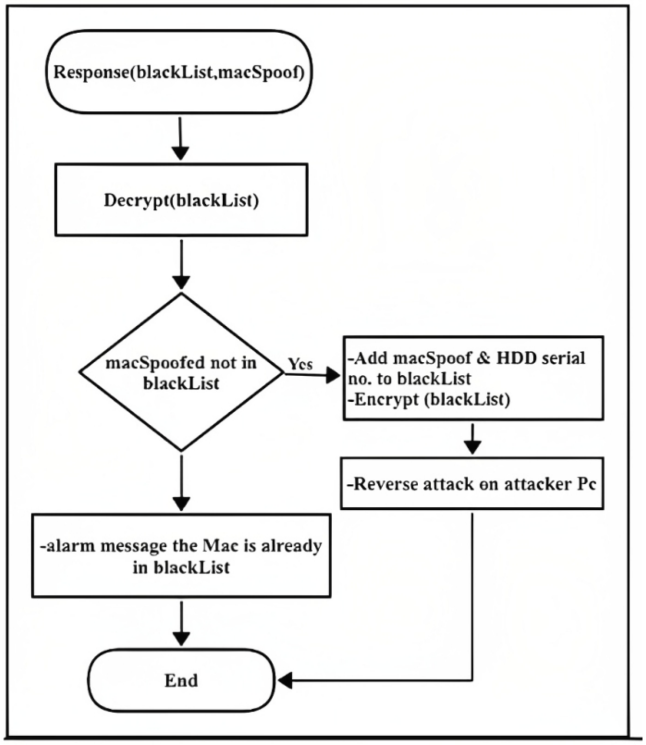

Response(blackList,macSpoof)
Decrypt(blackList)
macSpoofed not in blackList
Yes
-Add macSpoof & HDD serial no. to blackList
-Encrypt (blackList)
-Reverse attack on attacker Pc
-alarm message the Mac is already in blackList
End

Fig. 5. the response flowchartt

## V. Experimental results

### *A.* Implementation

The suggested defense against ARP spoofing doesn't require modifying the ARP protocol or imposing a particular topology. The network must not be expanded with new hardware or software to implement the protection mechanism. As a result, it is portable, quick, and can quickly identify an attack.

PyCharm must be installed on two PCs with Intel(R) Core(TM) i5-3320M CPUs running at 2.60 GHz, 8 GB of RAM, Windows 10 64-bit, and a router. Since Kali Linux has tools for testing, creating networks, and finding system security vulnerabilities, we used the ARPspoof-tool to simulate the attack scenario. Fig.6 shows our strategy in action.

### *B.* Results

During the experiment, initiate an ARP spoofing attack using the ARP Spoof tool in Kali Linux. The purpose of sending ARP response packets across networks is to attack the victim's ARP table. And after the attack was successful because the spy, in this case, could hear and change what the victim and the network device were saying to each other. In fig.7 shows the damaged ARP cache table after the attack scenario has been executed.

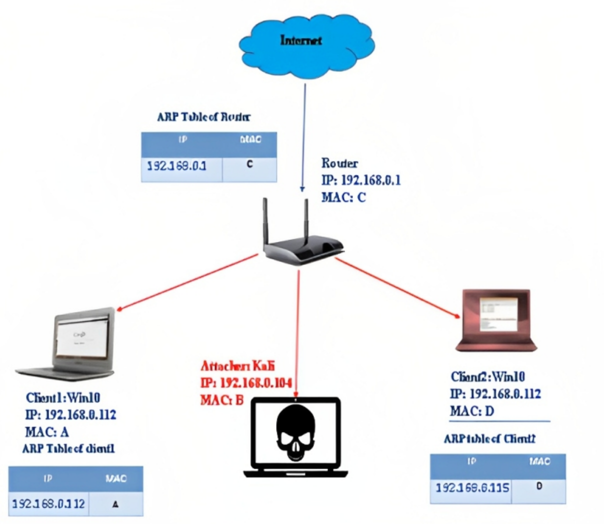
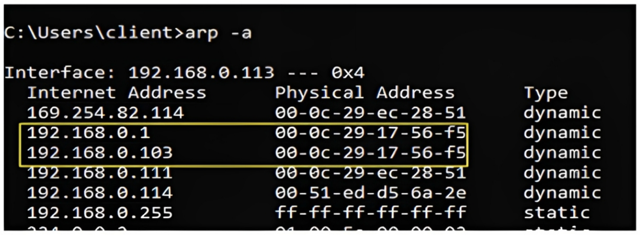


Fig. 6. Network topology

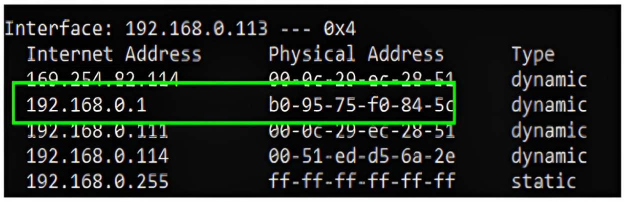


```
C:\Users\client>arp -a

Interface: 192.168.0.113 --- 0x4
  Internet Address      Physical Address      Type
  169.254.82.114        00-0c-29-ec-28-51     dynamic
  192.168.0.1           00-0c-29-17-56-f5     dynamic
  192.168.0.103         00-0c-29-17-56-f5     dynamic
  192.168.0.111         00-0c-29-ec-28-51     dynamic
  192.168.0.114         00-51-ed-d5-6a-2e     dynamic
  192.168.0.255         ff-ff-ff-ff-ff-ff     static
```

Fig. 7. ARPtable under attack

After that, try out the defense on Client 1. When an ARP cache table is poisoned, the script may restore it and block further attempts. Client 2, which does not use the protection script, will still be susceptible to ARP spoofing because of its unprotected ARP cache table. Once the code has been activated, the ARP cache table and the result of the proposed approach are shown in Figures below.

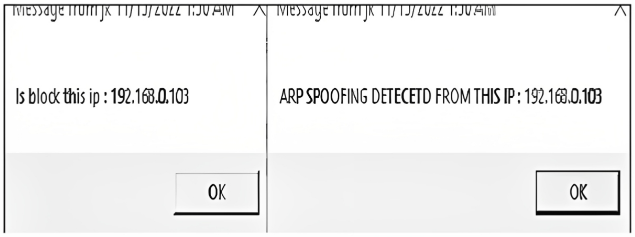


```
Interface: 192.168.0.113 --- 0x4
  Internet Address      Physical Address      Type
  169.254.82.114        00-0c-29-ec-28-51     dynamic
  192.168.0.1           b0-95-75-f0-84-5c     dynamic
  192.168.0.111         00-0c-29-ec-28-51     dynamic
  192.168.0.114         00-51-ed-d5-6a-2e     dynamic
  192.168.0.255         ff-ff-ff-ff-ff-ff     static
```

Fig. 8. ARPtable the proposal.

### *C.* Discussion

Ten different ARP spoofing attacks against the network Architectures defined in Section 5 were attempted in this test. The arpspoof tool is used in this experiment to do ARP spoofing, and the time was calculated using Python's time



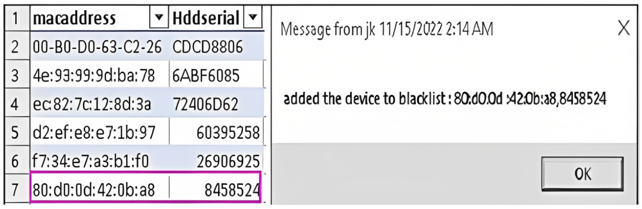
Message from jk 11/15/2022 1:30 AM

Is block this ip : 192.168.0.103

Message from jk 11/15/2022 1:30 AM

ARP SPOOFING DETECETD FROM THIS IP : 192.168.0.103

Fig. 9. Result detection algorithm.



| 1 | macaddress | Hddserial |
|---|---|---|
| 2 | 00-B0-D0-63-C2-26 | CDCD8806 |
| 3 | 4e:93:99:9d:ba:78 | 6ABF6085 |
| 4 | ec:82:7c:12:8d:3a | 72406D62 |
| 5 | d2:ef:e8:e7:1b:97 | 60395258 |
| 6 | f7:34:e7:a3:b1:f0 | 26906925 |
| 7 | 80:d0:0d:42:0b:a8 | 8458524 |

Message from jk 11/15/2022 2:14 AM

added the device to blacklist : 80:d0:0d :42:0b:a8,8458524

Fig. 10. Result response algorithm.

function. The results of 10 experiments measuring detection speed are as shown in fig.13.

The figure 13 shows how fast the method can find an attack on a protected network. Based on the results of the previous trial, it was determined that the median time to detect an attack was 0.933 seconds. Here's how fast the system can get back to normal after an attack has been stopped.

The average response time was 3.05 seconds in the prior experiment. Our proposed approach was tested intensively and found to be relatively secure against cyberattacks.

## VI. Security analysis

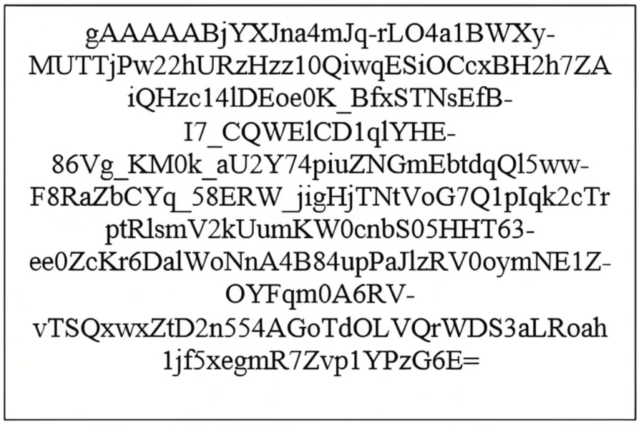
This section analyzes our proposal and discusses how our approach may successfully withstand common harmful approaches. Also, the method suggested is safe, and a comparison of technologies that are almost the same is given.

**Theorem1. Our proposed approach can resist a MITM attack.**

**Proof:** In a MITM attack on a local network, the first step is to change the ARP table. The attacker pretends to be the Mac of any device on the network (a victim) so that they can make fake requests, get real answers, and get to the victim's sensitive information. Before sending responses to requests sent and received over a network, the proposed method checks and filters them to ensure they are real. As shown in steps (7-9) of the algorithm (1), algorithm (2), and Figures 3 and 4, when the computer finds fake requests, it alerts the victim of an attack.

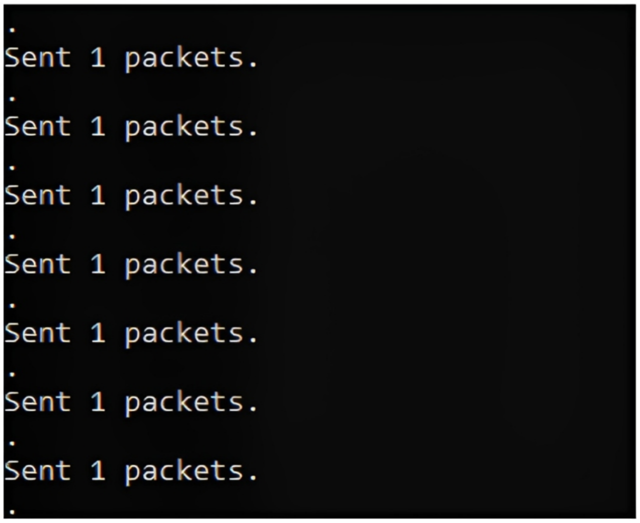
**Theorem2. Our proposed approach resists insider attacks.**

**Proof:** Insider attacks are conducted by current or former users with network access, gateway access, and the ability to perform fraudulent actions using other users' identities. This attack is mitigated by utilizing Fernet encryption to secure database records.



gAAAAABjYXJna4mJq-rLO4a1BWXy-
MUTTjPw22hURzHzz10QiwqESiOCcxBH2h7ZA
iQHzc14lDEoe0K_BfxSTNsEfB-
I7_CQWElCD1qlYHE-
86Vg_KM0k_aU2Y74piuZNGmEbtdqQl5ww-
F8RaZbCYq_58ERW_jigHjTNtVoG7Q1pIqk2cTr
ptRlsmV2kUumKW0cnbS05HHT63-
ee0ZcKr6DalWoNnA4B84upPaJlzRV0oymNE1Z-
OYFqm0A6RV-
vTSQxwxZtD2n554AGoTdOLVQrWDS3aLRoah
1jf5xegmR7Zvp1YPzG6E=

Fig. 11. Result decrypt_blacklist.



```
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
```

Fig. 12. Result reverse_attack.

**Theorem3. Our proposed approach resists DOS attacks.**

**Proof:** After getting the router's IP-MAC address, the attacker sends fake traffic or a lot of requests, which breaks the service. The method described here checks the blacklist, which contains all of the attackers' addresses, and if it finds the request, it ignores it and does not filter any network traffic related to the ARP table attack, as shown in steps 3-9 of the algorithm1. It also prevents the false IP-MAC from being used by all protocols and ports that the attacker may use to DOS the victim, preventing it from sending fake requests.

**Theorem4. Our proposed approach resists mac cloning**

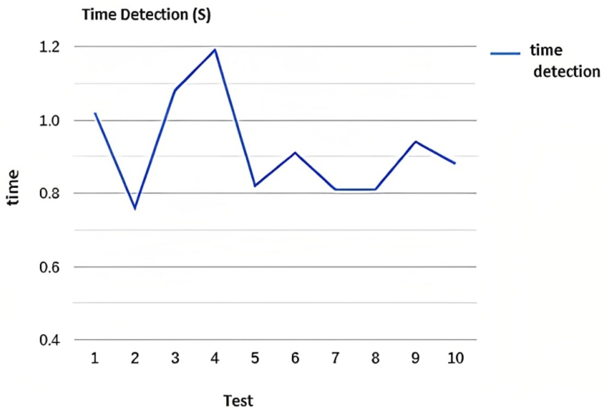


Fig. 13. Detection time.

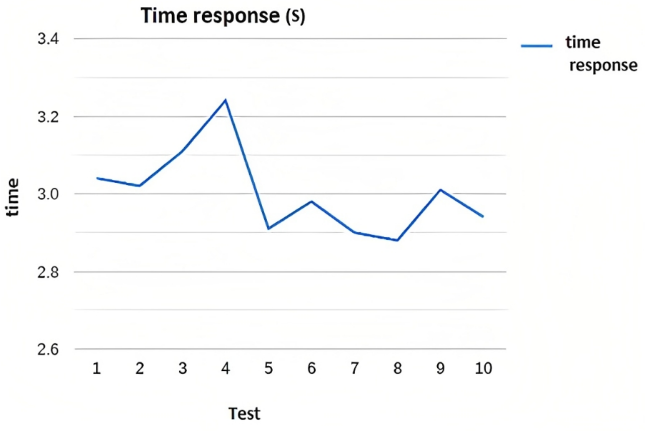


Fig. 14. Response time.

**attacks.**

**Proof:** This attack is straightforward since the attackers know and exploit ARP protocol weaknesses. The recommended approach prevents spoofing attacks by Algorithm1 creating two-factor authentication using a combination of Mac and HDD serial numbers and Algorithm2 validating the request and reply packets to guarantee they are safe against spoofing before transmission and reception. We also use Fernet encryption to keep unauthorized people from entering the database with verified IDs.

*Comparison with related works*

This section compares our proposed technique to prior studies. Tabel.I shows a direct comparison of many standard authentication methods:

C1: Static ARP;C2: ARP protocol;C3: Full Prevention of ARP spoofing;C4: Scalability;C5: Automation;C6: Authentication;C7: Cryptographic;C8: Cost Effective;C9: Time effective

Following are relevant comparisons of several conventional assault techniques, as seen in Tabel.II.

TABLE I.
THE MAIN METHODS

| **Methods** | **C1** | **C2** | **C3** | **C4** | **C5** | **C6** | **C7** | **C8** | **C9** |
|---|---|---|---|---|---|---|---|---|---|
| Proposed | $x$ | ✓ | ✓ | ✓ | $x$ | ✓ | ✓ | ✓ | ✓ |
| Filter packet-tool | ✓ | ✓ | ✓ | ✓ | $x$ | $x$ | $x$ | ✓ | ✓ |
| executable(batch) file | ✓ | ✓ | $x$ | ✓ | $x$ | $x$ | $x$ | ✓ | $x$ |
| ARPing tool-semi Static | ✓ | ✓ | $x$ | ✓ | $x$ | $x$ | $x$ | ✓ | $x$ |
| Fake list-GNS3 | $x$ | ✓ | ✓ | ✓ | $x$ | $x$ | $x$ | ✓ | $x$ |
| Defense tool | ✓ | ✓ | ✓ | ✓ | ✓ | $x$ | $x$ | ✓ | $x$ |
| SDN (IDPs) | $x$ | $x$ | ✓ | ✓ | ✓ | ✓ | $x$ | $x$ | $x$ |
| ARP server application | $x$ | ✓ | ✓ | ✓ | ✓ | ✓ | $x$ | $x$ | $x$ |
| Dynamic IP configuration | ✓ | ✓ | $x$ | ✓ | $x$ | ✓ | ✓ | ✓ | $x$ |
| MD5 method to create IP-MAC database | $x$ | $x$ | ✓ | ✓ | $x$ | ✓ | ✓ | ✓ | $x$ |
| Sender&reciver system | $x$ | $x$ | ✓ | ✓ | $x$ | ✓ | ✓ | ✓ | $x$ |

TABLE II.
THE MAIN ATTACKS COMPARISONS.

| **Methods** | **T1** | **T2** | **T3** | **T4** | **T5** |
|---|---|---|---|---|---|
| Our proposed | Yes | Yes | Yes | Yes | Yes |
| Filter packet-tool | Yes | No | No | No | Yes |
| executable(batch) file | Yes | Yes | Yes | No | No |
| ARPing tool-semi Static | Yes | No | Yes | Yes | Yes |
| Fake list-GNS3 | Yes | No | No | No | No |
| Defense tool | Yes | Yes | Yes | No | No |
| SDN (IDPs) | Yes | No | Yes | No | No |
| ARP server application | Yes | Yes | Yes | No | No |
| Dynamic IP configuration | Yes | No | No | No | No |
| MD5 method to create IP-MAC database | Yes | Yes | No | Yes | Yes |
| Sender&reciver system | Yes | No | Yes | No | Yes |

T1: Withstand MITM attack;
T2: Withstand insider attack;
T3: Withstand replay attack;
T4: Withstand user's mac cloning attack;
T5: Withstand DOS attack;

## VII. Conclusion

WLANs have quickly become a significant part of people's daily lives, making it crucial to provide users with secure connectivity that is also simple to set up. Like all other types of network security, wireless security involves protecting against unwanted access to or destruction of machines or data inside a wireless network.ARP spoofing is a severe problem in LAN security. Despite the fact that various proposals for solving the issue have lately been made, we have highlighted the flaws with those ideas in this article, such as the fact that they do not work, are too costly, or are difficult to manage.Furthermore, we have proposed an ARP spoof detection and protection approach with none of the above drawbacks. In addition, we have created a software prototype corresponding to the approach and tested it on the suggested network's architecture. The experimental findings demonstrate the positive features of our approach and demonstrate that it is a good and effective solution that is cost-effective in terms of hardware and useful for avoiding and preventing man-in-the-middle assaults in a timely manner. In the future, we want to conduct comprehensive testing of our approach on real-world networks (e.g., college or university networks, student housing networks, and workplace and business networks) in order to get honest user feedback.

## CONFLICT OF INTEREST

The authors have no conflict of relevant interest to this article.

## REFERENCES

[1] Y. Li, D. Li, W. Cui, and R. Zhang, "2011 ieee 3rd international conference on communication software and networks," *CCSN 2011*, pp. 554–557.

[2] M. Fareed and A. A. Yassin, "Privacy-preserving multi-factor authentication and role-based access control scheme for the e-healthcare system," *Bull. Electr. Eng. Informatics*, vol. 11, no. 4, pp. 2131–2141, Aug. 2022.

[3] H. A. Noman, S. M. Abdullah, and H. I. Mohammed, "An automated approach to detect deauthentication and disassociation dos attacks on wireless 802 . 11 networks," *IJCSI International Journal of Computer Science Issues*, vol. 12, no. 4, pp. 107–112, 2015.

[4] A. A. Galal, A. Z. Ghalwash, and M. Nasr, "A new approach for detecting and mitigating address resolution protocol (arp) poisoning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 6, 2022.

[5] Z. Abduljabbar, H. Jin, D. Zou, A. A. Yassin, Z. Hussien, and M. A. Hussain, "An efficient and robust one-time message authentication code scheme using feature extraction of iris in cloud computing," *Proc. 2014 Int. Conf. Cloud Comput. Internet Things, CCIOT 2014*, pp. 22–35, 2014.

[6] A. S. Yadav, P. M. Natu, D. M. Sethia, A. B.Mundkar, and S. S. Sambare, "Prevention of spoofing attacks in wireless networks," *International Conference on Computing Communication Control and Automation*, pp. 164–171, Feb. 2015.

[7] V. Rohatgi and S. Goyal, "A detailed survey for detection and mitigation techniques against arp spoofing," *4th International Conference on IoT in Social, Mobile, Analytics and Cloud, ISMAC 2020*, pp. 352–356, Oct. 2020.

[8] S. Duddu, A. Rishitasai, C. L. S. Sowjanya, G. R. Rao, and K. Siddabattula, "Secure socket layer stripping attack using address resolution protocol spoofing," *4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 973–978, May 2020.

[9] J. S. Meghana, T. Subashri, and K. R. Vimal, "A survey on arp cache poisoning and techniques for detection and mitigation," *4th International Conference on Signal Processing, Communication and Networking (ICSCN)*, pp. 1–6, Mar. 2017.

[10] R. Kumar, S. Verma, and G. S. Tomar, "Thwarting address resolution protocol poisoning using man in the middle attack in wlan," *Int. J. Reliab. Inf. Assur.*, vol. 1, no. 1, pp. 7–18, Dec. 2013.

[11] J. Xia, Z. Cai, G. Hu, and M. Xu, "An active defense solution for arp spoofing in open flow network," *Chinese Journal of Electronics*, vol. 28, no. 1, pp. 172–178, 2019.

[12] M. Alzuwaini and A. Yassin, "An efficient mechanism to prevent the phishing attacks," *Iraqi J. Electr. Electron. Eng.*, vol. 17, no. 1, pp. 1–11, Jun. 2021.

[13] S. Mahmood, S. M. Mohsin, and S. M. A. Akber, "Network security issues of data link layer: An overview," *3rd Int. Conf. Comput. Math. Eng. Technol. Idea to Innov. Build. Knowl. Econ. iCoMET 2020)*, March, 2020.

[14] B. Prabadevi and N. Jeyanthi, "A framework to mitigate arp sniffing attacks by cache poisoning," *Int. J. Adv. Intell. Paradig.*, vol. 10, no. 1/2, p. 146, 2018.

[15] 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.

[16] A. Majumdar, S. Raj, and T. Subbulakshmi, "Arp poisoning detection and prevention using scapy," *J. Phys. Conf. Ser.*, vol. 1911, no. 1, May 2021.

[17] G. A. Sukkar, R. Saifan, S. Khwaldeh, M. Maqableh, and I. Jafar, "Address resolution protocol (arp): Spoofing attack and proposed defense," *Commun. Netw.*, vol. 8, no. 3, pp. 118–130, 2016.

[18] M. Data, "The defense against arp spoofing attack using semi-static arp cache table," *3rd International Conference on Sustainable Information Engineering and Technology, SIET 2018 - Proceedings*, pp. 206–210, 2018.

[19] D. R. Rupal, D. Satasiya, H. Kumar, and A. Agrawal, "Detection and prevention of arp poisoning in dynamic ip configuration," *IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology (RTEICT)*, pp. 1240–1244, May 2016.

[20] F. A. Barbhuiya, S. Biswas, and S. Nandi, "An active host-based intrusion detection system for arp-related attacks and its verification," *International Journal of Network Security and Its Applications*, vol. 3, no. 3, pp. 163–180, 2011.

[21] G. Jinhua and X. Kejian, "Arp spoofing detection algorithm using icmp protocol," *in 2013 International Conference on Computer Communication and Informatics*, vol. 3, no. 5, pp. 1–6, Jan. 2013.

[22] J. C. Lin, M. J. Koo, and C. S. Wang, "A proposal for a schema for arp spoofing protection," *Appl. Mech. Mater.*, vol. 284–287, no. 5, pp. 3275–3279, Jan. 2013.

[23] V. Ramachandran and S. Nandi, "Detecting arp spoofing: An active technique," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3803, no. 5, pp. 239–250, 2005.

[24] A. Samvedi, S. Owlak, and V. K. Chaurasia, "Improved secure address resolution protocol," pp. 201–211, June 2014.

[25] I. Martinovic, F. A. Zdarsky, A. Bachorek, C. Jung, and J. B. Schmitt, "Phishing in the wireless: Implementation and analysis," pp. 145–156, 2007.

[26] A. M. AbdelSalam, W. S. Elkilani, and K. M. Amin, "An automated approach for preventing arp spoofing attack using static arp entries," *International Journal of Advanced Computer Science and Applications*, vol. 5, no. 1, 2014.

[27] Z. Trabelsi and K. Shuaib, "Spoofed arp packets detection in switched lan networks," *SECRYPT 2006 - Int. Conf. Secur. Cryptogr. Proc.*, pp. 40–47, 2006.