KIPS Tr. Comp. and Comm. Sys.
Vol.6, No.4 pp.203~210 pISSN: 2287-5891

https://doi.org/10.3745/KTCCS.2017.6.4.203

Address Auto-Resolution Network System for Neutralizing ARP-Based Attacks

RhongHo Jang[†] · KyungHee Lee[††] · DaeHun Nyang[†††] · HeungYoul Youm[††††]

ABSTRACT

Address resolution protocol (ARP) is used for binding a logical address to a physical address in many network technologies. However, since ARP is an stateless protocol, it always abused for performing ARP-based attacks. Researchers presented many technologies to improve ARP protocol, but most of them require a high implementation cost or scarify the network performance for improving security of ARP protocol. In this paper, we present an address auto-resoultion (AAR) network system to neutralize the ARP-based attacks. The AAR turns off the communication function of ARP messages(e.g. request and reply), but does not disable the ARP table. In our system, the MAC address of destination was designed to be derived from destination IP address so that the ARP table can be managed statically without prior knowledge (e.g., IP and MAC address pairs). In general, the AAR is safe from the ARP-based attacks since it disables the ARP messages and saves network traffics due to so.

Keywords : ARP Spoofing, ARP Message Disable, Static ARP Lable, Network Bandwidth Saving, MAC Address Derivation

ARP 기반 공격의 무력화를 위한 주소 자동 결정 네트워크 시스템

장 룡 호[†] · 이 경 희[††] · 양 대 헌[†††] · 염 흥 열[††††]

요 약

주소 결정 프로토콜, ARP(Address Resolution Protocol)는 네트워크의 논리 주소(IP)로부터 물리 주소(MAC)를 찾아내는 프로토콜로 많은 네트워크에 사용되고 있다. 그러나 ARP는 메시지 내용의 진위를 검증할 수 있는 수단이 없기 때문에, 이를 이용한 공격이 많이 이루어지고 있다. 많은 연구자들이 ARP의 안전성을 보완하는 많은 기술들을 제안하였지만, 대부분은 ARP의 안전성을 향상하는 대신에 네트워크의 성능을 희생하거나 많은 구현 비용을 요구한다. 이 논문에서는 ARP 기반의 공격을 무력화하는 AAR(Address Auto-Resolution, 주소 자동 결정) 네트워크를 제안한다. AAR에서는 기존의 ARP 테이블을 사용하지 않는 것이 아니라 요청(request), 응답(reply) 등 기본적인 ARP 메시지를 사용하지 않으며, MAC 주소를 IP 주소로부터 유도할 수 있도록 설계하여 ARP 테이블을 사전지식(IP, MAC 주소의 쌍) 없이 정적으로 관리할 수 있게 한다. 이 시스템은 기본적인 ARP 메시지 기능을 차단하므로 ARP 기반의 공격을 무력화할 수 있는 동시에 네트워크 트래픽을 절약하는 효과도 있다.

키워드 : ARP 스푸핑, ARP 메시지 차단, 정적 ARP 테이블, 네트워크 트래픽 절약, MAC 주소 유도

1. 서 론

ARP(Address Resolution Protocol)은 데이터 링크 계층에서 네트워크의 논리 주소(IP 주소)를 물리 주소(MAC 주소)로 바인딩 하는 프로토콜이다. 정상적으로 네트워크 통신이 이루어지기 위해서는 호스트들의 IP들과 MAC 주소를 바인딩 하여 저장하는 ARP 테이블(캐시)이 필요하며 ARP 요청(request) 및 응답(reply) 메시지에 통해 관리된다. 그러나 ARP 메시지는 누가 보냈는지를 검증할 수 있는 수단이 없기 때문에, 만약 같은 네트워크에 있는 호스트가 변조된 ARP 메시지를 보낼 경우 그 메시지를 받은 사용자는 잘못된 MAC 주소로 데이터를 보내게 된다. 이를 이용하여 ARP 스푸핑(spoofing) 공격을 수행할 수 있으며, 성공했을 경우 피해자는 중간자(Man-in-the-Middle) 공격 등 다양한 보안 위협에 노출되게 된다. ARP 스푸핑 공격은 오래된 네트워크 보안 이슈임에도 불구하고, 공격을 수행하기 위해서 많

※ 이 논문은 2016년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No. R0127-16-1051, IoT 환경에서 프라이버시 보호 국제 표준화).
† 준 회 원 : 인하대학교 컴퓨터공학부 박사과정
†††† 종신회원 : 수원대학교 전기공학과 부교수
††† 정 회 원 : 인하대학교 컴퓨터정보공학과 교수
†††† 비 회 원 : 순천향대학교 정보보호학과 정교수
Manuscript Received : January 18, 2017
First Revision : February 8, 2017
Accepted : February 20, 2017
* Corresponding Author : HeungYoul Youm(hyyoum@sch.ac.kr)

은 전문적인 지식이 필요하지 않으며 공격에 필요한 소프트웨어도 인터넷을 통해 쉽게 찾을 수 있으며, 공격 사례도 쉽게 찾을 수 있다[1]. 현재까지 ARP 스푸핑 공격을 탐지 및 방어하기 위해 SARP[2], ASA[3], Invite-accept[4] 등 다양한 기술들이 제안 되었지만, 이들은 높은 구현 비용을 요구하거나 네트워크 성능을 희생하여 보안성을 향상하는 경우가 대부분이다. 이런 이유에서 ARP 공격에 대한 연구가 아직까지도 지속적으로 진행하고 있다[5-11].

이 논문에서는 ARP 기반의 공격을 무력화하기 위한 주소 자동 결정(AAR, Address Auto-Resolution) 네트워크를 제안한다. 이 시스템에서는 동적 ARP 테이블 대신 정적 ARP 테이블을 사용하고, ARP 메시지를 사용하지 않기 때문에 ARP 메시지 기반의 공격을 무력화한다. 정적 ARP 테이블을 사용하려면 네트워크의 모든 호스트들의 IP 주소와 MAC 주소의 쌍에 대한 정보가 필요하지만, 이 시스템에서는 MAC 주소를 IP 주소로부터 유도할 수 있도록 설계하여, 정보 수집을 위한 인력 낭비가 발생하지 않는다. 부가적으로 ARP 메시지를 사용하지 않기 때문에, ARP 메시지가 사용하던 네트워크 트래픽을 절약할 수 있다.

2. 위협 모델 및 관련 연구

2.1 위협 모델

네트워크 사용자는 네트워크에 있는 다른 디바이스들과 통신하기 위해서 디바이스들의 논리 주소(IP 주소)뿐만 아니라 해당 주소를 사용하는 네트워크 디바이스의 물리 주소(MAC 주소)도 알아야한다. 이러한 IP 주소와 MAC 주소의 바인딩 정보들은 캐시 형태로 ARP 테이블에 저장이 되고, ARP 요청 및 응답을 통해 관리된다. ARP 요청은 특정 IP 주소를 가진 디바이스의 MAC 주소 정보를 요청하는 메시지이며 네트워크의 모든 디바이스에게 전송한다. 해당 정보를 가지고 있는 디바이스는 ARP 응답 메시지로 응답한다.

현재 사용 중인 ARP는 ARP 요청 및 응답 메시지 내용에 대한 검증을 수행하지 않고 있다. 따라서 어떤 사용자가 악의적으로 ARP 응답 메시지에 잘못된 정보를 담아 응답하면, 정보를 요청한 주체는 잘못된 IP 및 MAC 주소의 바인딩 정보를 ARP 캐시 테이블에 저장하여 다른 사용자한테 데이터를 보내게 되는데, 이러한 공격을 ARP 스푸핑(spoofing)이라고 한다.

Fig. 1은 ARP 스푸핑을 이용한 중간자(Man-in-the-Middle) 공격을 설명하고 있다. 공격자 PC-3가 PC-2에게 게이트웨이의 MAC 주소는 MAC-3이라고 지속적으로 응답하면서, 게이트웨이에게 PC-2의 MAC 주소가 MAC-3이라고 지속적으로 응답하면, PC-2와 게이트웨이의 ARP 캐시 테이블은 잘못된 정보로 업데이트 된다. 즉, PC-2와 게이트웨이는 서로에게 보내야하는 데이터를 모두 공격자 PC-3에게 보내게 되며, 공격자는 패킷 릴레이(relay) 기능을 수행하면서 PC-2와 게이트웨이의 통신을 도청할 수 있다.

이러한 ARP 기반의 공격을 방어하는 기법들은 암호학적

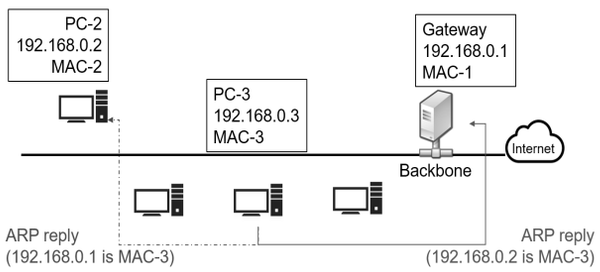


Fig. 1. Example of Man-in-the-Middle Attack Using ARP Spoofing

(cryptographic) 또는 비암호학적(non-cryptographic) 접근으로 분류할 수 있다. 암호학적 접근 방법은 표준 ARP의 구조를 변경하여 사용하는 것이 특징이며 암호화 및 복호화 등 기능을 추가적으로 수행하기 때문에 비암호학적 접근 방법보다 더 많은 연산이 필요로 한다. 비암호학적 접근 방법은 표준 ARP의 구조를 변경하지 않아 구현이 상대적으로 용이하다.

2.2 암호학적 방어 기법

SARP(Secure ARP)는 공개키(public key) 기반의 ARP 인증 시스템이다[2]. SARP에서는 모든 호스트가 자신의 IP 주소와 공개키를 AKD(Authoritative Key Distributor)에게 전달하며, AKD는 호스트들의 요청에 따라 IP 주소에 해당하는 공개키를 알려준다. AKD를 구성하기 위해서는 추가적인 장치가 필요하며, 비교적 더 많은 계산이 요구된다.

ASA(Agent-based Secure ARP)는 암호화된 UDP(User Datagram Protocol) 패킷을 ARP 패킷 대신 이용하여 ASA 에이전트들 사이에서 ARP의 역할을 수행한다[3]. ASA 에이전트에 저장되어있는 암호화된 정보는 ARP 테이블을 업데이트 하는데 사용된다.

2.3 비암호학적 방어 기법

Gouda 등은 ARP 프로토콜 외에 초대 수락(invite-accept) 프로토콜을 추가적으로 사용하는 방법을 제안하였다[4]. 초대 수락 프로토콜은 이더넷 통신을 하는 모든 호스트에서 동작하며 IP 주소와 MAC 주소 쌍의 유효성을 검증한다. 이 기법은 추가적인 프로토콜을 사용하기 때문에 더 많은 네트워크 대역폭 및 CPU 자원을 소모한다.

TARP(Ticket-based ARP)는 사용자가 LTA(local ticket agent)로부터 IP 주소와 MAC 주소의 쌍으로 이루어진 서명된 티켓을 발급받아서 ARP 응답메시지에 첨부하는 방식이다[5]. 첨부된 티켓은 서버에서 주소의 유효성을 검증하는 용도로 사용이 된다.

Enhanced ARP는 투표(voting) 기반의 방식으로 ARP 정보를 인근에 위치한 여러 사용자들로부터 모으는 방법을 사용한다[6]. 중복으로 ARP 메시지를 보내기 때문에 네트워크 성능에 영향을 준다.

PARP(Probe-based ARP)는 ARP 응답 메시지를 받으면 메시지 보낸 호스트한테 즉시 ICMP 에코 요청(echo request) 메시지를 보내, 응답여부에 따라 공격자인지를 판단한다[7].

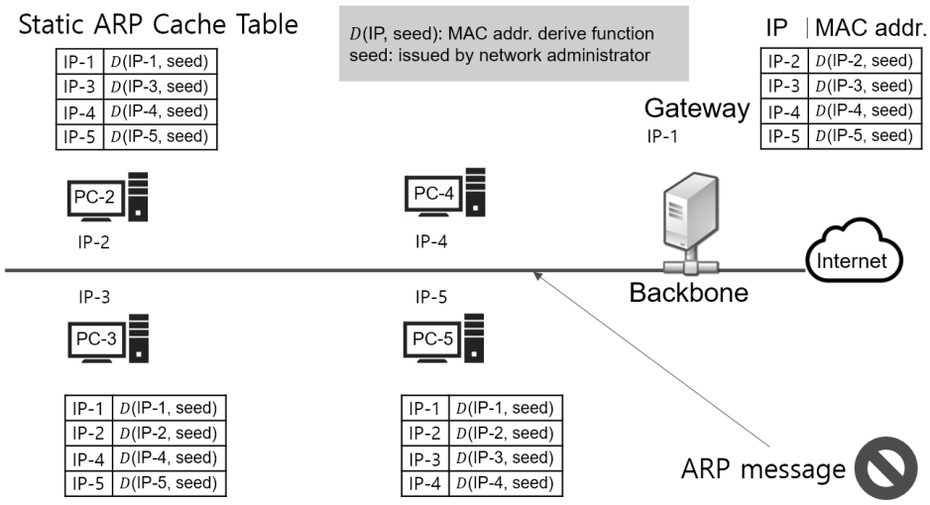


Fig. 2. Architecture of ARP-Disabled Network System

ARP 메시지를 보낸 호스트가 공격자인 경우, 요청에 따라 응답한 ICMP 메시지의 목적지 IP 주소와 자신이 알고 있는 IP 주소가 불일치하기 때문에 응답하지 않는다.

AbdelSalam 등은 정적 ARP 테이블을 사용하는 대신 테이블을 등록, 갱신, 삭제하는 목적으로 키 인증 기반의 메시지를 설계하여 자동으로 ARP 정보를 관리하도록 하였다[8]. 그러나 키가 공격자에게 노출될 경우 메시지의 변조가 가능하다. Srinath 등은 중앙 서버를 이용하여 네트워크에 있는 모든 호스트의 ARP 정보를 수집하여 신뢰 가능한 ARP 테이블을 유지하면서 ARP 기반의 공격을 모니터링 하는 방법을 제안했다[9]. Alharbi 등은 SDN을 이용하여 ARP 공격에 대한 방어 기법을 제시했다[10]. 이 기법은 SDN 스위치에 설치된 SARP NAT를 이용하여 ARP 요청 이력을 기록하고 이에 대한 ARP 응답의 유효성을 검증한 뒤, 중앙 컨트롤러에서 ARP 응답 메시지의 정확성을 보장하는 방식으로 ARP 기반의 공격을 방어한다. Moon 등은 각 사용자 장치에 설치된 에이전트에서 ARP 캐시 테이블에 변화를 모니터링하고 이를 이용하여 ARP 공격을 탐지하는 방법을 제안하였다[11]. 화웨이(Huawei Technologies)는 목적지 노드의 MAC 주소를 매핑 기능을 통해 목적지 노드의 IP 주소를 계산하여 데이터를 전송하는 기술을 소개했다[12]. 이는 네트워크 계층에 대한 설계를 다시 구성해야하기 때문에 많은 구현 비용이 요구된다.

기존의 제안되었던 많은 방법들은 호스트의 동적 구성에 대비하여 메시지 기반의 ARP 테이블을 관리하는 방법론을 채용하고 있거나, 추가적인 장비를 이용하여 메시지를 배제하고 ARP 테이블을 관리하도록 하고 있다. 이 논문에서 제안하는 AAR 네트워크 시스템은 ARP 테이블의 정보를 업데이트하기 위한 메시지를 사용하지 않으면서도 동시에 추가적인 장비를 이용하지 않아 경제성 및 효율성 측면에서 우수하다. 이 논문에서는 기존 연구결과 [13]의 바탕으로 AAR 시스템의 설계를 향상하였으며, 추가 실험 및 분석을 통해 AAR의 성능을 입증하였다.

3. AAR: 주소 자동 결정 네트워크

3.1 ARP 통신 기능 차단

ARP 스푸핑 공격이 가능한 이유는 ARP 응답 메시지 내용의 진위를 검증할 수 없기 때문이다. 이 시스템에서는 ARP 기반의 공격을 원천적으로 방어하기 위해서 ARP의 통신 기능을 차단한다. 여기서 ARP 통신 기능 차단은 ARP 요청, 응답 등 ARP 메시지의 송신 및 수신 기능을 차단한다는 것을 뜻하며, ARP 캐시 테이블은 기존과 동일하게 유지한다.

3.2 정적 ARP 테이블의 관리

ARP 요청, 응답 등 메시지의 통신 기능을 차단하면, ARP 테이블을 업데이트 할 수 없고, 통신하고자 하는 호스트의 물리 주소(MAC 주소)를 알 수 없어, 데이터 링크 계층(즉, MAC 레이어)을 통하여 통신이 불가능하다. 이 시스템에서는 표준 ARP에서 ARP 메시지를 이용하여 동적으로 ARP 테이블을 사용하는 대신, 정적 ARP 테이블을 사용한다.

Fig. 2에서 볼 수 있는 것처럼, PC-2가 게이트웨이와 통신하기 위해서는 게이트웨이의 IP 주소와 MAC 주소의 쌍을 ARP 캐시 테이블에 저장해야 한다. 게이트웨이 또한 PC-2의 IP 주소와 MAC 주소 쌍을 ARP 테이블에 저장해야한다. 이렇게 서로의 정보를 수동으로 ARP 테이블에 저장하면 정상적인 통신이 가능하다. 하지만 모든 디바이스들의 정보(IP 주소 및 MAC 주소 쌍)를 수집하기 위해서는 많은 인력 및 시간 낭비가 생기며, 디바이스의 변동(랜 카드 교체, IP 주소 변경 등)이 있을 때마다 ARP 테이블의 정보를 업데이트해야 하는 문제가 있다.

1) MAC 주소의 유도

AAR 시스템에서는 수동으로 ARP 테이블을 관리하는 대신 주어진 IP 주소로부터 MAC 주소를 유도하여 사용하는

Table 1. Example of Static ARP Table

IP 주소 (reverse hexadecimal)	MAC 주소
192.168.0.1 (0100A8C0)*	39:00:90:7e:9c:ce
192.168.0.2 (0200A8C0)	43:68:02:21:60:f3
192.168.0.3 (0300A8C0)	c6:35:53:e6:7f:cf
⋮	⋮
192.168.0.254 (FE00A8C0)	fe:81:66:13:c1:23

* Subnet gateway

방법을 사용한다. 일반적으로 서브넷의 IP 주소 범위는 서브넷 마스크와 할당된 IP 주소로부터 계산이 가능하다. 예를 들어, IP 주소와 서브넷 마스크가 각각 192.168.0.45/255.255.255.0 라면, 서브넷의 IP 주소 범위는 192.168.0.1~192.168.0.254까지이다. 네트워크 호스트들은, 이 서브넷의 각 IP 주소와 IP 주소로부터 유도한 MAC 주소를 식 (1)과 같이 계산 하여, ARP 테이블에 저장하고, 자신의 NIC (network interface card)의 MAC 주소를 자신의 IP 주소로부터 유도된 MAC 주소로 업데이트하면 서브넷의 모든 호스트들 사이에서 정상적인 통신이 이루어진다.

$$MAC_{Addr} = H(IP_{Addr} \| K) \quad (1)$$

여기서 IP_{Addr}는 IP 주소이고, K는 네트워크 관리자에 의해 발급한 시드(seed) 값으로 서브넷의 호스트들에게 공개된 값이다. $H()$는 SHA-256과 같은 암호학적 해시 함수를 수행하고 MAC_{Addr}의 크기(이더넷의 경우 48-bit)만큼 잘라 출력하는 함수이다. 예를 들어, 자신의 IP_{Addr}가 192.168.0.45이고 K가 '1234567890'일 때, 다음 Table 1과 같이 정적 ARP 테이블을 완성할 수 있다.

2) 셋업 알고리즘

AAR에서의 게이트웨이 및 클라이언트 장치의 셋업 과정(Algorithm. 1)은 (1) ARP 기능 차단, (2) NIC의 물리적 주소 설정, (3) 기존 ARP 캐시 삭제, (4) 정적 ARP 캐시 추가 단계로 이루어진다. 셋업은 새롭게 IP 주소를 할당 받을 때(DHCP를 통한 동적 할당 포함) 동작한다.

a) $TurnOff()$는 ARP 메시지의 전송 및 수신 기능을 차단하는 함수이며 ARP기반의 공격을 무력화한다.
b) 자신에게 할당된 IP_{Addr}와 공개된 시드 값 K를 이용하여 함수 H로 유도한 MAC_{Addr}을 계산한 다음, $SetMAC()$ 함수를 이용하여 자신의 네트워크 인터페이스 카드(NIC)의 물리적 주소를 MAC_{Addr}로 설정한다.
c) $ClearAll()$ 함수는 ARP 테이블(ARP_{table})에 있는 기존 ARP 캐시들을 삭제한다.
d) 자신의 IP_{Addr}와 공개된 $mask$를 이용하여 서브넷에 속한 모든 호스트들의 IP 주소와 서브넷의 크기를 $CalSubNetRange()$함수로 계산하고 이를 각각 배열 S와 변수 t에 저장한다.
e) S에 저장된 모든 IP 주소들은 MAC 주소 유도를 거쳐 (IP, MAC) 쌍의 형태로 ARP_{table}에 추가된다.

Algorithm 1. Setup Procedure

Input : IP_{Addr}, $mask(subnet\ mask)$, K,
$NIC(network\ interface\ card)$, ARP_{table}

```
1: TurnOff(ARP);
2:
3: // modifying my NIC's MAC addr
4: MAC_Addr ← H(IP_Addr ‖ K);
5: SetMAC(NIC, MAC_Addr);
6:
7: ClearAll(ARP_table);
8:
9: // adding IP and MAC pairs to ARP table
10: (S,t) ← CalSubNetRange(IP_Addr, mask);
11: For i=1 to t
12:     ARP_table += (S[i], H(S[i] ‖ K));
```

4. 네트워크 실험

4.1 실험 구성

이 논문에서는 AAR 네트워크 시스템의 보안성과 기본적인 성능을 검증하기 위해서 다양한 실험을 수행하였다. 이 논문에서는 이러한 실험을 위하여 OpenWRT[14] 시스템을 TP-Link AC1750에 탑재하여 유선 기능만을 사용하여 게이트웨이를 구성하였다(Table 2). OpenWRT는 리눅스 기반의 운영체제이며 다양한 리눅스 라이브러리 및 쉘 프로그래밍을 지원한다. 실험을 위해 사용한 쉘 스크립트는 ip[15], ifconfig[16]. arp[17] 등 리눅스 커맨드로 구성하였다. ARP의 통신 기능 차단하기 위해서 ip 커맨드를 사용하였고, NIC의 MAC 주소는 ifconfig 커맨드를 이용하여 변경하였으며, ARP 테이블의 추가 및 삭제는 arp 커맨드로 이루어졌다. MAC 주소를 계산하기 위해 호출하는 프로그램은 C로 작성하였으며, 내부의 해시 함수는

Table 2. Specification of Gateway

	Description
Router	TP-Link AC1750 (ver.2)
OS	OpenWRT (15.05,Chaos Calmer)
kernel	ver.3.18.20
toolkit	ip, ifconfig, arp

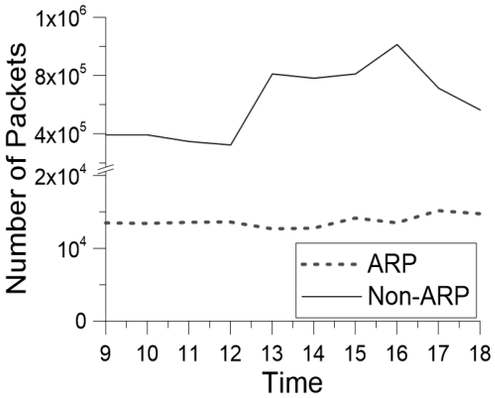


Fig. 3. 9 Hours(9am~6pm) Traffic Data Captured from Gateway of Standard ARP Network

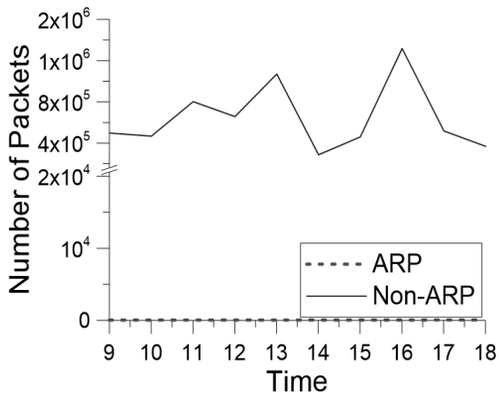


Fig. 4. 9 Hours(9am~6pm) Traffic Data Captured from Gateway of AAR Network

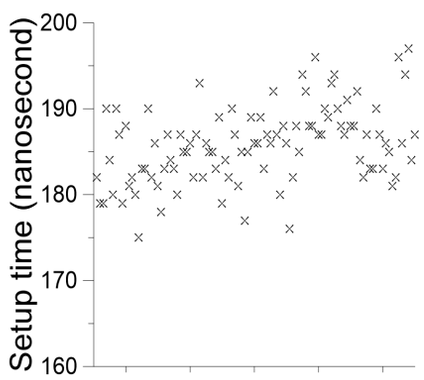


Fig. 5. Setup Time in Client Devices

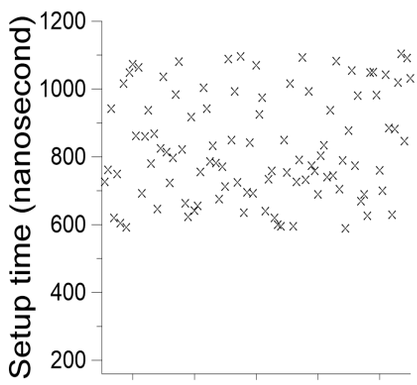


Fig. 6. Setup Time in Gateway

SHA-256을 사용하였다. 해당 쉘 스크립트는 리눅스 시스템의 기본적인 구조를 변경하지 않으며, 20줄 이하로 구성할 수 있었다. 클라이언트 디바이스는 데스크탑 환경에서 구성하였으며 사양은 Table 3과 같다.

이 논문에서는 AAR 네트워크의 ARP 공격에 대한 보안성을 검증하기 위하여, AAR 네트워크와 표준 ARP 네트워크에 ARP 스푸핑 공격을 각각 20번 반복하여 수행하였다. 이 실험에서는 Fig. 2와 같이 네트워크를 구성하고 공격자가 운영체제 Kali Linux(kernel ver.4.3.0)에서 ARPspoof [18]와 sslstrip[19]를 이용하여 게이트웨이의 ARP 캐시를 업데이트하는 공격 수행하였다.

Table 3. Specification of Client Devices

	Description
CPU	Intel Core i7-4770
Memory	32 GB
OS	Ubuntu 14.04
kernel	ver.3.18.0-20-generic

또한 AAR 네트워크의 트래픽 절약 효과를 측정하기 위하여 AAR 네트워크와 표준 ARP 네트워크를 사용할 때, 4개의 사용자 호스트가 존재하는 네트워크에서 게이트웨이를 통과한 모든 패킷을 오전 9시부터 오후 6시까지 분석하였다.

추가적으로 AAR 네트워크를 셋팅하기 위한 시간을 측정하기 위해서 클래스 C 네트워크의 모든 호스트들(254개의 IP 주소)에 해당하는 MAC 주소를 생성하는 수행 시간을 100번 반복하여 측정하였다.

4.2 결과 및 분석

1) 보안성 실험(ARP 스푸핑 공격 결과)

내부 공격자는 제안한 AAR 시스템에서 공개된 K, 함수 $H()$ 또는 유휴 IP 주소를 안다고 가정한다. 제안한 시스템의 게이트웨이에 대해 ARP 스푸핑 공격한 결과 20번의 공격이 모두 실패하였으며 AAR 시스템은 내부 공격자로부터 안전하다는 것을 증명하였다. 반면, 표준 ARP 네트워크에서는 20번의 ARP 스푸핑 공격이 모두 성공하였다.

2) 네트워크 트래픽 절약

AAR 시스템에서는 ARP 메시지 통신 기능을 차단하였기 때문에 네트워크의 통신구간에 ARP 패킷이 존재하지 않는다. Fig. 3에서 알 수 있다시피, 표준 ARP 네트워크에서는 트래픽의 사용량에 상관없이 ARP 패킷은 꾸준히 일정한 비율(약 2.2%)로 네트워크 트래픽을 사용하는 것으로 나타났다. 반면, Fig. 4에서 보다시피, AAR 시스템에서 같은 실험을 진행한 경우 모든 네트워크 상황에서 ARP 패킷이 존재하지 않는다. 따라서 제안한 시스템에서는 정상 네트워크 시스템에서 ARP 패킷이 차지한 트래픽만큼의 절약효과를 얻을 수 있었다.

3) 정적 ARP 테이블을 위한 셋업 비용

기존의 표준 ARP 네트워크는 동적으로 ARP 테이블을 업데이트한다. ARP 테이블의 (IP 주소 변동으로 인한) 유효성을 확인하기 위해서 네트워크의 모든 디바이스는 지속적으로 ARP 메시지를 보내게 된다. 따라서 네트워크 트래픽 뿐만 아니라 CPU 자원의 소모도 지속적으로 발생한다. 반대로 제안한 시스템은 IP 주소로부터 MAC 주소를 유도할 수 있기 때문에 ARP 테이블을 업데이트 할 필요는 없다. 정적 ARP 테이블의 셋업과정은 한 번 수행하지만 CPU 자원을 지속적으로 사용하지 않는다.

Fig. 5는 Table 3에 기술된 것과 같이 상대적으로 일반적인 클라이언트 디바이스에서의 셋업 시간을 보여준다. 평균 셋업시간은 185.7 ns를 기하였다. Fig. 6은 Table 2에 기술된 것과 같이 상대적으로 성능이 낮은 게이트웨이 디바이스에서의 셋업 시간을 보여준다. 평균 829.98 ns로 일반적인 클라이언트 디바이스보다 느린 것을 알 수 있다.

이러한 셋업은 새로운 IP 주소를 할당받거나 시드 값이 변경되었을 때만 발생하므로, DHCP와 같은 프로토콜을 이용하여 지나치게 자주 IP 주소를 갱신하는 경우만 아니라면 셋업 과정 자체가 호스트들에게 있어서 큰 부담은 아니다.

5. 다른 이슈들

5.1 시드 값의 변경

시드 값은 서브넷 네트워크에 공유되는 것을 기본으로 하며, 모든 네트워크에서 동일한 값을 사용하거나 아무런 값도 사용하지 않아도 문제가 발생하지 않는다. 시드 값은 기본적으로 IP 주소나 WLAN의 패스워드처럼 오프라인으로 직접 전달 받는다. 디지털 서명을 포함하여 온라인으로도 전달 받을 수도 있지만, 이를 위해서는 프로토콜을 새롭게 정의해야할 필요가 있다. 시드 값을 특정 서브넷에서만 비밀로 공유하도록 할 경우, 시드 값을 모를 때 통신이 불가능하도록 만들 수 있다. 이러한 특징을 이용하여 브로드캐스트 암호화(broadcast encryption)와 같은 암호학적 도구를 이용하면, 실시간으로 특정 호스트의 통신만을 차단할 수 있는 기능(L2 접근 제어)을 제공할 수 있다. 이는 앞으로의 연구로 남겨둔다.

5.2 서브넷의 크기 및 동적 구성

AAR 네트워크에서 지원할 수 있는 최대 서브넷의 크기는 호스트들이 할당할 수 있는 ARP 테이블의 크기와 일치한다. 서브넷의 크기는 이론적으로 A클래스 서브넷의 경우 16,777,214에 이르고 B클래스 서브넷의 경우 65,534가 된다. 그러나 일반적으로 이러한 규모를 갖는 네트워크는 원활한 통신이 이루어지지 않기 때문에 이보다 더 작은 서브넷으로 나누어 사용하는 것이 일반적이며, 서브넷의 크기가 작을수록 셋업에 걸리는 시간이 더 빠르다.

표준 ARP 네트워크의 경우 구성이 동적으로 변하는 네트워크 환경에 빠르게 동작하기 위해서 잦은 메시지 교환이 이루어진다. 그러나 AAR 네트워크의 경우 새롭게 IP 주소를 할당받은 호스트만이 자신의 NIC의 MAC 주소를 교환하고, 새롭게 시드 값을 할당 받았을 때 ARP 테이블을 갱신하는 것만으로 정상적인 통신을 위한 준비가 끝이 난다. 오히려 ARP 네트워크의 경우 잘못된 정보가 네트워크에 남아 있는 동안 정상적인 통신이 이루어지지 않을 가능성이 존재한다.

6. 결 론

ARP 기반의 공격이 오래된 보안 이슈이지만 관련된 연구는 지속적으로 진행되고 있다[7-11]. 그 이유는 현재까지 제안된 기법들은 구현 비용(Cost)이 높거나 기존 네트워크와의 호환성이 낮기 때문이다.

이 논문에서는 ARP 기반의 공격을 무력화하는 AAR 네트워크 시스템을 제안하였다. 이 시스템에서는 ARP의 통신 기능을 차단하므로, ARP 기반의 공격을 무력화할 수 있지만 정적 ARP 테이블을 사용하여 기존과 같이 정상적인 통신이 이루어진다. 정적 ARP 캐시를 사용하려면 네트워크상에 모든 디바이스의 IP, MAC 주소의 쌍에 대한 정보를 수집해야 되는데, 이 시스템에서는 디바이스의 MAC 주소를 IP주소로부터 유도할 수 있게 설계하여 정보 수집에 의한 인력 낭비 문제를 해결했다. 또한 네트워크의 통신 구간에 ARP 패킷이 존재 하지 않기 때문에 네트워크 트래픽을 절약하는 효과도 얻을 수 있었다. 제안한 AAR 시스템은 추가적인 장비치를 사용하지 않고 기존 네트워크 커널 구조의 변경도 요구되지 않기 때문에 쉽게 구현할 수 있었다.

앞으로 AAR 시스템에 브로드캐스트 암호화를 기반으로 시드 값을 동적으로 관리하도록 하여 L2 접근 제어 기능을 구현하는 것을 목표로 한다.

References

[1] Attack case, Insite, http://www.insight.co.kr/article.php?ArtNo=30180, Oct, 2015.

[2] D. Bruschi, A. Ornaghi, and E. Rosti, "S-ARP: a Secure Address Resolution Protocol," *19th Annual Computer security Application Conference (ACSAC), Las Vegas*, pp. 66-74, 2003.

[3] M. Oh, Y. Kim, S. Hong, and S.D. Cha, "ASA: Agent-based Secure ARP Cache Management," *IET Communications,* Vol.6, No.7, pp.685-693, 2012.

[4] M. G. Gouda and C. Huang, "A secure address Resolution Protocol," *Computer Networks,* Vol 41, No.1. pp.57-71, 2003.

[5] W. Lootah, W. Enck, and P. McDaniel, "TARP: Ticket-Based Address Resolution Protocol," *Computer Networks,* Vol.51, No.15, pp.4322-4337, Oct., 2007.

[6] S. Y. Nam, D. W. Kim, and J. G. Kim, "Enhanced ARP: Preventing ARP Poisoning-Based Man-in-the-Middle Attacks," *IEEE Communications Letters (ICL),* Vol.14, No.2, pp.187-189, 2010.

[7] P. Pandey, "Prevention of ARP spoofing: A Probe Packet based Technique," *Advance Computing Conference (IACC), Ghaziabad,* pp.147-153, 2013.

[8] A. M. Abdelsalam, W. S. Elkilani, and K. M. Amin, "An Automated Approach for Preventing ARP spoofing Attack using Static ARP Entries," *International Journal of Advanced Computer Science and Applocations (IJACSA),* Vol.5, No.1, 2014.

[9] D. Srinath, S. Panimalar, A. J. Simla, and J. Deepa, "Detection and Prevention of ARP spoofing using Centralized Server," *International Journal of Computer Applications,* Vol.113, No.19, Mar., 2015.

[10] T. Alharbi, D. Durando, F. Pakzad, and M. Portmann, "Securing ARP in Software Defined Networks," *41st IEEE Conference on Local Computer Networks,* Dubai, pp. 523-526, 2016.

[11] D. Moon, J. Lee, Y. Jeong, and J. Park, "RTNSS: a Routing Trace-Based Network Security System for Preventing ARP Spoofing Attacks," *The Journal of Supercomputing,* Vol.72, No.5, pp.1740-1756, 2016.

[12] Huawei Technologies Co., Ltd. "Media Access Control Address Resolution Using Internet Protocol Addresses," USA, US20160241471 A1, Aug., 2016.

[13] D. Battulga, R. H. Jang, and D. H. Nyang, "An ARP-Disabled Network System for Neutralizing ARP- Based Attack," *KIPS FALL,* Busan, pp.234-237, 2016.

[14] OpenWRT [Internet], https://openwrt.org/.

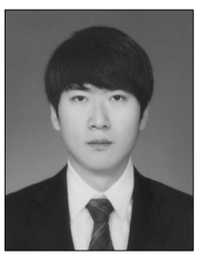
[15] Ip [Internet], https://linux.die.net/man/8/ip.

[16] Ifconfig [Internet], https://linux.die.net/man/8/ifconfig.

[17] Arp [Internet], https://linux.die.net/man/8/arp.

[18] ARPspoof [Internet], http://su2.info/doc/arpspoof.php.

[19] SSLstrip [Internet], http://tools.kali.org/information-gathering/sslstrip.



장 룡 호

e-mail : jiyoo@seclab.inha.ac.kr
2013년 인하대학교 컴퓨터공학부(학사)
2015년 인하대학교 컴퓨터공학부(석사)
2015년~현 재 인하대학교 컴퓨터공학부 박사과정
관심분야 : 네트워크 보안, 무선 인터넷 보안, SDN

이 경 희

e-mail : khlee@suwon.ac.kr
1993년 연세대학교 컴퓨터과학과(학사)
1998년 연세대학교 컴퓨터과학과(석사)
2004년 연세대학교 컴퓨터과학과(박사)
1993년~1996년 LG 소프트(주) 연구원
2000년~2005년 한국전자통신연구원 선임연구원
2005년~현 재 수원대학교 전기공학과 부교수
관심분야 : 바이오인식, 정보보호, 컴퓨터비전, 인공지능, 패턴인식

양 대 헌

e-mail : nyang@inha.ac.kr
1994년 한국과학기술원 과학기술대학 전기 및 전자공학과(학사)
1996년 연세대학교 컴퓨터과학과(석사)
2000년 연세대학교 컴퓨터과학과(박사)
2000년~2003년 한국전자통신연구원 정보보호연구본부 선임연구원
2003년~현 재 인하대학교 컴퓨터정보공학과 교수
관심분야 : 암호이론, 암호프로토콜, 인증프로토콜 무선 인터넷 보안, 네트워크 보안

염 흥 열

e-mail : hyyoum@sch.ac.kr

1981년 한양대학교 전자공학과(학사)

1983년 한양대학교 전자공학과(석사)

1990년 한양대학교 전자공학과(박사)

1982년～1990년 한국전자통신연구소 선임연구원

1990년～현 재 순천향대학교 정보보호학과 정교수

1997년～2000년 순천향대학교 산학연컨소시엄센터 소장

1997년～현 재 한국정보보호학회 총무이사, 학술이사, 교육이사, 논문지편집위원 위원장, 수석부회장(역), 학회장(역), (현) 명예회장

2005년～2008년 ITU-T SG17 Q9 Rapporteur(역)

2006년～2009년 정보통신연구진흥원 정보보호전문위원

2009년～현 재 ITU-T SG17 부의장/SG17 WP2/WP3 의장

관심분야 : 인터넷보안, USN 보안, IPTV 보안, 홈네트워크 보안, 암호 프로토콜