*2011 International Conference on Telecommunication Technology and Applications*
*Proc .of CSIT vol.5 (2011) © (2011) IACSIT Press, Singapore*

# P-ARP: A novel enhanced authentication scheme for securing ARP

P. Limmaneewichid and W. Lilakiatsakun

Faculty of Information Science and Technology
Mahanakorn University of Technology
Bangkok, 10530, Thailand

**Abstract.** This paper proposes a novel ARP authentication scheme based on an ARP authentication trailer and a new technique for hiding the target IP address in an ARP request message. The scheme provides a method to defend ARP attacks. In addition to avoid ARP attacks, the proposed scheme is backward compatible with existing ARP, no additional header fields are added to the protocol. The scheme supports both dynamic and static IP address assignment. Our experimental analysis shows that the proposed scheme can successfully defend various ARP attacks and can also improve overall network security without the need for complex configuration/installation or an additional server.

**Keywords:** ARP, Cache Poisoning, Spoofing

## 1. Introduction

Address Resolution Protocol (ARP) [1] is used to map IP address to MAC address in a local area network segment. Unfortunately, ARP is a stateless protocol and there is absolutely no authentication to determine whether the reply actually comes from the correct host or not. Thus, malicious users on your LAN can easily manipulate ARP cache table [2, 3, 4].

This paper proposes a novel authentication scheme based on an ARP authentication trailer and a new technique for hiding the target IP address in an ARP request message to avoid mentioned problems. The ARP authentication trailer contains a magic number, nonce, and the authentication data (see Section 4), which is the output of the HMAC hash function. For target IP hiding technique, a target IP address in the ARP request message is filled with zero. Each host on the LAN can check the target IP address in a request against its own IP address by matching its own computed ARP authentication data with an authentication data in the received ARP request packet (see Section 4). We call our enhanced authentication scheme for securing ARP “P-ARP”.

The rest of the paper is organized as the following. Section 2 provides some background about ARP attacks and techniques that are used. Section 3 provides an existing mechanism done in this area. Section 4 proposed scheme. Conclusion and future work is shown in Section 5.

## 2. BACKGROUND

### 2.1. ARP Cache Poisoning

ARP Cache Poisoning [5, 6], also known as ARP Spoofing, is a technique used to attack Ethernet wired or wireless networks. Any traffic meant for the compromised IP address will be mistakenly sent to the attacker host instead. The attacker could then either choose to forward the traffic to the actual default gateway (passive sniffing) or modify the data before forwarding it (man-in-the-middle attack). The attacker could also launch a denial-of-service attack against a victim by associating a nonexistent MAC address to the IP address of the victim's default gateway. Arpspoof [7], Arpoison [8] and Ettercap [9] are some of the tools that can be used to carry out ARP poisoning attacks.

## 2.2. Trailer Protocol

The trailer protocol [10] is a link-layer encapsulation technique that rearranges the data contents of packets sent on the physical network. Even though the "trailer ARP reply" packet is deprecated [11], it is still functional. As illustrated below the packet capture on the Windows 7 machine shows the ARP Reply with 18-byte trailer.

```
Frame 514: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
Ethernet II, Src: AlphaNet_39:af:15 (00:0f:a3:39:af:15), Dst: IntelCor_4a:a4:3c
(00:22:fb:4a:a4:3c)
    ...
    Type: ARP (0x0806)
    Trailer: 000000000000000000000000000000000000
Address Resolution Protocol (reply)
    ...
```

In this paper, we introduce a set of functionalities by using the 18-byte trailer to provide an authentication feature. Arpscan [12] is a proof of concept tool to send ARP packets with an additional trailer to hosts on the local network and display any responses that are received.

## 2.3. Stateful ARP Cache

Stateful ARP cache [13] is proposed to improve ARP cache security. When host A generates an ARP request to get the MAC address of host B, an entry is created in its stateful ARP cache, with the status of ”Waiting”. Host A will wait for an ARP reply within a predefined period of time. If an ARP reply comes, then host A waits for another timeout to be expired in order to collect other possible ARP replies sent by other hosts in the network. Note that if host A receives more than one ARP reply, it means that most likely more than one host has replied. Therefore, among those hosts, only one host is an honest host, which is host B. The others are probably malicious hosts, performing ARP cache poisoning attack to corrupt the ARP cache of host A.

The main differences between the current stateless ARP cache and the proposed stateful ARP cache are followings:

1) If a host receives an ARP reply, then the stateful ARP cache will not update the corresponding entry unless an ARP request has been generated before for that entry.

2) The stateful ARP cache will not update its entries from gratuitous ARP [22].

For our technique, we also implement stateful ARP cache that is embedded a cryptographic hash function verification to select the most trusted packet that will be used to update host A’s ARP cache.

# 3. RELATED WORK

It has already been known that ARP is most vulnerability of layer 2 security, several schemes have been proposed to mitigate and prevent ARP attacks, and however each individual has some limitations [14].

A simple scheme is to use static entries in the ARP cache [14], but it does not work in dynamic environments. Gouda et al. [15] requires changing the ARP protocol implementation of every host. The middleware approach proposed by Tripunitara et al. [16] requires changes on all the hosts in the network. S-ARP [17] relies on public-key cryptography to authenticate ARP replies. Additionally, a certification authority (called AKD) and a modification of DHCP called S-DHCP are required to make the scheme work properly. TARP [18] implements security by distributing centrally secure MAC/IP address mapping attestations (called tickets). These tickets are generated and signed by a Local Ticket Agent (LTA). In a dynamic IP network, the DHCP server needs to perform the functionality of the LTA. Goyal et al. [19] proposed a new architecture for secure address resolution, but it is not backward compatible with ARP and it can be very inefficient in highly dynamic networks. Goyal et al. [20] also proposed a modification to S-ARP based on the combination of digital signatures and one time passwords. Nevertheless, S-ARP problems are still unsolved.

For several disadvantages of existing methods mentioned above, we propose the novel scheme called P-ARP. It provides an authentication scheme for ARP packets to defend various ARP attacks and can also improve overall network security without the need for complex configuration/installation or an additional server.

# 4. P-ARP

## 4.1. ARP Authentication Trailer Format

In the proposed scheme, we use standard ARP request/reply packets. We only add an authentication data, in an ARP trailer [10] as illustrated in Figure 1.

To maintain compatibility with the original ARP specified in [1], Request/Response mechanisms of standard ARP is reserved.

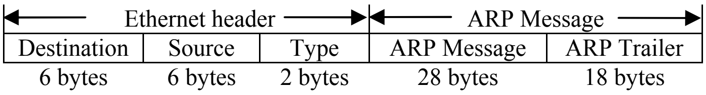

Fig. 1: ARP Packet Format

For the proposed method, so called the cryptographic trailer based authentication scheme for ARP, we make use of trailer protocol [10]. The trailer consists of 3 fields that are the *Magic Number*, *Nonce* and *Authentication Data* as shown in Figure 2.

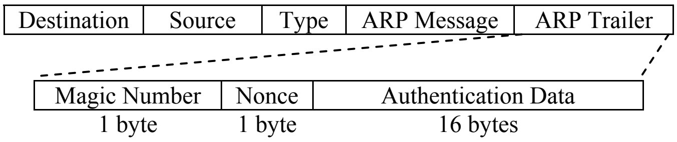

Fig. 2: ARP Authentication Trailer Format

The field '*Magic Number*' is used to distinguish whether an ARP packet carries an ARP authentication trailer. The Magic Number is a fixed constant defined as hexadecimal 0x22. It is used to distinguish whether a trailer carries the Authentication Data. The field '*Nonce*' is an unsigned random number that is used to prevent replay attacks. Finally, the field '*Authentication Data*' is the output of the keyed cryptographic hash function used to validate received ARP reply packets (see Section 4.2).

## 4.2. ARP Authentication Mechanism

As explained previously, the authentication data is inserted in ARP packets. We use keyed one-way function algorithm, HMAC-MD5 with 128-bit keys. HMAC-MD5 requires a shared secret key '*K*' for hosts that wish to use the scheme. The shared secret key '*K*' is combined with Nonce '*N*' and the peer-address attribute '*PA*' before the hashing operation. The result from HMAC-MD5 function is placed in the Authentication Data field. In order to validate ARP message, the receiver checks the Authentication Data contained in incoming ARP reply message. If the received Authentication data does not match its own computed ARP Authentication data, the ARP message will be discarded.

In Table 1, we summarize the notations that are used throughout this paper.

TABLE 1: Summary of notations.

| Notation | Description |
|---|---|
| $A$ | The host that sends out a broadcast ARP request packet. |
| $B$ | The host that sends back the ARP reply packet to *A*. |
| $K$ | The shared secret key for authentication. |
| $N_A$,$N_B$ | The random nonce chosen by *A* and *B*. |
| $M_A$,$M_B$ | The MAC address of *A* and *B*. |
| $IP_A$,$IP_B$ | The IP address of *A* and *B*. |
| $PA_A$,$PA_B$ | The peer-address attribute generated by *A* and *B*. |
| $AT_A$, $AT_B$ | The Authentication Data is computed by *A* and *B* |
| \|\| | The Concatenation |

When host *A* needs to find the MAC address of host *B*, the scenario is:

1) Host *A* computes the $AT_A$ using *K*, $N_A$ and *PA* as the following equation (1) and (2):

$$PA_A = M_A || IP_A || IP_B \quad (1)$$

$$AT_A = HMAC\ (K, N_A \ || \ PA_A) \quad (2)$$

For target IP hiding purpose, $PA_A$ includes the sender IP address, the sender MAC address, and also a target IP address.

2) Host $A$ generates an ARP request packet with the ARP Authentication Trailer and the target IP 0.0.0.0 as shown in Figure 3, then an entry in the stateful ARP cache is created, with the status of “Waiting” [13] as shown in Figure 4.

<table>
<tr><td colspan="9">Ethernet II</td></tr>
<tr><td colspan="3">FFFFFFFFFFFF</td><td colspan="4">$M_A$</td><td colspan="2">0x0806</td></tr>
<tr><td colspan="9">ARP-Request Message</td></tr>
<tr><td colspan="3">0x0001</td><td colspan="2">0x0800</td><td>6</td><td>4</td><td colspan="2">0x0001</td></tr>
<tr><td colspan="3">$M_A$</td><td>$IP_A$</td><td colspan="4">000000000000</td><td><i>0.0.0.0</i></td></tr>
<tr><td>0x22</td><td>$N_A$</td><td colspan="7">$AT_A$</td></tr>
</table>

Fig. 3: ARP Request Packet

| IP Address | MAC Address | Status |
|---|---|---|
| $IP_B$ | 000000000000 | *Waiting* |

Fig. 4: An entry in the stateful ARP cache

Then, host $A$ waits for an ARP reply, within a predefined timeout.

3) Once the host $B$, whose IP address matches the contents of the Target IP Address $IP_B$ of the ARP message, receives the ARP Request packet. Host $B$ indicates whether or not the packet is encapsulated using trailers.

- If the *Magic Number* field in an ARP Authentication Trailer is 0x22 and the target IP in ARP request message is 0.0.0.0, then host $B$ will compute the $AT_A$ using $K$, $N_A$ and $PA_A$ as equation (1) and (2).

- If the received $AT_A$ does not match the computed $AT_A$ value, the host $B$ discards the packet due to malicious attack or the packet is not for the host $B$. Otherwise, the host $B$ computes the $AT_B$ using $K$, $N_B$ and $PA_B$ as the following equation (3) and (4):

$$PA_B = M_B||IP_B||IP_A \tag{3}$$

$$AT_B = HMAC\ (K,\ N_B\ ||\ PA_B) \tag{4}$$

Next, Host $B$ will generate an ARP reply packet with the ARP Authentication Trailer as shown in Figure 5 and sends back to Host $A$.

<table>
<tr><td colspan="9">Ethernet II</td></tr>
<tr><td colspan="3">$M_A$</td><td colspan="4">$M_B$</td><td colspan="2">0x0806</td></tr>
<tr><td colspan="9">ARP-Reply Message</td></tr>
<tr><td colspan="3">0x0001</td><td colspan="2">0x0800</td><td>6</td><td>4</td><td colspan="2">0x0002</td></tr>
<tr><td colspan="3">$M_B$</td><td>$IP_B$</td><td colspan="4">$M_A$</td><td>$IP_A$</td></tr>
<tr><td>0x22</td><td>$N_B$</td><td colspan="7">$AT_B$</td></tr>
</table>

Fig. 5: ARP Request Packet

- For networks that are not using the scheme, all ARP packets are not encapsulated by the ARP authentication trailer, host $B$ thus generates a normal ARP reply packet to host $A$.

- The malicious node can receive ARP request message but cannot recognize the target IP address.

4) When host $A$ receives an ARP reply packet from host $B$, it determines whether the packet is encapsulated using trailers or not.

- If *Magic Number* field in an ARP Authentication Trailer is 0x22, then Host $A$ will compute the $AT_B$ using $K$, $N_B$ and $PA_B$ as equation (1) and (2).

- If the received $AT_B$ does not match the computed $AT_B$ value or the timeout is expired, the host $A$ discards the packet. Otherwise, host $A$ will update the stateful ARP cache entries.

- In case of host $A$ has received more than one ARP reply packets, only host that ARP reply messages passing the test will be selected to update the ARP cache.

## 5. CONCLUSION

The scheme can successfully defend various ARP attacks such as poisoning, spoofing and etc. As a result, overall network security can be improved. The scheme lowers the risk of these attacks by providing a novel authentication mechanism to verify ARP packets, and by protecting the ARP cache from potentially malicious updates. In addition, the target IP hiding technique can be used for preventing an attacker to construct a forged ARP reply packet, which will in turn mitigate ARP poisoning attack.

For the further study, we will analyze more in-depth in different shared-key selection and distribution schemes. For instance, Kungpisdan et al. [21] proposed an offline session key generation scheme. In addition, we are investigating how to hide out information in an ARP message to protect eavesdropping from attackers.

## 6. Acknowledgements

The author thanks Mahanakorn University of Technology (MUT) for financial support. Special thanks were given to Dr. Supakorn Kungpisdan for his useful comment.

## 7. References

[1] D. Plummer. An Ethernet address resolution protocol, Nov. 1982. *RFC 826*.

[2] M. Carnut and J. Gondim. ARP spoofing detection on switched Ethernet networks: A feasibility study. In *Proceedings of the 5th Simp´osio Seguranc¸a em Inform´atica*, Nov. 2003.

[3] T. Demuth and A. Leitner. ARP spoofing and poisoning: Traffic tricks. Linux Magazine, 56:26–31, July 2005.

[4] S. Whalen. An introduction to ARP spoofing. 2600: The Hacker Quarterly, *http://www.node99.org*

[5] Anatomy of an arp poisoning attack. *http://www.watchguard.com /infocenter/editorial/135324.asp*.

[6] B. Fleck and J. Dimov. Wireless access points and arp poisoning. *http://downloads.securityfocus.com/library*

[7] Dug Song, "Arpspoof", *http://arpspoof.sourceforge.net*

[8] Steve Buer, "Arpoison", *http:// www.arpoison.net*

[9] Alberto Ornaghi, "Ettercap", *http://ettercap.sourceforge.net*

[10] R. Braden. Requirements for Internet Hosts - Communication Layers, Oct. 1989. *RFC 1122*.

[11] R. W. Stevens. *TCP/IP Illustrated, vol 1*. Addison Wesley, 2001.

[12] Roy Hills and NTA Monitor Ltd., "ARP scanning and fingerprinting tool", *http://www.nta-monitor.com*

[13] Zouheir Trabelsi and Wassim El-Hajj, "Preventing ARP Attacks using a Fuzzy-based Stateful ARP Cache", *the IEEE International Conference on Communications (IEEE ICC 2007)*, 24-28 June 2007, Glasgow, Scotland, UK.

[14] Cristina L. Abad, Rafael I. Bonilla, "An Analysis on the Schemes for Detecting and Preventing ARP Cache Poisoning Attacks," icdcsw, pp.60, *27th International Conference on Distributed Computing Systems Workshops (ICDCSW'07)*, 2007.

[15] M. Gouda and C.-T. Huang. A secure address resolution protocol. *Computer Networks*, 41(1):57–71, Jan. 2003.

[16] M. Tripunitara and P. Dutta. A middleware approach to asynchronous and backward compatible detection and prevention of ARP cache poisoning. In *Proceedings of the 15th Annual Computer Security Applications Conference (ACSAC'99)*, Dec. 1999.

[17] D. Bruschi, A. Ornaghi, and E. Rosti. S-ARP: A secure address resolution protocol. In *Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC '03)*, Dec. 2003.

[18] W. Lootah, W. Enck, and P. McDaniel. TARP: Ticket-based address resolution protocol. In *Proceedings of the 21st Annual Computer Security Applications Conference (ACSAC'05)*, Dec. 2005.

[19] V. Goyal, V. Kumar, and M. Singh. A new architecture for address resolution, 2005. *http://www.itbhu.ac.in*.

[20] V. Goyal and A. Abraham. An efficient solution to the ARP cache poisoning problem. In *Proceedings of the 10th Australasian Conference on Information Security and Privacy (ACISP '05)*, published in Lecture Notes in Computer Science (LNCS 3574), pages 40–51, July 2005.

[21] S. Kungpisdan and S. Metheekul, A Secure Offline Key Generation With Protection Against Key Compromise, *Proceedings of the 13th World Multiconference on Systemics, Cybernetics and Informatics (WMSCI 2009)*, Orlando, Florida, USA, July 10-13, 2009, pp. 63-67.

[22] S. Cheshire. Dynamic Configuration of IPv4 Link-Local Addresses, May. 2005. *RFC 3927*.