



Defending a wireless LAN against ARP spoofing attacks using a Raspberry Pi

Hiba Imad Nasser*, Mohammed Abdulridha Hussain

Department of Computer Science, College of Education for Pure Science, University of Basrah, Basrah, Iraq.

ARTICLE INFO

Received 07 September 2022
Accepted 08 October 2022
Published 30 December 2022

Keywords :

Network Security, MITM, ARP, ARP Spoofing, Authentication, De-authentication attack.

ABSTRACT

The Address Resolution Protocol (ARP) is a protocol that converts Internet Protocol (IP) addresses to Media Access Control (MAC) addresses. Due to a security issue known as "Man in the Middle," identity theft is feasible using the ARP protocol. ARP spoofing is one of the weaknesses in wireless networks when an attacker effectively masquerades as a legitimate one. Spoofing attacks will reduce network performance and break several security measures. In networks that use MAC address-based filtering to verify clients, all a spoofer needs is an actual MAC address from an authorised client to gain an unfair advantage. The research recommends developing a security system recognising and preventing ARP spoofing attacks. This system detects ARP spoofing attempts by comparing the static MAC address of the original router to the router's MAC address in the ARP cache table. After detecting the attack using information collected from the router's MAC address in the ARP cache table, the system will conduct a de-authentication attack against the attacker's MAC address. If the attacker is disconnected from the WLAN, they cannot perform ARP spoofing attacks. This system is operated using a Raspberry Pi Model B. Most ARP spoofing attacks can be detected in 0.93 seconds, and responding takes 3.05 seconds.

Citation: H.I. Nasser, M.A. Hussain, J. Basrah Res. (Sci.) 48(2), 123 (2022).
[DOI:https://doi.org/10.56714/bjrs.48.2.12](https://doi.org/10.56714/bjrs.48.2.12)

1. Introduction

The Internet is necessary to create an era of information disclosure [1]. The main benefit of using the Internet is to exchange information among users. These benefits can be felt due to the existence of adequate supporting infrastructure. The Internet network infrastructure that connects Internet Service Providers (ISPs) with users is the main component of supporting facilities for internet use[2].

An internet communication network has many components, such as communication protocols, network hardware, and other supporting software[3]. Each of these components has a vital role in sending data over the network. The communication protocol is a component that regulates how communication between devices can be carried out[4].

*Corresponding author email : eduppg.hiba.amad@uobasrah.edu.iq



Data sent over the network has the potential to be stolen or intentionally damaged by irresponsible parties. One way that is used to do this is by exploiting vulnerabilities that exist in network communication protocols. Address Resolution Protocol (ARP) is a protocol that is very vulnerable to exploitation, especially on WLAN (Wireless Local Area Network) networks[5]. The exploitation of this protocol is known as ARP spoofing or ARP poisoning. Examples of attacks that exploit these protocols include Man in the Middle Attack (MiTM), denial of service (DoS) attacks, social engineering, Netcut, and others.

For this reason, there needs to be a security mechanism to minimise the risk of exploiting communication protocols in the network[6]. Based on the above background, the researcher is trying to create a method of security control in the form of software that can detect, mitigate, and secure WLAN users in the region from attacks using the ARP exploit[7]. This method compares the MAC address of the original router, which is stored in a static variable, with the MAC address of the router, which is dynamically stored in the ARP cache table. If the MAC addresses being compared do not match, an ARP spoofing attack can be indicated on the network. It implements the detection method. The method will detect the attack using a de-authorisation attack, remove the attacker's machine from the network based on the MAC address obtained from the ARP cache table, and return the table to its state before the attack. Then, the response method is implemented by adding the attacker's Mac to the list blocked to prevent it from entering the network again.

1.1 Problem Statement

The ARP is a standard component of every computer system. ARP packets convert IP addresses to MAC addresses when communicating on a local area network. Several vulnerabilities in the ARP protocol might be used in attacks[8]. The initial stage in a MitM attack is changing the ARP tables on the victim's PC and the router. ARP spoofing may be used to initiate DoS attacks on local area networks (LANs) by blocking the transmission of otherwise valid packets.

The formulation of the problem in this study:

1. Are passive detection and de-authentication attacks effective for securing WLAN networks from ARP spoofing attacks?
2. How much time does it take the system to detect and respond to ARP spoofing attacks on WLAN networks using passive detection and de-authentication attacks?

The experiment showed that the algorithm meets the stated purpose.

- Stop spoofing the hacker's MAC address by comparing it to its original address in the ARP table and de-authenticating it if they don't match.
- Use the attacker's list, which blocks each unauthorised user, to ensure the attacker can't get into the ARP caches of the nodes.
- It prevents attackers from accessing nodes' ARP caches by de-authenticating them, thus reducing the number of lost network packets.
- Utilising detection and response approaches keep attackers away from infrastructure networks without firewalls.
- The method used does not require any additional network resources.

Organising this paper: Section 2 describes the problem addressed by this paper and provides a Literature Review on the subject. Our research's algorithm and system architecture are presented in Section 3. Section 4 contains the efficacy test and study findings. In the concluding part, a summary of the results is provided.

2. Literature Review

2.1. Address Resolution Protocol

"ARP (Address Resolution Protocol) is a protocol that translates IP addresses into MAC addresses" [9]. When two or more hosts on a network want to communicate with each other, they need the MAC to address each host. Initially, no host has the MAC address of another host on the same network. The host that wants to initiate communication sends an ARP request packet containing the host's IP address and MAC address[10]. Along with the ARP request packet, there is a WHO_HAS packet that asks all hosts on the same network via IP broadcast who owns the IP of the destination host. The host with the destination IP then replies with an ARP reply packet containing the destination host's IP address and MAC address. With this protocol, each host has an IP address and MAC address to communicate[11].

"ARP does not have authentication facilities when communicating and, therefore, is very vulnerable to ARP spoofing attacks"[12]. It also makes it easy for attackers to forge ARP packets to attack targets. That way, the attacker can manipulate the target's communications on the network.

2.2. ARP Spoofing

According to Data [13], ARP spoofing is an attack carried out by making fake ARP requests and ARP reply packets to the target of the attack. Generally, the attacker spoofs the MAC address of the gateway. The attacker convinces the target to send a frame destined for the gateway to the MAC address the attacker wants. That way, the attacker can read the packet sent by the target and change the packet according to the attacker's wishes.

According to Ramachandran and Nandi[14], there are two methods—an active approach and a passive approach—for detecting ARP spoofing. In the passive strategy, ARP traffic is watched for irregularities while the IP-MAC mapping is investigated. In the meantime, active detection involves injecting ARP packets into the network to check for any inconsistencies in the IP-MAC mapping. This study's methodology is a passive one.

2.3. Deauthentication Attack

Deauthentication attacks are often the beginning of gateway hacks, which are more severe attacks. To collect WPA/WPA2 4-Way Handshakes, hackers often need to de-authenticate a user from a network by tricking them into connecting to a false access point or a captive portal[15].

WLAN networks are vulnerable to MITM and DoS attacks in a way that attackers can use fake de-authentication commands to force access points to re-authenticate connected devices[16]. This attack can be carried out using a tool called airplay-ng. The tool lets you make fake packets and send them to one or more devices that are connected to the access point. These tools send de-authentication packets to send de-authentication packets to the access point and target in a loop. The number of loops corresponds to the number of packets the user enters. To find and scan devices connected to a particular access point, attackers can use airodump-ng [17].

The criteria for comparison with previous studies include the devices' specifications regarding the operating system, processor speed, memory, and tools used to control the network. The Raspberry Pi Model B4 The computer used to run a programme is slower than the systems that came before it.

3. System Design

The design of a security system against ARP spoofing attacks includes the design of program algorithms and system design. The program algorithm used combines the concept of passive detection and de-authentication attack. The two concepts are combined into one in this study.

There are two phases in this system, the detection phase and the response phase. The detection phase is when the system monitors the ARP cache table to see if there are changes to the address pairs of the access points stored in the system's ARP cache table; if there are changes, the system will enter the response phase. In the response phase, the system performs a counterattack in the form of a de-authentication attack to the target indicated to have carried out an ARP spoofing attack obtained from the MAC address in the changed ARP cache table. Then the system will re-enter the detection phase after the attacker is removed from the network.

3.1. Hardware Design

The hardware design of the security system against ARP spoofing attacks is carried out by configuring the hardware to meet the system's purpose. The hardware needed to make this system is described in the following table:

Table 1. Hardware_Requirements.

NO.	Name	Amount
1	Access Point	1 Unit
2	Pc sys win10	1 Unit
3	Pc sys Kali Linux	1 Unit
4	Raspberry pi	1 Unit
5	Micro SD 16G	1 Unit

3.2. Software Design

Software design is a sequence of procedures, including research-oriented software design and coding. Typically, system algorithms are defined at the beginning of software development. The developed method will then be implemented using the Python programming language. Algorithm 1 shows the sequence of defence steps:

Algorithm 1: defence algorithm

BEGIN:

Input: ARP request packet.

Output: ARP reply packet.

Step1 read declare variables ipMac, macAP, channel, and attack list

Step2: change the wireless interface of the system into monitor mode using the aireplay-ng tool.

Step3: The script enters an infinite loop for passive detection.

Step4: If NOT passive, goto step10

Step5: else secure = true

Step6: secure = detection (macAp) and check

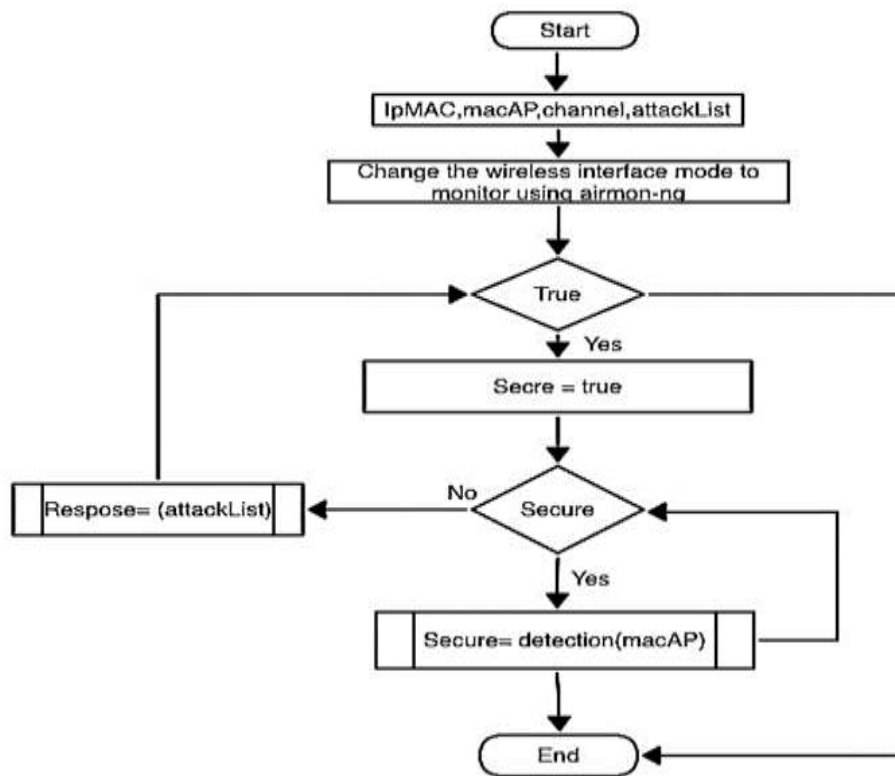
Step7: if secure = true go to step10

Step8: else return to step6

Step9: respons = attacklist() and return to step3

Step10: END: //end of the procedure

The program reads and declares variables ipMac, macAP, channel, and attack list, which include the network access point's IP address, MAC address, WLAN track, and a list of network attackers so as not to attack the same target repeatedly. The programme changes the wireless interface into monitor mode to attack the attacker. The program will then switch the wireless interface of the device into monitor mode to launch a counterattack against the attacker. Monitor mode allows the wireless interface to send de-authentication packets using the aireplay-ng tool. To change the wireless interface mode, airmon-ng tools are used, equipped with a channel where WLAN works. Then the program will enter an infinite loop so that the system can perform passive detection continuously. In this infinite loop, the program works in two phases, detection and response. In the detection phase, the program will check whether the MAC address of the access point stored in the MACAP variable matches the MAC address stored dynamically in the ARP cache table. If these two things do not match, the detection program will return a false value stored in the "secure" variable and vice versa. The program will enter the response phase if the secure variable is false. The response phase contains a command to strike back at attackers who try to poison the network using ARP spoofing. A model of the



defensive strategy is shown in Fig. 1.

Fig. 1. The defence program flowchart.

A. Detection Program

The program creates an if statement to check whether the MAC address of the original access point stored in the macAP variable matches the MAC address that is dynamically stored in the ARP cache table. If there is a difference, the program will detect an ARP spoofing attack and return false, and vice versa. Algorithm 2 and Fig. 2 show the detection steps:

Algorithm 2: detection algorithm

BEGIN:

Input: macAP.

Output: secure value (true or false).

Step1: checking for any types of attacks ARP Request packet

Step2: If (the packet is an ARP packet and ARP reply (op=2), then

Step3: get the actual MAC address of the sender and response MAC from the ARP Reply packet

Step4: If they're different, it's an assault.

Step5: If (NOT match), then

Step6: S = false goto step

Step7: else S = true goto step

Step8: return S.

END: //end of the procedure

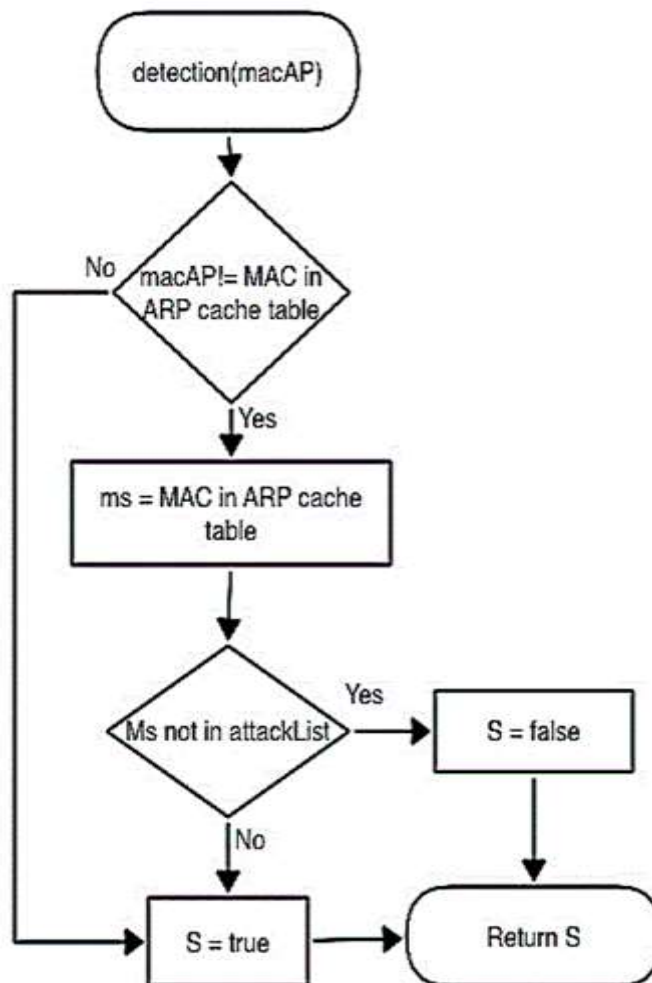


Fig. 2. Detection Program Flowchart.

B. Incident Response Program

After the system detects an attack, it will use an incident response program to follow up on the attack. This program will create a macSpoofed variable containing the gateway's MAC address in the ARP cache table that has been detected as an attack. Then the program will see if macSpoofed is in the attack list, a list of attackers who have been or are being authenticated to prevent attacks on the same target soon. Then the program will issue a device with the same MAC address as in the macSpoofed variable. In the last stage, the system will send an ARP request packet to the router to restore the original ARP cache table.

Algorithm 3 and Fig. 3 show the Incident Response steps:

Algorithm 3: Incident Response algorithm

BEGIN:

Input: secure value (false).

Output: secure value (true or false).

Step1: checking for any mac spoof

Step2: If (mac spoof in attack list), then goto to step7

Step3: else

Step4: add the mac spoof to the attack list (de-authentication).

Step5: restore the original Mac for the ARP cache table and router.

Step6: goto step7

Step7: return S.

END: //end of the procedure

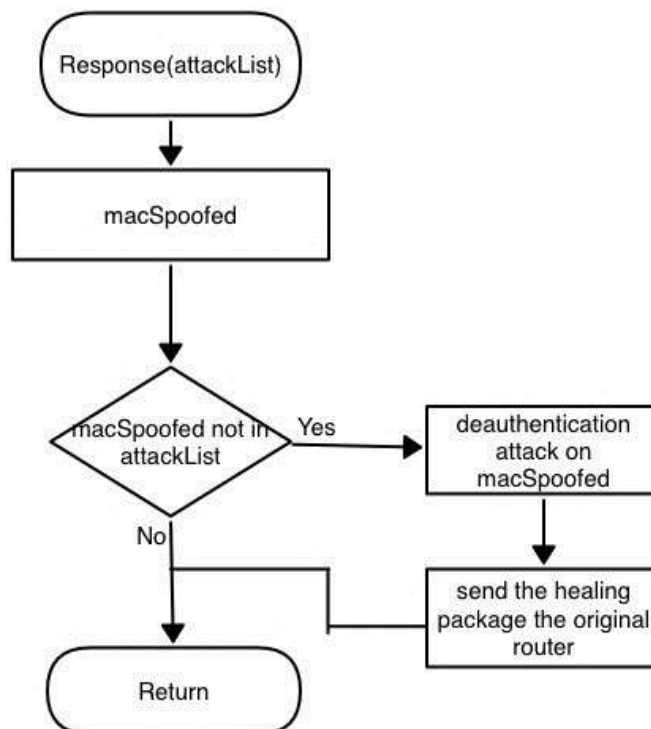


Fig. 3. Incident Response Program Flowchart.

4. System Test

The tools needed to perform system testing, among others.

Table 2. Devices in Topology.

NO.	Information	Model
1	Access Point	Tplink
2	Attacker PC	Lenovo – sys Kali Linux
3	Client PC	Lenovo – sys Win10
4	Smartphone	iPhone -sys mac16
5	RaspberryPi	Raspberry Pi 4 Model B

The testing phase is carried out to test the effectiveness of the methods and approaches used and to see how quickly this system can detect and respond to attacks on the network. The network topology used during testing is as follows.



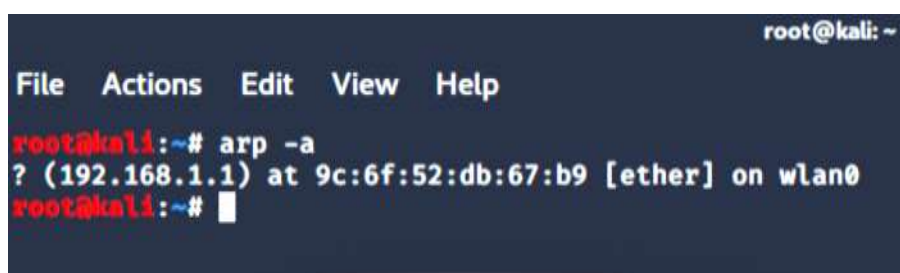
Fig. 4. Network Topology.

4.1. Effectiveness Test

The effectiveness test is carried out in two stages, before and after security, to see the security system's effectiveness. The test is carried out by attacking the network with ARP spoofing attacks using the Bettercap tool. In this test, the MAC address of a client device connected to the network is used as a parameter. If the MAC address of a client device changes, it can be determined that the network is not safe from ARP spoofing attacks.

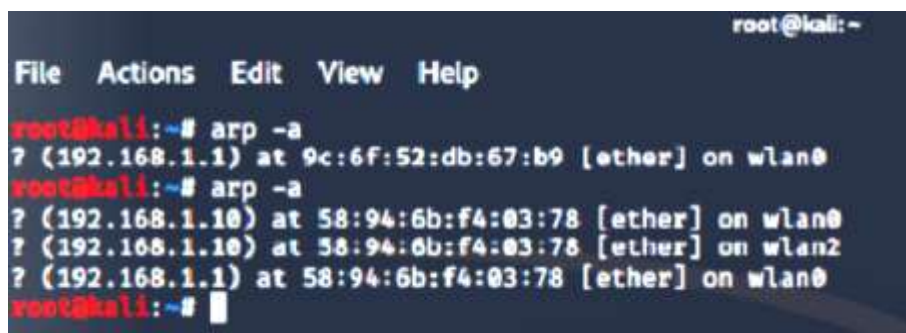
4.1.1. Effectiveness Testing Before Security

Testing before security is carried out without a network security system. This test was conducted to see if ARP spoofing attacks could be carried out on this network. Here is the ARP cache table on the client before the attack.



```
root@kali: ~  
File Actions Edit View Help  
root@kali:~# arp -a  
? (192.168.1.1) at 9c:6f:52:db:67:b9 [ether] on wlan0  
root@kali:~#
```

Picture 1. before the attack.



```
root@kali: ~  
File Actions Edit View Help  
root@kali:~# arp -a  
? (192.168.1.1) at 9c:6f:52:db:67:b9 [ether] on wlan0  
root@kali:~# arp -a  
? (192.168.1.10) at 58:94:6b:f4:03:78 [ether] on wlan0  
? (192.168.1.10) at 58:94:0b:f4:03:78 [ether] on wlan2  
? (192.168.1.1) at 58:94:6b:f4:03:78 [ether] on wlan0  
root@kali:~#
```

Picture 2. after the attack

The pictures above represent an unsecured ARP cache table client. It can be seen that there is a change in the MAC address of the router in the ARP cache table. Means that the attack on this network was successful, and the attacker was able to prevent the client and router from connecting.

4.1.2. Effectiveness Testing After Security

This second test is carried out after the security system is installed on the network you want to test. With a security system installed, the effectiveness of security in a network that is attacked using ARP spoofing attacks can be seen. The parameter for the success of network security is that there is no change in the MAC

address of the client device after the security system detects and responds to attacks using a de-authentication attack. The following is the ARP cache table after the attack.

```

root@kali:~/Skripsi# python3 mac.py 192.168.1.1
airmon-ng start wlan1 9 > /dev/null
ARP protection system initiated!

Detection system initiated

MITM attack detected!
The attacker is: 58:94:6b:f4:03:78
[]
response initiated
58:94:6b:f4:03:78
The attacker is: 58:94:6b:f4:03:78
Initiating attack sequence: aireplay-ng wlan1mon --deauth 1000 -a 9c:6f:52:db:67:b9 -c 58:94:6b:f4:03:78 > /
dev/null &
The attacker has been neutralized
Restoring ARP cache table.
Your system is secure now!

Detection system initiated

^[[B^[[B^[[B^[[A^[[A^[[A^CTraceback (most recent call last):
  File "mac.py", line 50, in <module>
    secure = detection(macAP)
  File "mac.py", line 7, in detection
    if macAP != subprocess.check_output(f'arp -a | grep -m 1 "{ipMAC}" | cut -d " " -f 4', shell=True).str
ip().decode('ascii'):
  File "/usr/lib/python3.7/subprocess.py", line 411, in check_output
    **kwargs).stdout
  File "/usr/lib/python3.7/subprocess.py", line 490, in run
    stdout, stderr = process.communicate(input, timeout=timeout)
  File "/usr/lib/python3.7/subprocess.py", line 951, in communicate
    stdout = self.stdout.read()
KeyboardInterrupt
root@kali:~/Skripsi# arp -a
? (192.168.1.10) at 58:94:6b:f4:03:78 [ether] on wlan0
? (192.168.1.10) at 58:94:6b:f4:03:78 [ether] on wlan2
? (192.168.1.1) at 9c:6f:52:db:67:b9 [ether] on wlan0
root@kali:~/Skripsi#

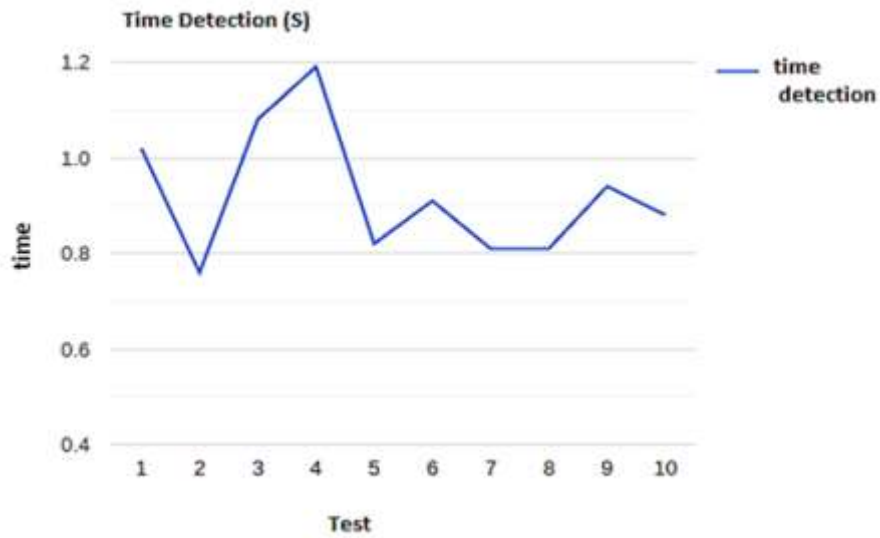
```

Picture 3. ARP Table Client After Attack Secured.

It can be seen that the MAC address of the router in the ARP cache table does not change after the security system responds to the ARP spoofing attack by the attacker. Based on this test, it was found that the security system's security is effective in thwarting ARP spoofing attacks on protected networks.

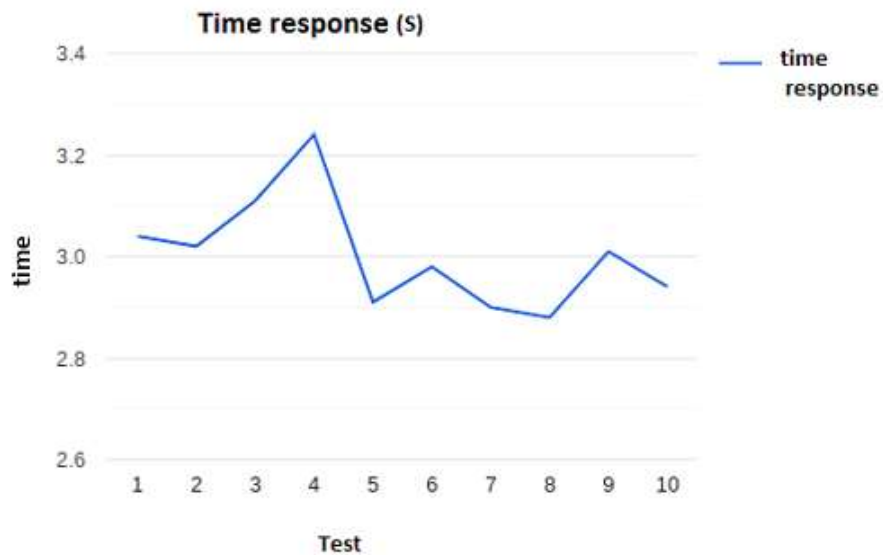
4.2. Detection and Response Speed Test

In this test, ten attempts were made to ARP Spoofing attacks on networks protected by the Raspberry Pi. The tools used in this experiment to perform ARP spoofing are arpspoof, and the time is calculated using a stopwatch from a smartphone. The results of the detection speed measurement obtained from the ten experiments are:



Picture 4. Result of Detection Measurement Experiment.

The graph above shows the speed of detection made by the system when the protected network is attacked. Based on the results of the above experiment, the average attack detection speed was 0.933 seconds. The following are the results of measuring the system's response speed after successfully detecting attacks in the network.



Picture 5. Results of Response Measurement Experiments.

According to the results of the previous experiment, the average time required to respond is 3.05 seconds. All the network security tests we conducted on our proposed technology demonstrated its relative efficacy. However, even these solutions cannot provide complete security. Even though all we do is look at the traffic on the network, we can't find attacks that take advantage of programming language mistakes.

5. Conclusion

As dependency on the network develops, network-related security issues arise more regularly, and the number of examples of damage caused by network-based security events is also expanding fast. The objective of this study is to avoid these issues and incidents of harm. The ARP poisoning attack uses an automated programme to constantly broadcast irregular ARP response packets for a continuous spoofing attack. The security proposal mainly focused on protecting against ARP spoofing attacks and leakage of critical data outside the local network. At the testing time, connecting the proposed network to the Internet was essential to test these approaches thoroughly.

After A series of experiments and tests of WLAN network security systems against ARP spoofing attacks showed that passive detection and de-authentication attacks effectively protect WLAN networks against ARP spoofing attacks. It takes an average of 0.922 seconds to detect an ARP spoofing attack and an average of 3.02 seconds to respond to an ARP spoofing and de-authentication attack. Finally, The router used was the weak security link of the network. Without using the defence mechanisms it provided, he could not detect a simple Man-in-the-middle attack. This is a router which is mainly suitable for use on home networks. In more extensive corporate networks, it should be replaced with a more powerful router with more options for setting up security.

References

- [1] Q. Zu, B. Sun, Lecture Notes in Computer Science **8944**, 598 (2015).
- [2] A.I. Abdulsada, D. G. Honi, S. Al-Darraji, Indonesian Journal of Electrical Engineering and Computer Science **23**(1), 510 (2021).
- [3] P.P. Stepanov, G. V Nikonova, T. S. Pavlychenko, A. S. Gil, Journal of Physics: Conference Series **1791**, 012061 (2021).
- [4] M.S. Hossain, A. Paul, M.H. Islam, M. Atiquzzaman, Network Protocols and Algorithms **10**(1), 83 (2018).
- [5] S. Singh, D. Singh, Indian Journal of Science and Technology **11**(22), 1 (2018).
- [6] S. Hijazi, M.S. Obaidat, IEEE Systems Journal **13**(3), 2732 (2019).
- [7] M.F. Karzon, M.S. Modabbes, International Journal and Computer.Applications **175**(32), 20 (2020).
- [8] S.M. Morsy, D. Nashat, IEEE Access **10**, 49142 (2022).
- [9] J.D. Wu, G.H. Huang, Applied Mechanics and Materials **433–435**, 1681 (2013).
- [10] B. Prabadevi, N. Jeyanthi, International Journal of Advanced Intelligence Paradigms **10**(1/2), 146 (2018).
- [11] S. Venkatramulu, C.V.G. Rao, International Journal of Scientific and Research Publications **3**(7), 1 (2013).
- [12] M.H. Alzuwaini, A.A. Yassin, Iraqi Journal for Electrical and Electronic Engineering **17**(1), 125 (2021)
- [13] M. Data, 3rd International Conference on Sustainable Information Engineering and Technology, SIET Proceedings. 206 (2018).
- [14] V. Ramachandran, S. Nandi, International conference on information systems security. Springer, Berlin, Heidelberg, 239 (2005).
- [15] J.W. Choi, Journal of the Korea Institute of Information and Communication Engineering **21**(1), 61 (2017).
- [16] H.A. Noman, S.M. Abdullah, H.I. Mohammed, IJCSI International Journal of Computer Science **12**(4), **107** (2015).
- [17] D. Srinath, S.P.S. Panimalar, A.J. Simla, J.D.J. Deepa, International Journal of Computer Applications **113**(19), 26 (2015).

الدفاع عن شبكة LAN لاسلكية ضد هجمات انتحال ARP باستخدام Raspberry Pi

هبة عماد ناصر* ، محمد حسين عبدالرضا

قسم علوم الحاسوب، كلية التربية للعلوم الصرفة، جامعة البصرة، البصرة، العراق.

الملخص

معلومات البحث

(IP) هو بروتوكول يحول عناوين بروتوكول الإنترنت (ARP) بروتوكول تحليل العنوان نظراً لوجود مشكلة أمنية تُعرف باسم (MAC) إلى عناوين التحكم في الوصول إلى الوسائط ARP ، فإن سرقة الهوية أمر ممكن باستخدام بروتوكول "Man in the Middle" هو أحد نقاط الضعف في الشبكات ARP انتحال (Address Resolution protocol). اللاسلكية عندما يتكرر المهاجم بشكل فعال في صورة شرعي. ستؤدي هجمات الانتحال إلى تقليل أداء الشبكة وكسر العديد من إجراءات الأمان. في الشبكات التي تستخدم التصفية المستندة فعلي من MAC للتحقق من العملاء ، كل ما يحتاجه المخادع هو عنوان MAC إلى عنوان عميل مرخص لكسب ميزة غير عادلة. يوصي البحث بتطوير نظام أمان يتعرف على هجمات من خلال مقارنة عنوان ARP ويمنعها. يكتشف هذا النظام محاولات انتحال ARP انتحال الخاص بالموجه في جدول ذاكرة MAC الثابت لجهاز التوجيه الأصلي بعنوان MAC بعد اكتشاف الهجوم باستخدام المعلومات التي تم جمعها من عنوان ARP التخزين المؤقت لـ ، سيجري النظام هجوماً ARP الخاص بالموجه في جدول ذاكرة التخزين المؤقت لـ MAC الخاص بالمهاجم. إذا تم قطع اتصال المهاجم بشبكة MAC لإلغاء المصادقة على عنوان يتم تشغيل هذا النظام باستخدام ARP ، فلن يتمكن من تنفيذ هجمات انتحال WLAN في 0.93 ثانية ، ARP يمكن اكتشاف معظم هجمات انتحال Raspberry Pi Model B. وتستغرق الاستجابة 3.05 ثانية.

الاستلام 07 ايلول 2022
القبول 08 تشرين الاول 2022
النشر 30 كانون الاول 2022

الكلمات المفتاحية

أمن الشبكات، هجوم الرجل في المنتصف MITM ، بروتوكول ARP هجوم انتحال ARP ، المصادقة هجوم إلغاء المصادقة.

Citation: H.I. Nasser, M.A. Hussain, J. Basrah Res. (Sci.) 48(2), 123 (2022).
[DOI:https://doi.org/10.56714/bjrs.48.2.12](https://doi.org/10.56714/bjrs.48.2.12)

*Corresponding author email : eduppg.hiba.amad@uobasrah.edu.iq

