Advances in Nonlinear Variational Inequalities
ISSN: 1092-910X
Vol 28 No. 4s (2025)

Detection of ARP Spoofing Attack by using ETTERCAP

Mohammad Daud[1], Siddhartha Sankar Biswas[2], Farheen Siddiqui[3]
[1,2,3] Department of Computer Science & Engineering, SEST, Jamia Hamdard, India

Article History:

Received: *23-10-2024*

Revised: *30-11-2024*

Accepted: *07-12-2024*

Abstract:

In our day-to-day life, we share or communicate over the internet in so many ways but to share or communicate we use some set of protocols so that we can send the information. ARP (Address Resolution Protocol) is one of them to communicate over the internet, but there are some chances of being spoofed by using the Address Resolution Protocol as attackers can steal your sensitive information through a Man-In-the-Middle attack. In this attack, a third person can be impersonated or spoofed the IP and we call it an IP spoofing attack. Therefore, to detect this attack we have used the Ettercap tool for detecting the ARP spoofing. In this detection method, we gave an approach in which Ettercap monitors the network and it is a modified Python-based script that is capable of sniffing the ARP packet transmission between the clients. Therefore, Ettercap is used for detecting ARP spoofing which is experimentally studied.

Keywords: Address resolution protocol (ARP), Man-in-the-middle attack, Cache poisoning, spoofing, Ettercap.

1. Introduction

In a standard ARP connection, the host PC transmits a packet including the source and destination IP addresses and broadcasts it to all network devices. The device with the target IP address will respond with an ARP response containing simply its MAC address and then communicate. The ARP protocol is not secure, and there is no funny mechanism in the ARP cache that causes a huge concern.

ARP requests and reply do not require any authentication or verification because all hosts on the network will trust ARP answers. We can also update the ARP cache tables by putting a fixed entry for the gateway so that the manipulator does not manipulate the gateway entry in the table, but it is not a solution because we need to change the ARP table gateway entry regularly as soon as we move. Secure socket layers are used for secure connection via HTTP or Transport Layer. The web browser will search the web server's certificates and authenticate its validity. If the certificate is verified, we will have a secure connection, but if we have a problem with the certificate, then it will be called unreliable.

The 48-bit MAC address is used to determine the interface for which an Ethernet frame is destined when it is sent from one PC on a LAN to another. The process of determining the MAC addresses of machines on a network is referred to as address resolution. This protocol enables dynamic mapping between the two types of addresses used by the data connection

layer, IP addresses and MAC addresses. This process is dynamic because it happens automatically and usually does not have to worry about the app user or the system administrator. In a shared Ethernet, where hosts use TCP / IP suites for communication, IP packets need to encapsulate in the Ethernet frame before transmitting it on the wire.

Man-in-the Middle Attack

A person in the middle (MITM) attack occurs when a criminal insert himself into an interaction between a user and an app - either by prejudice or impersonation of one party. An attacker's role is to steal personal information such as login credentials, account information, and credit card numbers. Goals are typically users of financial applications, e-commerce sites, and other services that require registration. Information obtained during an attack can be exploited for a variety of purposes, including identity theft, unauthorized fund transfers, and incorrect password changes. Consider two people, that is, Alice and Bob, communicating with each other. The third party is the Man in the middle attacker. But if the authentication protocol is not reliable, then Eve can apply for both Alice and Bob as other communication parties.

Furthermore, messages between Alice and Bob will go through Eve because Eve Bob will impersonate Alice and Alice as Bob. Both Alice and Bob will not know about intercepting their messages. Therefore, Eve takes advantage of Alice and Bob without her consent and receives important information.

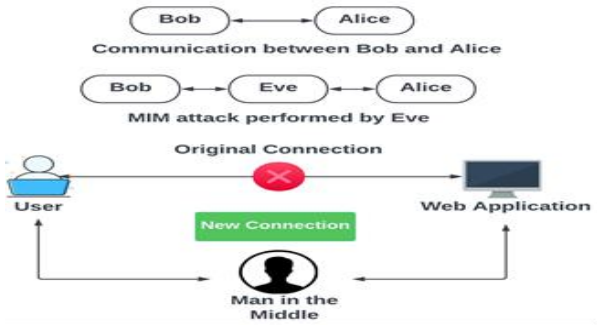

Fig. 1. Overview of Man in the Middle Attack.

Address Resolution Protocol (ARP)

In a standard ARP connection, the host PC transmits a packet including the source and destination IP addresses and broadcasts it to all network devices. The device with the target IP address will respond with an ARP response containing simply its MAC address and then communicate. The ARP protocol is not secure, and there is no funny mechanism in the ARP cache that causes a huge concern.

ARP reply packet can be easily screwed, and it can be sent to machine without sending an ARP request to know that this is not a real machine, but there is an attack to cause data breaches. This is because the attacker will update ARP cache tables and therefore all network traffic will go by the attacker machine, and he will have all the data and will get the maximum benefit. This is the best attack on the LAN networks.

Many types of tools are accessible in the market for performing ARP cache poisoning attack. Some are Ettercap, Dsniff, and Cain and Abel's, etc. We can try to defend this ARP cache poisoning by using dynamic ARP inspection (DAI). DAI is a security feature used to validate the ARP packet in a network and leave the invalid IP for MAC address binding.

Fig. 2. Regular traffic flow between two Computers.

ARP requests and reply do not require any authentication or verification because all hosts on the network will trust ARP answers. We can also update the ARP cache tables by putting a fixed entry for the gateway so that the manipulator does not manipulate the gateway entry in the table, but it is not a solution because we need to change the ARP table gateway entry regularly as soon as we move. Secure socket layers are used for secure connection via HTTP or Transport Layer. The web browser will search the web server's certificates and authenticate its validity. If the certificate is verified, we will have a secure connection, but if we have a problem with the certificate, then it will be called unreliable.

2. LITERATURE REVIEW

Today's technology cannot provide complete protection from ARP attacks, but we can keep ourselves safe with IDS and special ARP manipulation sensors, explained in upcoming chapters, to find out the most manipulation efforts.

The authors of [1] attempt to create methods for examining the hazards and risks in the Address Resolution Protocol (ARP) by associating the IP address with the MAC address.

A specialist-based ARP store harming discovery (AACPD) discussed in [2] helps in recognizing the MITM and DoS attack in a switch LAN scenario. The target of this plan is to give insurance from insider pernicious clients.

[3] proposes a protected SDN-based IoT engineering to oversee and lessen ARP mocking assaults by conveying another machine close to the SDN regulator to deal with address goal questions.

A technique for detecting man-in-the-middle attack using Deep Packet Inspection and Deep Flow Inspection based on DPI Feature Library and DPI Method Library as well as DFI Feature Library and DFI Method Library for network traffic distinguishing proof and bundle sifting of approaching organization traffic has been discussed in [4].

The authors of [5] proposed an Intrusion Detection and Prevention System (IDPS) based on Software-Defined Networking (SDN) that protects against ARP mocking and Blacklisted MAC Addresses.

[6] The weaknesses on the specialist co-ops will expand with the expansion of the interest for CC. Efficient refusal of administration (EDoS) assaults assume control over the provider monetarily influencing the divergent associations that use the cloud information.

[7] To Develop an essential degree of safety to a LAN-based framework to forestall assaults that might actually quit serving solicitations to different clients. for example, online assaults that could deny the IP to get to the asset present in the web server of the host.

[8] The most fundamental element of each and every shrewd gadget or endpoint gadget in IoT is to gather the huge measure of information that is being created and direct it to an objective server through the Internet. IoT network is defenseless against assaults and ID of these malevolent ways of behaving at a beginning phase can save the information from assaults.

[9] Networks not utilizing objective side source address approval (DSAV) open themselves to a class of malicious assaults which could be forestalled by separating inbound traffic indicating to start from inside the organization. In this work, we study the inescapability of organizations helpless against penetration utilizing caricature addresses inward to the organization.

[10] IoT frameworks are helpless against different digital assaults as they structure a subset of the Internet. Insider assaults find more importance since numerous gadgets are designed to get to the Internet without interruption location frameworks or firewalls set up. Current work centers around three insider assaults, specifically, blackhole assault, sinkhole assault and wormhole assault.

3. PROPOSED DETECTION TECHNIQUE

The attack can be performed the task of implementing the attack, preventing and detection on a system with the following methods.

For creating a virtual environment for picturing local area network, here used the VMware approach to virtualization. It offers a secure virtualization platform that can scale over a large number of interconnected computers and storage devices to form a complete virtual infrastructure. ARP spoofing attacks can be performed using the inbuilt tool in Kali Linux and also with the commands.

Here provided some easy to implement techniques to prevent ARP attacks of every kind. We mentioned the basic idea of what is required by a system to protect itself from a spoofing attack in the local area network. So, our mechanism needs to fulfil the following:

• It should be sufficient in itself to adequately protect against all kind of ARP attacks.

• It should have characteristics for tracking any malicious acts.

• There should be a management function for the administrator for keeping the security of the network intact.

ARP is very confident and can continue to work smoothly. That is, getting MAC with its IP from a specific computer in the network. The MAC address is requested, and the response is received. Now the issue arises whether the solution obtained is with a valid pairing of < IP,

MAC>. Is it okay to update the fresh entry cache or not. It relies on all safe and reliable work. Therefore, every effort is made to guarantee that the response received is authentic.

Prevention technique

Using ARP packet request to the gateway:

It uses the user's shell script, which is sufficiently conscious to avoid them from changing the cache according to their wishes from the multiple assaults. These scripts are kept in the background by users. This also offers appropriate protection against attacks. The shell script is as:

While [1]

Do

Arping –f 192.168.x.x

Sleep 5

Done

Above shell script periodically keeps sending a ping request to the gateway to keep the MAC address of the gateway system always in the ARP cache. Arping is a kind of broadcast request that is sent to every network user. But we limited it to the gateway only here in the shell script, and gateway would depend on its own MAC address. Since this is running periodically, it would not be possible for hackers to poison the cache. If ARP becomes poisoned anyhow, the running script will restore the initial pairing < IP, MAC>.

• Monitoring arp –a table database:

A Linux might use this shell script to monitor the system's arp entry or arp table. Any system always checks its arp cache before sending a message for sending the data into the MAC layer. If in the arp cache there is an inaccurate MAC-IP connection, then there is the chance to deliver information to a wrong customer. And the reels of the scheme are under assault by arp poisoning. The script below uses the awk command that regulates the user's arp cache entries. It regularly checks the content of the arp table at periodic intervals of time. In the script, the gateway's MAC address is being screened for arp spoofing option and the user will be alerted if it happens. Using various strong Linux commands, the user can take the necessary action to get the alert. The required shell scripts is as:

!/bin/sh

while [1]

do

a = arp –a | awk {print S4} | grep xx:xx:xx:xx:xx:xx

If [$ a -gt 1]

then

notify-send -t 0 System is poisoned

31

fi

sleep 8.5

done

This alternative stops Arp from sending emails to the portal instead of checking the Arp cache of the system at a regular interval. It generally checks how many IP addresses get associated with the MAC of the gateway. If there are more than two such associations, it alerts the user that ARP is being played. But the customer also requires this alternative to understand the MAC gateway in advance. If any changes are made in the portal by the network administrator, the customer will have to modify the request. Once any poisoning has been completed, the user may use the command' arp-d' to delete the poisoned arp entry.

• IDS Snort :

Snort can detect the ARP spoofing assaults by altering the Snort configuration file which is snort.conf, specifically as follows:

Open the snort. Conf, find the two lines below:

preprocessor arpspoof :

preprocessor arpspoof-detect-host : host-ip host-mac

preprocessor arpspoof-detect-host : gateway-ip gateway-mac

snort, we get to understand the spoofing attack being carried out, and we can discover the malicious hosts using our detection method.

• Storing the IP-MAC during the Broadcast:

It has been noted from the previous debate on ARP that the primary issue with ARP is that it is a protocol of trust. We have not only created the application to be broadcast here in this alternative, but also the answer to be broadcast. Here, by storing the data from the request, we are attempting to create our protocol function as a stateful protocol. In this, the devices in the network except the host will store pairing corresponding to the host in the ARP cache in both ARP request and ARP response when the request and response is broadcast. The procedure is as in the following steps:

1. Presume a Machine A wants to communicate to D, but A is only knowing the IP address of D.

2. So, A broadcasts an ARP Request with an IP address of D.

3. All LAN devices obtain the ARP request and update their ARP cache with A's MAC address.

4. Now, by transmitting ARP Reply, D acknowledges its IP and responds with its MAC address.

5. All schemes attach their various ARP caches with the MAC of D.

6. Now A can speak to D and submit packets of information.

Working ARP has proven to be much safer and effective with this shift in sending ARP response in a broadcast fashion. Accordingly, whenever any ARP response is obtained by any host, it checks whether the input corresponds to the target in its local ARP cache.

The entry of the host is present or not. If the entry is already present, the answer will be accepted by the source host. Otherwise, it's just going to be discarded. It will also check for the ultimate combination of destination IP and MAC addresses in the ARP Reply Frame before broadcasting the Reply Frame. Twice will also take place in this operating update of ARP caches. The first is when ARP application is transferred (the source host will be registered during this IP and MAC) and the second time when ARP response is assigned (the destination host will be collected during this IP and MAC).

Detection technique

The detection mechanism is based on a script which is written in python. It is capable of sniffing the ongoing ARP packet transmission between the respected AP and its clients. It is designed to capture only the ARP broadcast traffic belonging to that network and discard all other packets. It detects the attack based on the analysis of the captured ARP request frames.

In the detection phase, the Detection technique would follow the steps:

Step 1: Capture all incoming traffic.

Step 2: Check ARP traffic from the traffic.

Step 3: Extract sender IP, sender MAC, destination IP (victim IP), and packet number from each ARP request message. Then further filter the ARP request messages which contain gateway IP address in the field of Sender IP.

Step 4: Check request threshold, if exceeds check IP-MAC pair.

Step 5: if the IP-MAC is not genuine, show detected MAC address

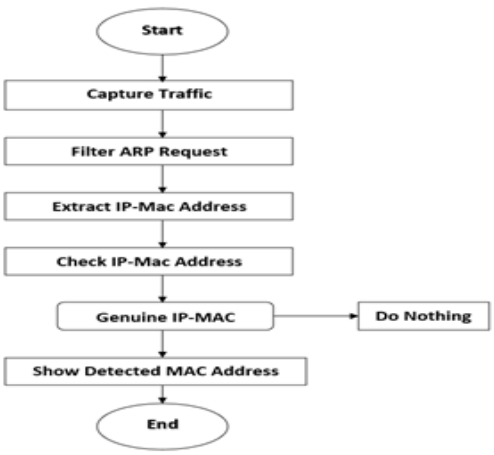

Fig. 3. Detection stage

Implementation of attack

Step1: Ettercap scans the host at selected interface. Victim IP added to Target 1 and gateway IP is added as Target 2. After running unified sniffed the attacker poisoned the ARP cache of victim. Fig 4. shows the interface of Ettercap with selected target.

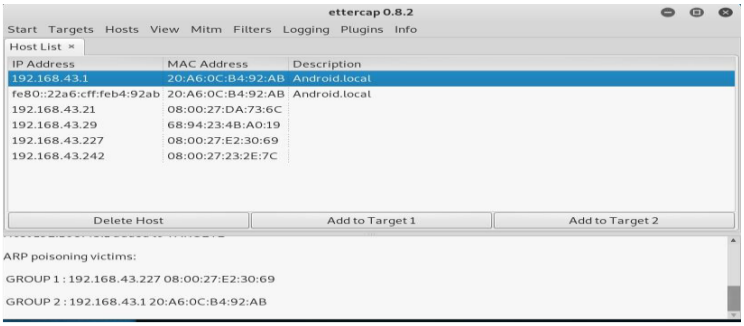

Fig. 4. Shows the selected targets in Ettercap

Step2: Checking ARP cache of victim machine to check the MAC address entry.

Fig 4.1 highlights the IP-MAC entry in victim's machine. The MAC address of gateway is replaced by attacker MAC address. Command used: arp –a

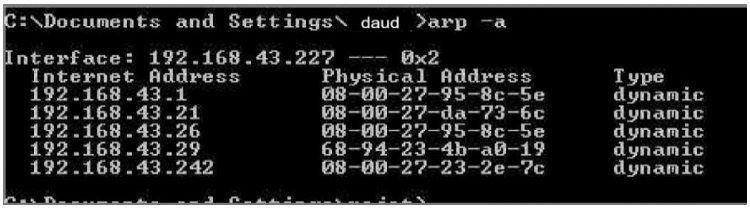

Fig. 4.1 Victim MAC Arp cache after the attack

Step3: The attacker executes sslstrip command to get the username and password entered by the victim in his machine. Command used is: Sslstrip –l 8080

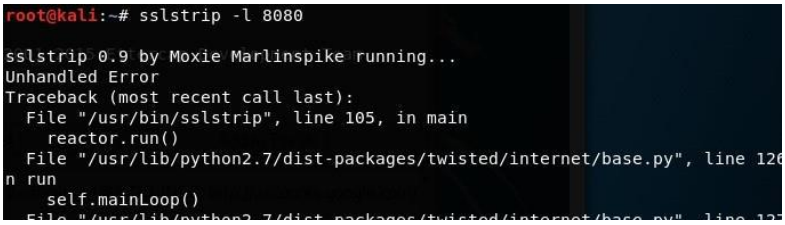

Fig. 4.2 Attacker executing the command

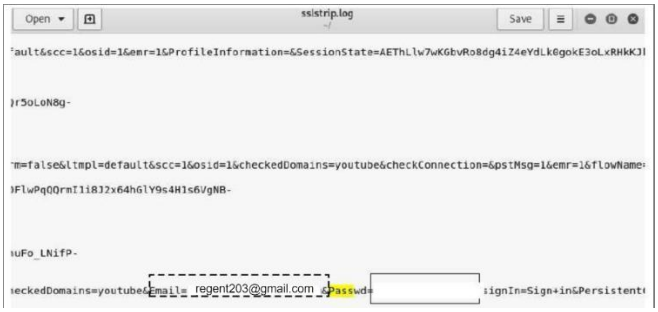

Fig. 4.3 Logfile of sslstrip at attacker side

Step5: Run the Python script, as shown in Fig. 4.4

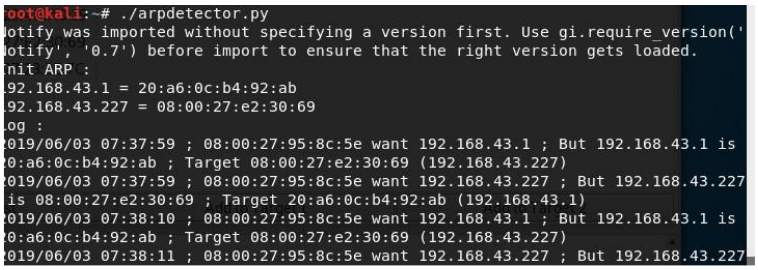

Fig. 4.4 Output of the script

Step6: Get Notification alert of attack

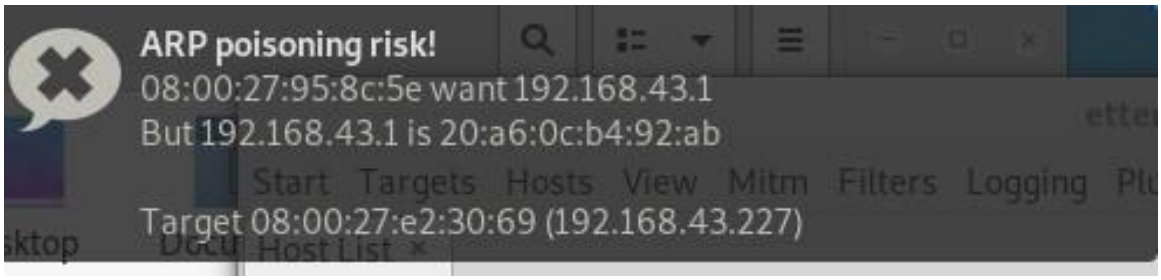

Fig. 4.5 Alert of detecting attack

4. Results and Discussion

The ARP spoofing attack is implemented, and the results of the detection techniques are being discussed. The ARP spoofing attack is very easy to launch. The tools required for launching this attack are freely available on the Internet. The sophistication of the attack lies in the fact that the victims will have no idea that they have given away any confidential information to the attacker if the attack goes undetected.

Parameters used

Three parameters are used to evaluate the results that are explained as follows:

1) Detection time is measured by calculating the difference of time at which the attacker starts the attack and the time at which the script '.py' Found the first ARP reply packet.

2) Mitigation time is measured by calculating the difference between the time at which the attacker starts the attack and the time at which the victim received first reply ARP packet from the detection node.

Let T1 = time at which attacker send attack packet,

T2 = time at which script detects the attack packet,

T3 = time at which victim received re-ARP packet,

Then, Detection time, Td =T2- T1

Mitigation time, Tm = T3- T1

Attack duration = Mitigation(Tm) = T3- T1

(3) The detection rate is referred to as the total count of intrusion instances that the system identified as a (True Positive) divided by the total number of intrusion occurrences present in the test set.

Detection rate = (Number of intrusion attempts /Total no. of intrusion attempts) * 100

Calculation Detection time

To measure the detection time of attack, i.e., the difference between at which attacker started to send the poisoned ARP request.

Result: Time at which script found first packet containing mismatch binding for gateway IP-MAC.

Analysis: Time is taken to detect the attack increases as the no. of packets increases. As shown in the figure the minimum packets require the proposed to detect the attack is around 2 secs when no of packets is high. This is because when no. of hosts increases, it makes the traffic overload, which increases the detection time.

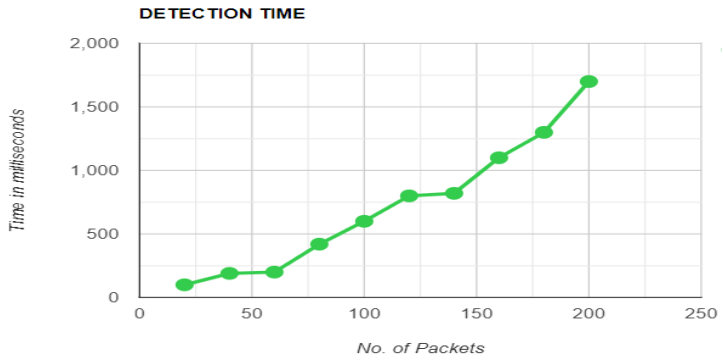

Fig. 5. Detection time w.r.t. No. of packets

There are two test cases at different intervals of time. First is use one attack packet at each intrusion attempt and second is use five attack packets in each attempt. Detection rate ata different time and in both cases is plotted in the table shown below. The detection rate when the attack was performed using single packet drops significantly to 90.5%, but when the attack is passed out by sending multiple consecutive packets, the detection rate is about 100%.

In this, we tested our proposed scheme with three different cases of LAN attacks. In the the first one is of regular operation in the absence of the proposed mechanism. The second case is when the proposed method is active, and there are no attacks created in the network. 3rd case is when attacker injected <100 spoofed ARP into the LAN and our mechanism is running. We observe almost same amount of ARP traffic under the normal situation with and without proposed method running. In case of attack, a little more traffic is generated by our detection method for the probes. With each spoofed ARP packet, our detection scheme sends a request for a probe and expects at least two responses (one from the normal and one from the attacker) by adding only three ARP packets for each spoofed packet.

It has been shown that the method is quite safe and effective.

Providing the ARP response method also helps to prove safety against ARP cache poisoning. As if any intruder attempts to send spoofed ARP response packets to any host in the network, this response would also be obtained by the selected host whose IP address is used to mapping the attacker device with MAC.

So, this host gets to comprehend the spoofed response to this ARP. And the attacker is the system that matches the Mac. Therefore, we can conclude that the notion of transmitting the ARP response is also safer and more efficient. No cryptography is used, so no performance degradation.
Details are in the table below.

Time (s)	**10**		**30**		**50**		**70**		**90**	
Attempts	**No. of packets**		**No. of packets**		**No. of packets**		**No. of packets**		**No. of packets**	
	1	5	1	5	1	5	1	5	1	5
1	All	All	All	All	All	All	All	All	All	All
2	All	All	All	All	All	All	All	All	All	4
3	All	All	All	All	All	All	All	All	All	4
4	All	All	All	All	All	All	All	All	All	All
5	All	All	Miss	All	All	All	All	All	All	All
6	All	All	All	All	All	All	Miss	All	Miss	4
7	All	All	All	All	All	All	All	All	All	All
8	All	All	All	All	All	All	All	All	All	All
9	All	All	All	All	All	All	All	All	All	All
10	All	All	All	All	All	All	All	All	All	All
11	All	All	All	All	All	All	All	All	All	3
12	All	All	All	All	All	All	All	All	All	All
13	All	All	All	All	All	All	All	All	Miss	4
14	All	All	All	All	Miss	All	All	3	All	All
15	All	All	All	All	All	All	All	All	All	4
Detection rate (% age)	100	100	93.33	100	93.33	100	86.67	100	80	100

5. CONCLUSION AND FUTURE SCOPE

Due to the Unix operating system patches and the widespread use of random sequence numbering, ARP spoofing is therefore less of a threat today. Many security experts are predicting a shift from ARP spoofing attacks to application-related spoofing, in which hackers can exploit a weakness in a specific service to send and receive information using false identities. As the hacker community continues to scan our systems and networks for weaknesses and vulnerabilities, a research stream of updates and new challenges is guaranteed.

Many protocols are made to make setting up networks easier. Although some of them do not take security concerns. In this paper efforts have been made to expose some of the vulnerabilities that exists in a standard and broadcast Network Protocol, Address Resolution Protocol (ARP) Protocol. Effectually, it is possible to implement a user-friendly and an easy tool that exploits the weaknesses of this protocol to mislead a victim machine and a router through a kind of man-in-middle (MITM) attack.

This allows the attacker to inspect the victim's data packets, extract valuable data (such as passwords), and manipulate or change these data packets. To separate it, a defense mechanism and equipment that counts the attack, warns the user, and discloses some information about the attacker can be used. Linux is an operating system to implement both attack and defense equipment. It can be successfully used for defense mechanisms in detecting ARP related attacks effortlessly and efficiently. In this region, however, there is a lot to be investigated. And to create an integrated technology solution that could minimize the danger posed by ARP spoofing.

Refrences

[1] Hijazi, S., & Obaidat, M. S. (2018), A New Detection and Prevention System for ARP Attacks Using Static Entry. IEEE Systems Journal.
[2] Daniyal Sakhawat, Abdul Nasir Khan, (2019) on Agent-based ARP cache poisoning detection in switched LAN environments, IET Networks Journal.
[3] Hamza Aldabbas & Rashid Amin, (2021) A novel mechanism to handle address spoofing attacks in SDN based IoT, Springer
[4] Argha Ghosh, A. Senthilrajan, (2020) "An Approach for Detecting Man-In-The-Middle Attack Using DPI and DFI, Springer
[5] Thomas Girdler, Vassilios G. Vassilakis (2021). Implementing an intrusion detection and prevention system using Software-Defined Networking: Defending against ARP spoofing attacks and Blacklisted MAC Addresses.
[6] S. B. Ribin Jones, N. Kumar (2021). An efficient EDoS-DOME system in cloud computing using obfuscated IP spoofing technique and RCDH-ENN detection technique.
[7] R Abhijith; B.J Santhosh Kumar (2021). First Level Security System for Intrusion Detection and Prevention in LAN
[8] K.V.V.N.L.Sai Kirana R.N. Kamakshi Devisettya N. Pavan Kalyana K.Mukundinia R.Karthi (2020). Building a Intrusion Detection System for IoT Environment using Machine Learning Techniques
[9] Alden Hilton, Joel Hirschmann, Casey Deccio, (2022), Beware of IPs in Sheep's Clothing: Measurement and Disclosure of IP Spoofing Vulnerabilities. IEEE.
[10] K. N. Ambili & Jimmy Jose, (2019), Trust Based Intrusion Detection System to Detect Insider Attacks in IoT Systems.