Advanced approaches to prevent ARP attacks

Qijun Li[1,3,†], Yipeng Dong[2,4,†]

[1]Department of Computer Science and Technology, Xidian University 710126, China
[2]School of Cyberspace Security, Xi'an University of Posts and Telecommunications 710121, China

[3]1176839330@qq.com
[4]594810824@qq.com
[†]These authors contributed equally to this work and should be considered co-first authors.

Abstract. Nowadays, there exist various types of Address Resolution Protocol(ARP)-based attacks, such as ARP flood attacks, ARP spoofing host attacks, attacks that spoof gateways, man-in-the-middle attacks and Internet Protocol(IP) address collision attacks. Focusing on the prevention to ARP spoofing, this paper first introduces S-ARP, a secure version of ARP utilizing asymmetric cryptography and focusing on message authentication rather than traffic confidentiality that aims to mitigate such attacks. It then talks about a modular approach based on multiple modules utilizing databases instead of focusing on ARP table cache to detect and mitigate ARP cache poisoning. At last the paper talks about an approach with Software Defined Network(SDN) to prevent cloud computing from being vulnerable to ARP poisoning. We then make some comparisons of these methods from three aspects in the comparison section and give the advantages of each method. In the end, these scenarios are summarized in the concluding section of the paper.

Keywords: ARP attacks, S-ARP, Module, SDN, Cloud computing.

1. Introduction

The internet has evolved into an essential tool for our daily routines, with approximately 4.2 billion individuals, accounting for around 54.2% of the global human population, actively utilizing it today [1]. The majority of internet users rely on the internet for communication and information sharing, making it crucial to secure the data that passes through it. One of the ongoing challenges is ensuring the protection of user data. The internet operates on a packet-switching communication model, where data is broken down into smaller packets and sent through various channels in a sequential manner before rejoining at the destination node. However, this architecture faces potential threats, including attacks on the data link layer like ARP spoofing or ARP poisoning [2]. This particular attack takes advantage of the vulnerability within the ARP protocol, which is responsible for translating logical addresses into corresponding physical addresses of devices.

Over the course of several decades, numerous research studies have put forward various methods with the goal of reducing the impact of ARP spoofing attacks.

This article aims to show three kinds of solution to prevent ARP spoofing. Our purposes are to compare different solutions in different situations. The three solutions consist of a fundamental solution that is S-ARP, a modular solution, and a solution in an SDN-based cloud environment.

In order to present our findings and arguments effectively, we have structured this paper in the following manner. In section II, we provide an overview of ARP spoofing technique and attack. In section III, we describe S-ARP. We explain the modular solution using databases and in an SDN-based cloud environment in section IV and section V. In the final section, we provide a comprehensive summary and conclusion of the entire paper.

2. Background

The ARP is a Transmission Control Protocol(TCP)/IP protocol used for mapping IP addresses to physical addresses. When a host needs to find the physical address of a destination IP address, it broadcasts an ARP request to all hosts on the network. The host then receives a response that contains the physical address corresponding to the requested IP address. The IP address and physical address are stored in the local ARP cache for a certain period of time. This allows for quick retrieval of the physical address when needed, thus conserving resources [3]. ARP attacks are primarily committed to spoofing IP addresses and MAC addresses so that the source, target, and ARP packet protocol addresses of Ethernet packets do not match. As a result, large amounts of ARP communication are generated in the network, resulting in network interruptions or intermediary attacks. ARP attacks mainly exist in the local area network, and if one of the computers is infected with an ARP virus, it will try to intercept the information of other computers in the local area network through ARP spoofing, causing computer communication failures within the local area networks [4]. ARP spoofing is the use of impersonating gateways or other hosts to cause traffic arriving at gateways or hosts to be forwarded through an attack. This can control traffic or obtain confidential information. ARP spoofing does not really make the network unable to communicate normally, but ARP spoofing sends false information to other hosts in the LAN, which contains the IP address of the gateway and the MAC address of the host; and also sents ARP reply to the gateway, when the host and gateway in the LAN receive the ARP reply and the new ARP table, the traffic between the host and the gateway needs to be forwarded through the attack host [5].

3. Protocol-based security upgrade solutions

This section introduces a robust variant of the ARP designed to safeguard against ARP poisoning attacks. Each host on the LAN has a certified public/private key pair issued by a trusted local entity acting as a Certification Authority. This ensures that messages sent by a host are digitally signed, preventing the injection of fake or spoofed information. The proposed solution was successfully validated. Performance evaluations demonstrate the practicality of employing a Public Key Infrastructure(PKI) for authentication, even in the case of lower-level protocols. As long as the verification overhead for key validity remains minimal, the utilization of PKI proves feasible [6].

3.1. Protocol Overview

S-ARP is a technique that utilizes asymmetric cryptography to prevent ARP spoofing. It focuses on message authentication rather than traffic confidentiality. Every host connected to the LAN possesses a certified public/private key pair that has been issued by a trusted local entity serving as a Certification Authority (CA). A fundamental certificate is utilized to associate the identity of the host with the corresponding public key it possesses. The host transmits its authenticated certificate to the Automatic Key Distribution(AKD) system. The AKD system receives the host's signed certificate. The AKD system then adds the public key and IP address to its local database. No revocation list is stored to avoid key compromise.

To counteract replay attacks and establish a synchronized time reference for evaluating outdated replies, AKD employs the distribution of a clock value to all hosts. This ensures that all hosts are in sync and can accurately determine the validity of replies.

In the S-ARP protocol, the sender digitally signs all reply messages using their private key. After receiving the message, the receiver host uses the corresponding public key to perform signature verification. If the recipient's keyring does not exist in the public key or cannot be verified signature, the recipient requests the sender's public key from the AKD. To ensure the authenticity of the public key, the AKD responds to the requesting host by sending it in a digitally signed message.

3.2. Protocol design

3.2.1. S-ARP Setup: To set up a LAN with S-ARP, first, to securely distribute the public key and MAC address of the AKD to all other hosts, a detection process is initiated to identify the AKD. Once the AKD is identified, its public key and MAC address are securely shared with all other hosts. This is done by manually encrypting communications during initial host installation. To connect to a LAN, a host must generate a public/private key pair and then share its signing certificate with AKD. The network administrator and the host's public key verify the accuracy of the information. Once verified, the host's public key and IP address are stored in the AKD repository. This process is only required when the host initially joins the LAN. By signing a request with the previous key, the host transmits the newly generated key to the AKD in order to renew it. The AKD updates its records accordingly for accurate association. After joining the LAN, a host aligns its local S-ARP clock with the clock of the AKD through synchronization.

3.2.2. Message Format: S-ARP messages have similarities to ARP messages, but include an additional section at the end to ensure compatibility with the original protocol. ARP replies retain the S-ARP header, while ARP requests remain unchanged. The S-ARP header encompasses the digital signature of the sender, a timestamp, and details about the message type and length.

3.2.3. Identity Verification: Each host maintains a circular list of previously requested public keys and their corresponding IP addresses obtained from the AKD. Upon receiving an S-ARP reply, a host searches its list for the sender's IP address and retrieves the associated public key. If a match is found, the host uses this information to verify the signature. If no entry is found, the host sends a certificate request to the AKD. In this case, the packet is placed on the "pending reply list" for further processing.

AKD responds to the request with a digitally signed reply that includes the requested public key and the current timestamp. Upon receipt of this reply, the host does the following: checks to see if the local clock needs to be resynchronized, adds the public key to its list and proceeds to verify the signature by storing the public key in its list and performing signature verification. If the old key is no longer valid and the new key matches the key stored in the cache, the host considers the reply invalid and ignores it. Yet, should there be a change in the key, the host will update its cache and proceed to verify the signature using the updated key.

To prevent possible replay attacks, the host verifies the timestamp in the S-ARP reply. If the timestamp is considered too old, suggesting a potential replay, the host rejects the reply. A time difference of approximately 30 seconds between the timestamp and the local clock is acceptable due to imperfect synchronization among hosts. The exact allowable time difference is established by the network administrator and configured to safeguard against replay attacks. Setting this value too high could leave the system vulnerable and susceptible to exploitation by attackers.

3.2.4. Key Management: The management of keys in networks is typically considered independent of whether IP addresses are assigned statically or dynamically.

For networks that have IP addresses statically assigned, keys are generated and associated with specific IP addresses. These keys are stored in the AKD repository. In the event that an attacker tries to create a legitimate signature for an IP address that differs from their own, they would be unsuccessful. This method effectively hinders the attacker's ability to transmit deceitful ARP replies and redirect traffic through their network adapter. However, it is still possible for the attacker to announce an incorrect MAC address for their network adapter. This MAC address could belong to another host or even be non-existent.

In an S-ARP network with dynamically assigned IP addresses via a Dynamic Host Configuration Protocol(DHCP) server, the binding of keys to IP addresses is not done during their generation. Instead, the association is dynamic and gets updated whenever the host receives a new IP address assignment. To ensure security, the assignment of dynamic IP addresses is limited to authorized computers that have gone through an enrollment process involving authentication or authorization.

During the enrollment procedure, the host generates a public-private key pair and obtains a certificate. Initially, the certificate's IP field is blank. To finalize the enrollment process, AKD manually includes the certificate with a null IP address and its corresponding public key certificate. Use a secure channel to map the public key to its secure key repository. Please note that this process is a one-time step that is performed prior to the host entering the system.

If a host decides to modify its key in the future, it can easily send a key exchange packet to AKD. Whenever S-DHCP allocates a new IP address to a host, whether it be due to lease expiration or a new request, it notifies AKD about the updated association. Similarly, when the S-DHCP renews the lease with the host, it informs the AKD about the renewal.

3.3. Experimental evaluation

The prototype was created as a demonstration of the concept. It consists of two components: a kernel patch and a user-space daemon, as depicted in Figure 1 [6]. The kernel patch utilizes the *dev remove pack()* function to eliminate the ARP packet retrieved from the incoming packet list.

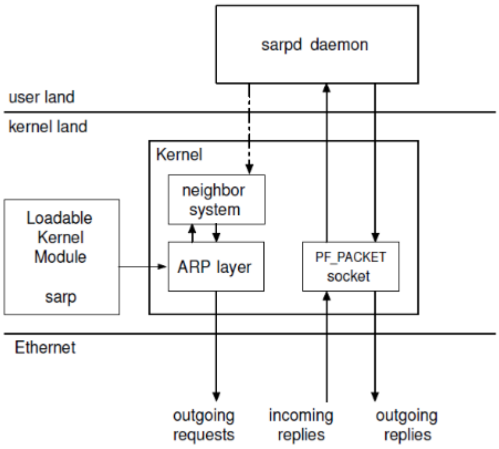


Figure 1. The Structure of S-ARP

Two sets of measurements were performed: one is used to measure the signature operation in isolation, while the other is employed to indirectly assess the impact of S-ARP on address resolution.

1). Signature Performance: Using the given details, Table 1 [6] summarizes the results of 1000 tests conducted to determine the time taken for generating a signature that incorporates pre-computation of exponential factors and another 1000 tests to measure the time for verifying a signature using both 512-bit and 1024-bit keys.

Table 1. Execution times in microseconds (μsec) for various key lengths (in bits) for signature operations (exponential factor computation, signature generation, signature verification) were obtained by averaging 1000 tests.

Key len.	operation	min	max	mean	st. dev.
3*512	exp. fact.	923	1082	982.47	16.91
	sig. gen.	32	58	33.45	1.53
	sig. verif.	1133	1255	1201.46	15.45
3*1024	exp. fact.	2565	2819	2721.67	38.05
	sig. gen.	34	59	35.36	1.50
	sig. verif.	3204	3458	3346.24	38.07

2). ICMP Performance: To evaluate the performance of S-ARP, Internet Control Message Protocol(ICMP) messages were utilized as an indirect measurement. Ping commands were executed with and without the implementation of S- ARP, without any additional parameters. Ping measures the roundtrip delay of ICMP messages, including the time for address resolution in the first message. This allows us to estimate the impact of S-ARP on ICMP execution time.

Two sets of experiments were performed. In the first case, the two hosts had never communicated before, so they did not possess each other's public keys. They requested the public keys from the AKD. This scenario only occurred during the initial request for a new MAC address.

The second scenario represented the typical operation of S- ARP, where the keys were already cached in all subsequent requests, resulting in faster execution.

Table 2 [6] displays the recorded roundtrip delays of ICMP echo requests for both 512-bit and 1024-bit keys in 20 repetitions, as obtained from the ping command. While the time taken in both cases is noteworthy, it is crucial to consider that this delay only transpires during the initial occurrence and does not substantially affect performance in the average scenario.

Table 2. The roundtrip delay in microseconds (μsec) for ICMP echo request messages with cold key caches was measured for various key lengths (in bits).

Key len.	min	max	Mean	st. dev.
512	17.7	18.1	17.86	0.12
1024	48.0	48.8	48.49	0.22
classic ARP	0.6	0.8	0.70	0.05

Table 3 [6] presents the roundtrip delay measurements of ICMP echo requests for 512-bit and 1024-bit keys in 20 repetitions, obtained from the ping command. The data presented in the table reveals that the time recorded in this particular scenario is nearly half of the time recorded when caches are not optimized, suggesting a reasonable level of overhead. In this scenario, two verification operations on the AKD messages are eliminated, resulting in reduced time consumption.

Table 3. The roundtrip delay in microseconds (μsec) for ICMP echo request messages with cached keys was collected for various key lengths.

key len.	min	max	mean	std. dev.
512 bit	8.8	9.3	8.96	0.13
1024 bit	23.6	24.4	24	0.2
classic ARP	0.4	0.5	0.46	0.05

3.4. Conclusion

This part introduces a practical solution to ARP poisoning attacks. The absence of message authentication is what leads to ARP poisoning, allowing any host within the LAN to generate forged messages. It suggests implementing a public key cryptography-based authentication scheme for ARP replies, extending it to S-ARP. It effectively eliminates ARP poisoning attempts.

4. Modular solution

This proposed solution [7] is based on multiple modules to guarantee that both internal and external ARP attacks can be prevented. Instead of focusing on the ARP table cache, this approach uses hash codes in a database to check IP-MAC mapping and detect various attacks. Framework in [8] can provide internal security measures to make all data flows checked in the build-in mechanism and different parts in the framework offer their certain functions The survey in [9] also recommends the best security measures for detection should include four requirements. Firstly, the cryptographic processing should be as little as possible. Secondly, the approach has to be compatible with ARP. Thirdly, all the ARP attacks should be considered. Fourthly, management network costs are important. Based on the idea of the framework and the survey, the modular solution is proposed.

4.1. Design

This novel ARP attack prevention approach consists of three modules to detect ARP cache poisoning more efficiently. Figure 2 [7] shows the overview of this modular approach.

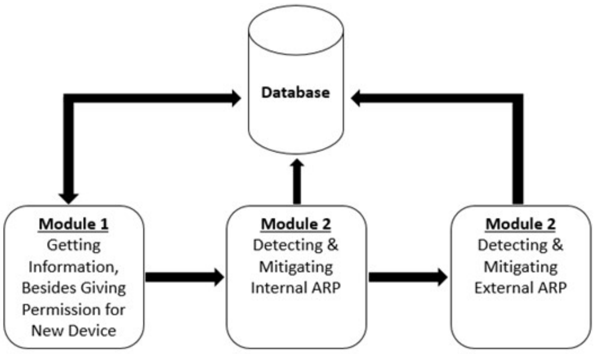

Database
Module 1
Getting Information, Besides Giving Permission for New Device
Module 2
Detecting & Mitigating Internal ARP
Module 2
Detecting & Mitigating External ARP

Figure 2. module structure overview

Module 1 stores the data information of eligible devices in the database initially, then decides to forward the packets or not before the data packets flow into the framework. The information stored in the database includes seven elements:

1) IP Address
2) MAC Address
3) Hash for IP and MAC used MD5
4) Port Number
5) Time
6) Date
7) Switch IP

The admin has the right to block packets or give permission for each device. Hash codes can represent the unique mapping of IP-MAC easily and facilitate the process of identifying types of attacks. We can also trace the attacker through the port number and switch IP. Therefore, this single database can be strong enough to fulfill various requirements. Figure 3 [7] shows how Module 1 works.

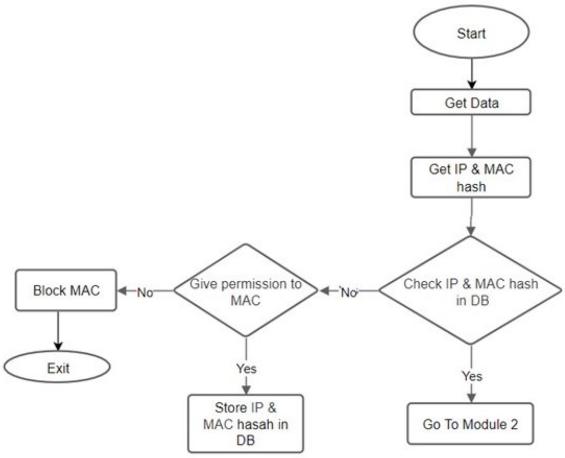


Figure 3. Module1

Internal ARP attacks can be prevented in Module 2. After checking the device information in the DataBase(DB) and assuming the information is safe, Module 2 will compare the IP-MAC stored in the ARP table with the hash of this mapping. Updating the ARP table is also included in this Module. This process can discover two suspicious conditions. Firstly, if one IP-MAC mapping exists in the ARP table but the database doesn't have its hash code, it indicates that the IP and MAC may be forged since we assumed that the information in the database is reliable. Secondly, if we can't find the IP-MAC in ARP table and the hash code of this mapping in the database, it indicates that this IP-MAC received is totally fake. Finally, packets under these two conditions will be moved to Module 3 to analyze further details. Detailed process is shown in Figure 4 [7].

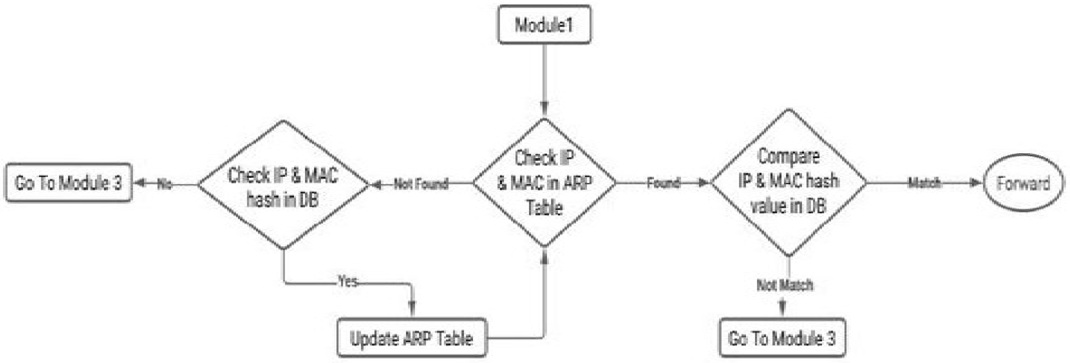


Figure 4. Module2

Module 3 can discriminate whether a MAC has more than one IP or an IP has more than one MAC. Because of the single IP-MAC-like fingerprint stored in the database, we can easily conclude which one is fake by retrieving the hash codes and forwarding packets if the MAC belongs to that IP matched according to the database. It should be noted that if we discover a MAC address has more than one IP address but no information of these IP-MAC mappings stored in the database, the admin might update the new IP with this MAC since the user may change the previous IP address with the same MAC address. Figure 5 [7] shows the flowchart of Module 3.

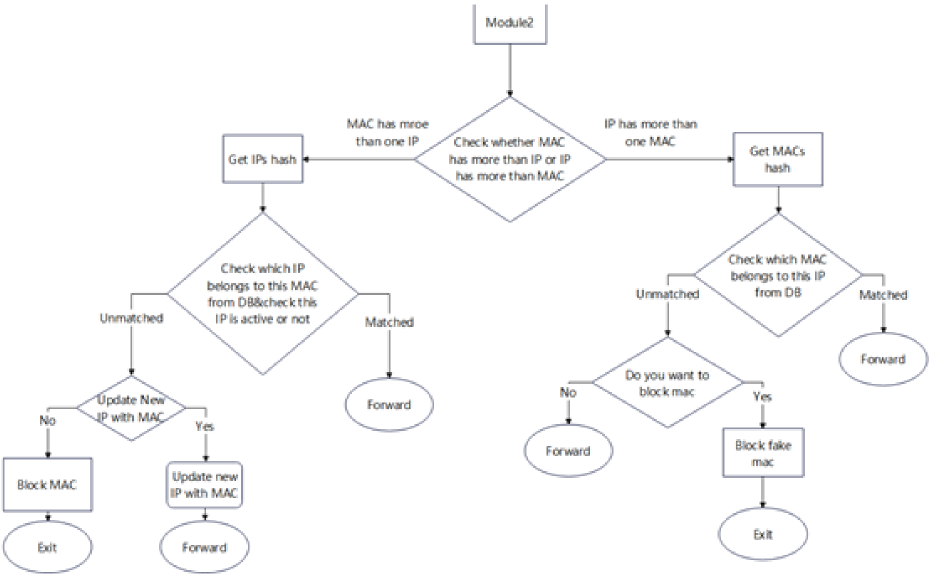


Figure 5. Module3

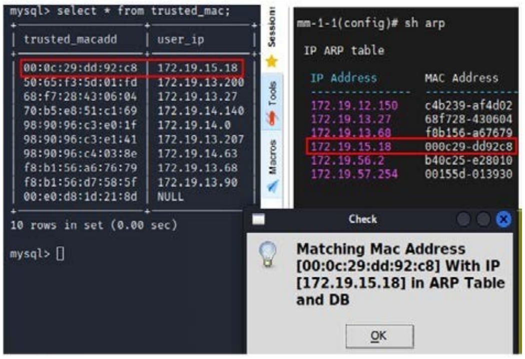
4.2. Experiment

This paper finally builds up an environment using VMWare Workstation Pro, Kali Linux, MySQL, and Python to simulate different conditions between ARP tables and Modules. The attacker is created with a Kali Linux virtual machine using Ettercap. The experiment has three phases to demonstrate the validity of the modular solution.

1). Initialization: Every new node will be checked with the database to get the connecting permission. The admin will make the decision and update the database with new device information.

2). Authentication: After building the reliable database, IP and MAC will be checked both in the ARP table and Database. Figure 6 [7]shows the checking process of one of the conditions that an IP-MAC exists in the ARP table and matches the mapping information stored in DB.

3). Poisoning Detection: The exact type of attack will be classified in module 3. Figure 7 [7] indicates that the mechanism is able to detect the suspicious IP-MAC in the ARP table and prevent it immediately.



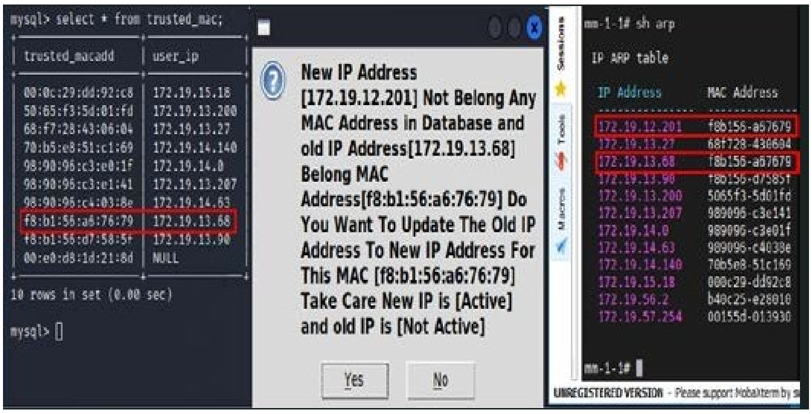
```
mysql> select * from trusted_mac;
+-------------------+----------------+
| trusted_macadd    | user_ip        |
+-------------------+----------------+
| 00:0c:29:dd:92:c8 | 172.19.15.18   |
| 50:65:f3:5d:01:fd | 172.19.13.200  |
| 68:f7:28:43:06:04 | 172.19.13.27   |
| 70:b5:e8:51:c1:69 | 172.19.14.140  |
| 98:90:96:c3:e0:1f | 172.19.14.0    |
| 98:90:96:c3:e1:41 | 172.19.13.207  |
| 98:90:96:c4:03:8e | 172.19.14.63   |
| f8:b1:56:a6:76:79 | 172.19.13.68   |
| f8:b1:56:d7:58:5f | 172.19.13.90   |
| 00:e0:d8:1d:21:8d | NULL           |
+-------------------+----------------+
10 rows in set (0.00 sec)

mysql> 
```

```
mm-1-1(config)# sh arp

 IP ARP table

  IP Address        MAC Address
  ---------------   --------------
  172.19.12.150     c4b239-af4d02
  172.19.13.27      68f728-430604
  172.19.13.68      f8b156-a67679
  172.19.15.18      000c29-dd92c8
  172.19.56.2       b40c25-e28010
  172.19.57.254     00155d-013930
```

Check

Matching Mac Address [00:0c:29:dd:92:c8] With IP [172.19.15.18] in ARP Table and DB

Figure 6. Module2 Check



```
mysql> select * from trusted_mac;
+-------------------+---------------+
| trusted_macadd    | user_ip       |
+-------------------+---------------+
| 00:0c:29:dd:92:c8 | 172.19.15.18  |
| 50:65:f3:5d:01:fd | 172.19.13.200 |
| 68:f7:28:43:06:04 | 172.19.13.27  |
| 70:b5:e8:51:c1:69 | 172.19.14.140 |
| 98:90:96:c3:e0:1f | 172.19.14.0   |
| 98:90:96:c3:e1:41 | 172.19.13.207 |
| 98:90:96:c4:03:8e | 172.19.14.63  |
| f8:b1:56:a6:76:79 | 172.19.13.68  |
| f8:b1:56:d7:58:5f | 172.19.13.90  |
| 00:e0:d8:1d:21:8d | NULL          |
+-------------------+---------------+
10 rows in set (0.00 sec)

mysql> []
```

New IP Address [172.19.12.201] Not Belong Any MAC Address in Database and old IP Address[172.19.13.68] Belong MAC Address[f8:b1:56:a6:76:79] Do You Want To Update The Old IP Address To New IP Address For This MAC [f8:b1:56:a6:76:79] Take Care New IP is [Active] and old IP is [Not Active]

```
mm-1-1# sh arp

 IP ARP table

 IP Address         MAC Address
 ---------------    ---------------
 172.19.12.201      f8b156-a67679
 172.19.13.27       68f728-430604
 172.19.13.68       f8b156-a67679
 172.19.13.90       f8b156-d7585f
 172.19.13.200      5065f3-5d01fd
 172.19.13.207      989096-c3e141
 172.19.14.0        989096-c3e01f
 172.19.14.63       989096-c4038e
 172.19.14.140      70b5e8-51c169
 172.19.15.18       000c29-dd92c8
 172.19.56.2        b40c25-e28010
 172.19.57.254      00155d-013930

mm-1-1# ▌
```

Figure 7. Module3 Detection

4.3. Conclusion

This module framework typically provides multiple layers of ARP spoofing defense mechanisms. These mechanisms can be combined to create a complex and diverse protection system. For example, it may include ARP cache monitoring, ARP packet filtering, dynamic ARP detection, and other methods, enhancing the overall defense effectiveness.

In addition, this approach can maintain the mappings between IP addresses and their corresponding MAC addresses in the network. Storing known IP-MAC mappings in the database enables faster detection of ARP spoofing. If an ARP response is received that does not match the information in the database, the system can promptly raise an alert or take appropriate defensive measures.

More importantly, it can reduce broadcast traffic in the network and manage the devices more flexibly.

5. Detection used SDN in cloud environment

This proposed approach [10] is applied with Software-Defined Networking(SDN) to protect the cloud environment from ARP attacks. The author builds a mechanism structure based on SDN's three-tier architecture and uses a cluster of controllers to manage the network. This approach can process packets at real time and monitor traffic to prevent ARP flooding.

5.1. Background

Cloud computing [11] is a hot trend in the field of information technology, consisting of a set of resources and services provided with data existing on the web for sharing among users. It can focus on the power of thousands of computers on one single problem, enabling users to finish their work faster than ever. The cloud resources are transparent to users, allowing them to access applications and data anytime and anywhere. Cloud computing has become a basic technique almost for every enterprise today. However, privacy security remains one of the biggest challenges facing cloud computing since data can be accessed from any location. ARP is the base of network communication so ARP attacks in cloud computing can cause several disasters like Data Tampering, Denial of Service, and Information Leakage. If an ARP attack can't be prevented, all the data including IP address or configuration details might be exposed to the network. Therefore, protecting cloud computing from ARP attacks has become a significant topic issue.

5.2. Using SDN

SDN has a three-tier architecture presented in Figure 8 [12].

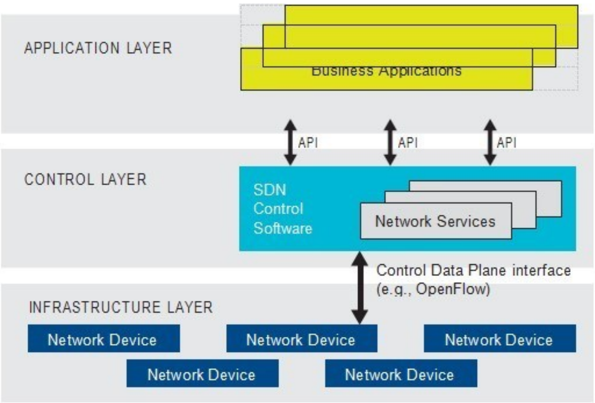


Figure 8. SDN structure

The application layer consists of different upper applications. The middle layer is composed of controllers deployed algorithms and commands to manage the network. The infrastructure layer has network devices, especially switches. The feature of SDN is that it can separate the data layer from the control layer. Controllers work with executing strategies and installing flow entries while switches just need to deliver packets following the entries. Hence, using SDN to detect ARP attacks has several merits.

1). Real-time response: SDN provides a central controller that can monitor network traffic faster and makes it easier for network admins to detect and identify potentially malicious ARP responses. It's crucial for a cloud environment since cloud devices need to adapt rapidly to changing demands.

2). Network Isolation: SDN virtualizes the network, enabling the isolation of network resources. So SDN can prevent ARP attacks from affecting specific cloud services.

3). Flexible and Programmability: SDN decouples network control from traditional network devices, making the whole network programmable and flexible. It can meet different requirements when we detect ARP attacks.

5.3. Detailed design

The proposed structure is also divided into three layers presented in Figure 9 [10].

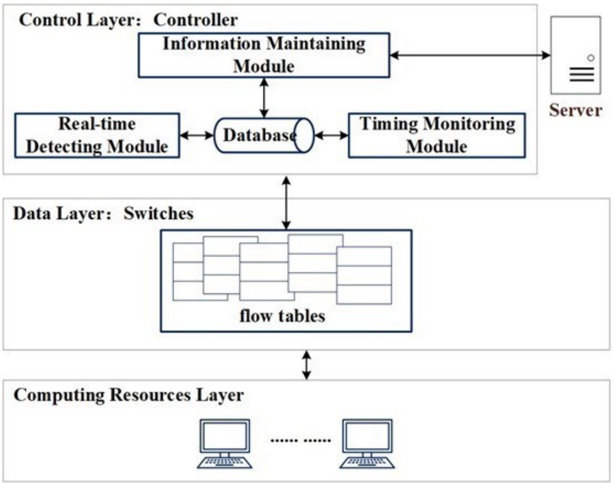

Control Layer: Controller
Information Maintaining Module
Server
Real-time Detecting Module
Database
Timing Monitoring Module
Data Layer: Switches
flow tables
Computing Resources Layer
....

Figure 9. Designed structure

1). Computing resource layer: Cloud providers can supply different computing resources. Every host maintains its own ARP table.

2). Data layer: This layer is full of network devices and OpenFlow switches [13] are applied in this layer to make communications between hosts and controllers. Flow entries reflect how a switch deals with received packets. If the switch finds an entry according to the packet, it will tackle the packet at once. Otherwise, it will send a packet to learn the next step and establish a flow entry in order to avoid repeatedly disposing of akin packets.

3). Control layer: This layer with three parts is designed to manage the whole network. The real-time detecting module can detect ARP packets delivered by switches in real time. Algorithm1 [10] shows how it works. If receiving a request packet, this part will initially check the MAC address with that in the head of the Ethernet frame. Then the authenticity of IP-MAC mappings will be checked with the information stored in the database. Finally, if the packet has a target IP existing in the database, it will be considered valid and the controller updates the 'mac to port' table and finds the output port to forward the packet. The disposal of ARP reply packets is similar to that of dealing with request packets. Real-time modules can prevent ARP spoofing attacks by detecting and dropping forged packets immediately. Timing Monitoring Module monitors statistical data on edge switches and it can prevent ARP flooding typically. Algorithm 2 [10] shows how it detects the ARP flooding. A cluster of controllers will send requests to edge switches once in a while and conclude if ARP flooding occurs by comparing the number of replies from switches with an average threshold. Flow entries are also applied to block harmful packets. Information Maintaining module is designed to collect different information including statistical data from edge switches and reliable mappings of IP and MAC from all hosts. Controllers extract the IP-MAC mapping only from the Dynamic Host Configuration Protocol(DHCP) Acknowledge(ACK) packet. Otherwise, controllers will send the DHCP discover packets or request packets to the DHCP server and firm the authenticity of these packets. Finally, controllers will transport reliable packets to the DHCP server and broadcast the IP-MAC to other controllers. This module can also receive information from other controller servers with communications. Figure 10 [10] shows the main process of the Information Maintaining Module.

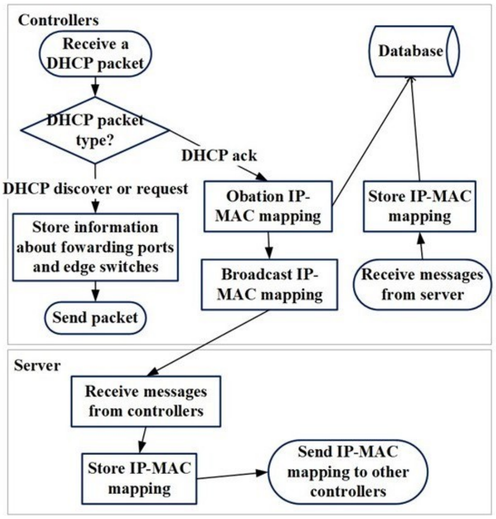


Figure 10. Information Maintaining Module Disposal

Each controller has several tables with multiple functions. Table 'arp list' stores accurate IP-MAC mappings of hosts extracted from DHCP ACK packets. Table 'mac port' has the path information to forward a packet to the destination. Table 'edge port' includes the statistical data of edge switches used for analysis of ARP flooding. These tables guarantee that controllers have enough information to make attack detection, judgments and implement a traffic verification mechanism to ensure that only authorized devices can send valid ARP requests and responses. At the same time, controllers can set access policies to restrict ARP communications between different devices and reduce potential ARP attack entry points.

5.4. Implementation and evaluation

This paper finally builds a cloud network using Mininet and chooses Ryu as controllers to monitor the network under a topography presented in Figure 11 [10] including two controllers and ten OpenFlow switches.

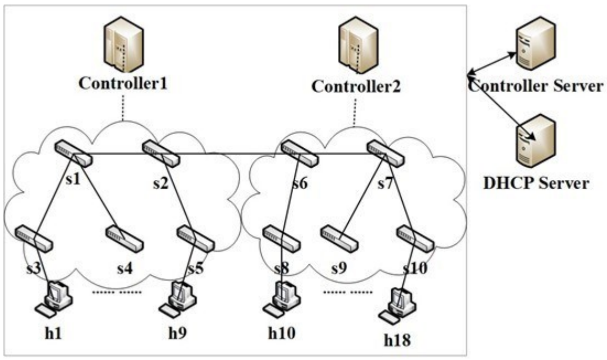


Figure 11. Testing topography

After simulating ARP spoofing by using ‘arp spoof’, the controller can detect and drop forged packets in a short time nearly 190 milliseconds with different numbers of hosts. As for ARP flooding, the proposed approach has less response time compared with normal SDN solutions since the IP-MAC of hosts has been collected before communications between controllers and hosts. Compared with normal solutions, it also has less ARP response time in Figure 12 [10].

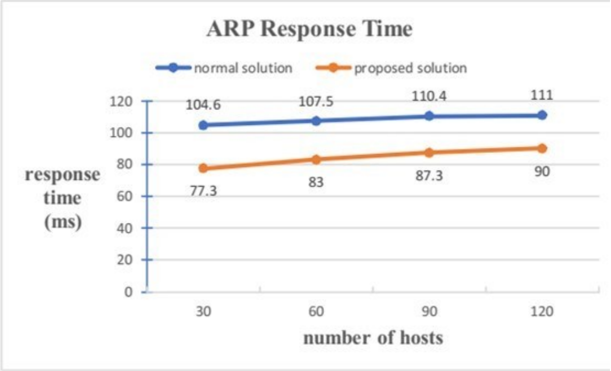


Figure 12. performance

5.5. Conclusion and outlook

In this proposed solution, nodes within the cluster of controllers can share resources and status information. In this way, controllers in the network have the same network view and data, ensuring consistency for ARP defense in a cloud environment. In addition, it simplifies network admins' work since they can manage the whole cluster as a unified entity and eliminate the need to manage each node individually.

However, different numbers of controllers have different performances of protecting the cloud environment according to the experiment result. If we can find the most appropriate number of controllers in a certain network, the efficiency of the solution might be more noticeable.

6. Comparison

As for security, S-ARP can effectively tackle the lack of packet authentication since certified public/private key pairs can guarantee unique identity and IP-MAC mapping security. It can also adapt the dynamic IP address by signing the request with the old key.

As for cost and complexity, compared with S-ARP, the second solution can simplify the detection of ARP attacks. Using reliable information stored in the database, the mechanism can check the IP-MAC mapping easily, reducing the extra costs. The application of the framework also expands the detection system when dealing with multiple attacks.

As for adaptability, the detection used SDN can tackle the ARP attacks in a new trend. Traditional solutions can't be applied in cloud computing since cloud computing environments involve the dynamic creation, deletion, and migration of Virtual Machines (VMs) and containers. What's more, traditional ARP attack defense often relies on LAN broadcast mechanisms, but in the cloud environment, the broadcast can increase the attack surface, making ARP attacks more challenging to defend. However, using controllers can dynamically manage and optimize the flow of packets based on network traffic conditions. If an ARP attack is detected, the controller can swiftly adjust network routing to take actions which is necessary for cloud computing security.

7. Conclusion

In our paper, we first introduced S-ARP, a secure alternative to ARP, to offer robust protection against ARP poisoning attacks. Every host on the LAN possesses a certified public/private key pair issued by a trusted local authority, known as the Certification Authority. To guarantee the integrity and authenticity of messages, the sender digitally signs each message. This cryptographic process prevents the injection of false or manipulated information, including spurious and spoofed data. After this, the following paper gives another approach including three modules, and the architecture incorporates a highly efficient and feature-rich database system that provides excellent functionality and reliable support for storing and managing ARP table information. The last paper shows the implementation of SDN coupled with cloud computing provides a proactive defence mechanism against ARP attacks. This approach enhances the security of cloud computing environments by implementing measures to prevent and mitigate ARP-based threats. This method has the capability to process network packets in real time while also monitoring traffic to effectively mitigate ARP flooding. We believe that there are good prospects for development in these three directions in the future.

Acknowledgment

Qijun Li and Yipeng Dong contributed equally to this work and should be considered co-first authors.

References

[1] Statista. Number of internet and social media users worldwide as of April 2023(in billions)), 2023.

[2] D Srinath, S Panimalar, A Jerrin Simla, and J Deepa. Detection and prevention of arp spoofing using centralized server. *International Journal of Computer Applications*, 113(19), 2015.

[3] Chunhong Shi. A brief discussion of arp attacks and their defenses. *Computer Knowledge and Technology*, (10X):26–27, 2016.

[4] Xiaoping Qin and Xianglei Zhang. Arp attacks and protection in local area networks. *Science Technology Vision*, 25, 2018.

[5] Ting Cui. Attacks and preventive measures for arp in lans. *Network Security Technology Application*, (5):26–27, 2019.

[6] Danilo Bruschi, Alberto Ornaghi, and Emilia Rosti. S-arp: a secure address resolution protocol. In *19th Annual Computer Security Appli- cations Conference, 2003. Proceedings.*, pages 66–74. IEEE, 2003.

[7] Ahmed A Galal, Atef Z Ghalwash, and Mona Nasr. A new approach for detecting and mitigating address resolution protocol (arp) poisoning. *International Journal of Advanced Computer Science and Applications*, 13(6), 2022.

[8] Debadyuti Bhattacharya, N Sri Hari Karthick, Prem Suresh, and N Bha- laji. Detecsec: A framework to detect and mitigate arp cache poisoning attacks. In *Evolutionary Computing and*

Mobile Sustainable Networks: Proceedings of ICECMSN 2021, pages 997–1007. Springer, 2022.

[9] Sherin Hijazi and Mohammad S Obaidat. Address resolution protocol spoofing attacks and security approaches: A survey. *Security and Privacy*, 2(1):e49, 2019.

[10] Sixian Sun, Xiao Fu, Bin Luo, and Xiaojiang Du. Detecting and mitigating arp attacks in sdn-based cloud environment. In *IEEE INFO- COM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 659–664. IEEE, 2020.

[11] Matthew NO Sadiku, Sarhan M Musa, and Omonowo D Momoh. Cloud computing: opportunities and challenges. *IEEE potentials*, 33(1):34–36, 2014.

[12] V Thirupathi, CH Sandeep, Naresh Kumar, and P Pramod Kumar. A comprehensive review on sdn architecture, applications and major benifits of sdn. *International Journal of Advanced Science and Tech- nology*, 28(20):607–614, 2019.

[13] Alexander Gelberger, Niv Yemini, and Ran Giladi. Performance analysis of software-defined networking (sdn). In *2013 IEEE 21st International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 389–393. IEEE, 2013.