무선랜 환경의 DNS Spoofing 방어 시스템
(DNS Spoofing Protection System for Wi-Fi Networks)

장 대 희 † 박 용 수 ††
(DaeHee Chang) (YongSu Park)

요 약 DNS(Domain Name Service)는 UDP를 사용하는 서버-클라이언트 기반의 질의응답 서비스이다. DNS 위조(Spoofing) 공격은 이러한 DNS 서비스의 중간에서 공격자가 DNS 응답을 위/변조 시키는 것이다. 그 결과 사용자는 공격자가 의도한 악성 사이트로 유도된다. 이러한 공격은 도청에 취약한 무선랜상에서 더욱 위협적이다. 본 논문에서는 공개된 Wi-Fi 무선 환경을 전제로 DNS 위조 공격을 재현한 뒤 무선 AP(Access Point) 단에서 이를 효과적으로 감지하고 방어 할 수 있는 방법을 제안한다. 제안 방법의 효과를 검증하기 위하여 방어시스템을 직접 개발하여 실험 하였고, 그 결과 본 논문에서 제시하는 방법은 네트워크 프로토콜과 클라이언트측 시스템을 수정 할 필요가 없다는 점에서 공개된 무선 네트워크 상의 DNS Spoofing 방어에 효과적임을 확인하였다.

키워드 : 정보보호, 무선랜, 도메인이름, 스푸핑

Abstract DNS(Domain Name Service) is a request/response service based on UDP client-server. DNS spoofing attack is an activity which an attacker manipulates DNS traffic, thus creating a fake/false DNS response. consequently this attack will lead the client into malicious website. This attack is more threatening in wireless network. In this paper we will demonstrate the DNS spoofing attack in public wireless network and propose a defense system which effectively detects and protects the attack in AP(Access Point) level. To verify the effectiveness of our proposal, we have built our own system and conducted some experiments. As the result, we have concluded the proposal of this paper can effectively defend DNS Spoofing from public wireless network in terms of maintaining original network protocol and client system.

Key words : Information Security, DNS, Wireless Network, Domain Name, Spoofing

· 이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(2009-0069740, 2009-0090108)
· 이 논문은 제38회 추계학술발표회에서 '무선랜 환경의 DNS Spoofing 방어 시스템'의 제목으로 발표된 논문을 확장한 것임

† 비 회 원 : 한양대학교 컴퓨터공학부
declspec@naver.com
†† 종신회원 : 한양대학교 컴퓨터공학부 교수
yongsu@hanyang.ac.kr
(Corresponding author임)

논문접수 : 2011년 12월 30일
심사완료 : 2012년 2월 16일

정보과학회논문지: 컴퓨팅의 실제 및 레터 제18권 제5호(2012.5)

1. 서 론

통상 인터넷을 이용하기 위해서는 해당 사이트의 IP 주소를 알아야 하며, 이 주소는 일례로, 166.104.121.1과 같이 외우기 힘들게 되어 있다. 이에 일례로, 'www.yahoo.com'과 같은 기억하기 쉬운 이름 체계를 만들게 되었다(통상 FQDN이라 부른다).

사용자들이 사용하는 FQDN을 실제 인터넷에서 필요한 IP 주소로 변환하는 서비스가 필요하며, 이를 DNS(Domain Name Service)라고 한다. 이는 웹상에서 호스트 컴퓨터가 도메인 이름을 IP주소로 바꾸기 위하여 DNS 서버에게 질의 및 응답을 하는 서비스를 말한다. DNS 위조(Spoofing) 공격은 이 트래픽을 공격자가 가로채고 위조된 IP 주소로 응답하여 호스트 컴퓨터가 위조된 IP 주소로 접근하게끔 하는 공격을 말한다.

일반적으로 공공장소에서의 Wi-Fi는 암호화가 어려우며, 비록 암호화를 한 경우도 암호를 공개하기 때문에 사실상 보안성이 없다고 볼 수 있다. 이에, 대부분의 민감한 네트워크 서비스는 상위계층 프로토콜에서 암호화를 지원하지만 DNS 서비스의 경우 인터넷에서 가장 핵심적인 서비스 중 하나임에도 불구하고 상위계층 프로토콜이 매우 취약하여 공격의 대상이 된다.

본 논문에서는 암호화 되지 않은 Wi-Fi 무선 환경에서 서비스 이용자들의 불편함이 없이 DNS 위조 공격을 효과적으로 감지하고 이를 방어할 수 있는 방법을 제시한다. 우선, 우리는 DNS 위조 공격을 재현하는데 성공하였으며 그 결과를 간단히 서술할 것이다. 이후 공격 방어를 위한 제안 방법을 설명하고, 그 효과를 검증하기 위하여 OpenWrt 임베디드 리눅스 및 Atheros 디바이스 드라이버를 수정하여 제안 방법을 구현한 내용을 설명한다.

2. DNS 위조 공격 개요

사용자가 인터넷 창에 방문하고자 하는 사이트의 FQDN이 담긴 URL을 입력하고, 접속하려고 한다고 가정하자. 그러면, 컴퓨터는 해당 사이트를 방문하기 위하여 IP 주소를 알아야 한다. 이에 브라우저는 캐시가 없

는 경우 1차 DNS 서버에게 URL의 FQDN에 해당하는 IP 주소를 찾아달라고 요청을 하게 되고, 서버는 이 질문에 대한 올바른 IP 주소를 응답하게 된다. 이때 요청/응답 패킷은 UDP서비스를 이용하는데, TCP 와 같이 연결을 맺거나 순서번호 등을 알아 낼 필요가 없으므로 공격자는 손쉽게 중간에 끼어들고 이들 간의 통신을 조작 할 수 있다. 공격이 성공할 경우 웹브라우저에 올바른 URL 주소를 입력해도 잘못된 서버로 접속이 된다.

3. DNS 위조 공격 재현

DNS 위조 공격 방법은 크게 2 가지로 나눌 수 있다. 첫 번째는 공격자가 서버와 클라이언트의 중간에 완전하게 끼어들어서 중간에 데이터를 변조시키는 방법으로서(Man In The Middle), 이는 LAN 상에서 ARP 프로토콜의 취약성에 기반 한 ARP 테이블 변조를 통해서 비교적 쉽게 수행 가능하다.

그림 1이 보여주는 이러한 공격은 'ARP Spoofing'이라는 이름으로 이미 잘 알려져 있으며 IDS를 활용하여 탐지하는 등 여러 가지 대안이 존재한다.

두 번째는 구현하기 어렵지만 그만큼 대응하기도 어려운 방법으로서 Wireless Sniffing에 기반 하는 공격이다. 공격자는 DNS 클라이언트와 동일한 LAN 영역에서 패킷을 수동적으로 도청(Sniffing)하다가 DNS 요청이 일어나는 시점을 탐지하고, 실제 DNS 서버의 응답이 오기 전에 재빠르게 가짜 DNS 응답을 생성하여 공격 대상에게 응답해주는 방법이다. 웹 브라우저는 일단 자신이 요청한 내용에 대한 정당한 응답을 받으면 두 번째로 오는 응답은 무시하게 되는데 이는 DNS 프로토콜의 통신에서 무결성 및 인증과정이 취약한 것을 보여준다(인증 과정이 없는 것은 아니다).

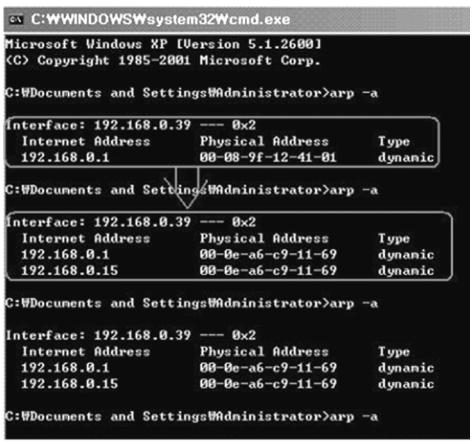

그림 1 ARP Spoofing에 의해 Gateway의 MAC주소가 변조된 모습

본 논문에서는 두 번째 공격 방법(Wireless Sniffing)을 실제로 구현하여 그 위험성을 보이고, 이를 효과적으로 방어하기 위한 대응책을 제시하고자 한다. 우리는 이에 각종 기술문서들을 참고하며[1-3] 독자적으로 'dspoof'라는 리눅스 기반의 해킹 툴을 개발하였고 실제로 iptimes, netgear, dlink등 시중에 판매되는 유무선 공유기 제품을 대상으로 모두 공격시연에 성공하였다.

공격을 하기 위해서는 도청용 무선 NIC를 준비하고 공격 대상이 사용 중인 Wi-Fi 주파수 대역을 알아내어 NIC의 주파수를 맞춘 뒤 promiscuous monitor 모드를 설정해야 한다. 무선 도청중인 NIC 는 패킷송신이 불가능하므로 패킷송신용 NIC 가 추가적으로 필요하다. 즉, dspoof 는 도청용 NIC 와 송신용 NIC 두 개를 요구한다. 공격을 위한 모든 준비 작업이 끝난 뒤에 dspoof를 구동하게 되면 dspoof 는 promiscuous mode로 수신한 트래픽을 실시간 검사하며 DNS 요청이 일어나는 것을 감지한다. 감지한 즉시 최대한 빠르게 위조된 DNS 응답 패킷을 만들고 공격할 호스트에게 전송한다. 공격당한 호스트는 공격자가 보낸 DNS 응답을 실제 DNS 응답이라 믿고 이에 따라 엉뚱한 웹서버로 접속하게 된다.

그림 2는 일반적인 LAN 상황을 간단하게 표현한 것이다. 중앙의 Gateway 로부터 왼쪽은 인터넷 망을 표현한 것이며 오른쪽은 LAN 환경을 표현한 것이다. Gateway 는 유/무선 공유기 또는 라우터라고 가정 하였으며, LAN 상에 Victim 과 Attacker 호스트를 표현하였다. Attacker Host 는 도청용 무선 NIC(wifi0)를 가지고 DNS 요청시점을 탐지하며 대기한다. 아래 그림은 dspoof를 실행하는 모습인데 송신용 인터페이스로 192.168.0.3을 사용하고 DNS 위조 응답용 IP를 202.30.31.52로 설정하여 공격하는 모습이다. 감청용 인터페이스는 따로 지정 가능하지만 지정하지 않는 경우 시스템이 수신하는 모든 패킷을 분석한다. 또한 여기서는 실험을 위한 것이므로 공격대상은 신호 수신범위의 모든 호스트이다. 아래에는 실제로 공격을 수행하는 화면과 해킹 프로그램의 구조, 그리고 공격이 성공한 결과를 보이고 있다. 그림 3에서 상자부분은 dspoof가 192.168.0.7

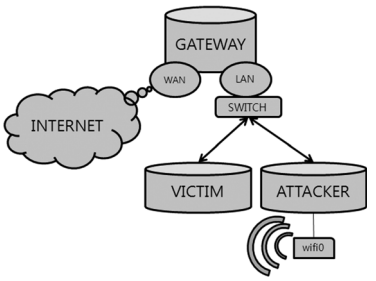

그림 2 네트워크 공격 상황

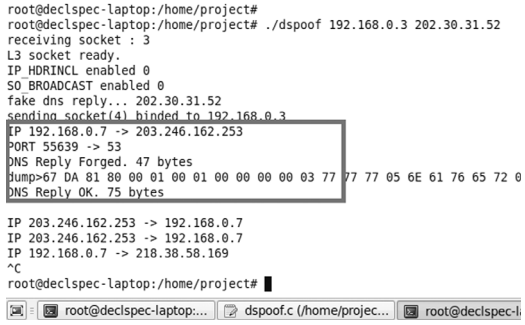

그림 3 DNS Spoofing 수행 화면

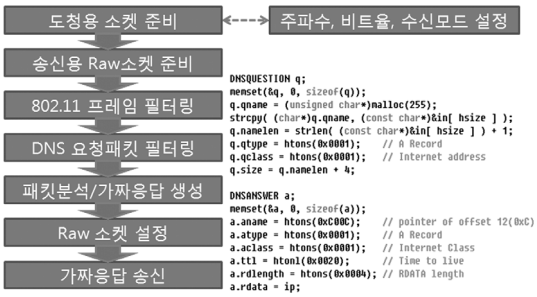

그림 4 dspoof 프로그램의 구조

의 호스트가 203.246.162.253으로 DNS 요청을 하는 것을 탐지한 것에 대한 내용과 함께 자신이 생성한 위조 응답에 대한 dump를 표기한 것이다.

그림 4는 dspoof 프로그램의 흐름도와 C 소스코드의 일부 이다. 프로그램의 구현 방식은 도청된 패킷의 바이트스트림으로부터 DNS 헤더를 검사하고 정보를 추출한 뒤 해당 정보를 기반으로 위조응답을 생성하여 IP 헤더 조작이 가능한 Raw 소켓으로 타겟 호스트에게 송신하는 것이다.

그림 5는 공격당한 호스트의 웹브라우저에 대한 패킷 덤프 내용이다(그림 3에 대응되는 것은 아니다). 여기서 '5C 83'이라는 것은 DNS 프로토콜의 16비트 ID번호인

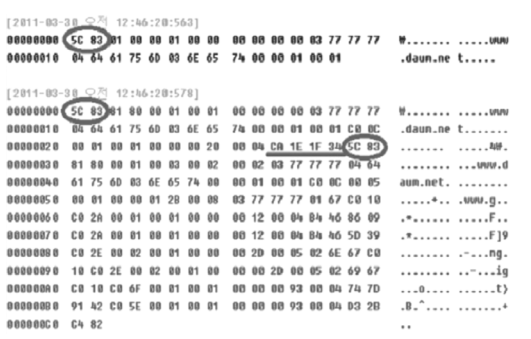

그림 5 공격당한 호스트의 패킷덤프

표 1 DNS 위조 공격 실험 결과

스니핑 인터페이스	1차 DNS 서버 위치	채널 사용률(KB/S)	공격 성공률
Ralink 2870	WAN (8.8.8.8)	10	90% 이상
		100	50%
		500	2~30%
	AS내부	10	50%
		100	2~30%
		500	10%이하
Ralink 2860	WAN (8.8.8.8)	10	10%이하
		100	0%
		500	0%
	AS내부	10	10%이하
		100	0%
		500	0%

데 이것이 응답에서 2번 나타난 것을 볼 수 있다. 즉, 2개의 응답이 왔으며 첫 번째 응답이 공격자가 위조한 내용이고 두 번째 응답이 실제 DNS 서버의 응답이다. 두 번째 응답패킷이 나타나기 바로 전 붉은 색으로 밑줄 친 부분에 'CA 1E 1F 34'라는 HEX코드가 보이는데, 이것은 그림 3에서 가짜 응답용 IP로 설정했던 '202.30.31.52'의 16진수 코드이다. 공격자의 응답이 실제 응답보다 빨랐기 때문에 공격이 성공했다. 이렇게 위조된 DNS 응답을 받은 호스트의 웹브라우저는 잘못된 웹서버로 접속 하게 된다.

표 1은 이러한 DNS 위조 공격을 수행한 실험들에 대한 결과로서 이것은 OpenWrt가 설치된 Buffalo 유무선 공유기 상에서 54Mbps의 802.11g 모드로 실험한 것이다. 스니핑 인터페이스로 Ralink 2870 칩셋의 USB형 랜카드와 Ralink 2860 칩셋의 내장형 랜카드를 사용하였다. 표 1에서 Ralink 2860 칩셋을 사용하는 경우 공격 성공률이 매우 낮은데 이것은 NIC의 패킷 처리속도가 느리기 때문이다. 1차 DNS 서버의 위치는 네트워크상에서 멀수록 공격 성공률이 높았다(응답 시간이 느리므로). 표 1에서의 '8.8.8.8'은 Google public DNS 서버의 IP로서 네트워크상 멀리 떨어져 있다. AS 내부의 경우는 서버가 게이트웨이와 1~2홉 이내로 떨어진 위치에 존재하는 경우로서 상대적으로 가깝다. 또한 공격자가 혼잡한 무선 채널에 있는 경우 감청이 어렵고 느려지기 때문에 공격 성공률과 채널 사용률은 반비례 하는 것을 볼 수 있다.

4. DNS 위조 공격 방어 시스템 설계

본 논문에서는 앞에서 살펴본 공격에 대한 방어 시스템을 제시함에 있어서 기존 시스템의 변경을 최소화 하는 것을 높은 우선순위로 고려하였다. 이를 위해서 방어 시스템은 DNS 서비스를 받는 호스트에 보안 모듈을 탑

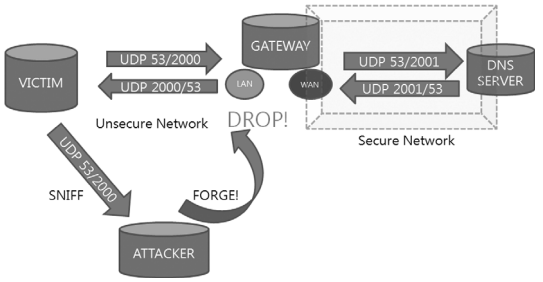

그림 6 Transparent Proxy Module 모식도

재하지 않고 네트워크 게이트웨이 커널 상에 보안 모듈을 탑재 한다(PC는 일체 변경될 일이 없다).

많은 연구 끝에 우리는 Transparent Proxy 라는 개념을 적용 하였다. 이는 네트워크 서비스의 양쪽 단말에게 투명하도록 중간에 특정 서비스를 적용하는 개념이다. 방어 시스템의 개괄적인 구조는 그림 6과 같다.

방어 모듈은 오가는 DNS 패킷의 신원에 대한 정보 중 서비스 이용에 영향을 주지 않는 부분을 변조 및 복원 시키는데, 트랜스포트 계층의 근원지 포트번호를 이용한다. 예를 들어 피해자 호스트는 53번(DNS) 포트를 목적지로 하는 UDP 패킷을 AP를 거쳐 DNS 서버로 송신한다(AP 는 반드시 거친다).

근원지 포트번호는 OS에 의해서 랜덤으로 생성되는데 이것이 2000 이라고 하자. 방어 모듈은 게이트웨이의 네트워크 스택에 후킹을 걸고 UDP 53번을 향하는 패킷의 근원지 포트번호를 변경(그림 6의 예에서는 2001로) 시킨다. 그와 동시에 커널 모듈이 관리하는 테이블 상에 이러한 변조에 대한 기록을 남기고 원래의 근원지 포트를 목적지로 하는 UDP 패킷을 차단한다. 얼마 후 DNS 서버에서 변조했던 2001번 포트를 목적지로 UDP 응답이 오면 방어 모듈은 이것을 다시 2000으로 복원시켜 준 뒤에 UDP 2000에 대한 차단을 해제하고 테이블상의 기록을 삭제한다.

공격자가 위조한 응답패킷에서의 목적지 포트번호는 무선랜 감청을 통해 얻은 정보에 기반 한 것이기 때문에 Transparent Proxy에 의해 변조되기 전의 정보이다. 따라서 공격자가 스니핑을 통해 변조되기 이전의 목적지 포트 번호를 가진 위조패킷을 생성하여 전송하게 되면 AP 상의 방어모듈이 이를 탐지하고 차단할 수 있게 된다. 이런 방식의 신원정보 변조는 brute-force 공격에(공격자가 변조된 값을 무차별 추측하는 경우) 대해서 더욱 견고하게 만들기 위해, DNS 식별ID에 대해서도 추가적으로 수행 할 수 있으나 방어 시스템이 DNS 패킷 포맷에 의존하게 되는 점과 처리 오버헤드가 증가한다는 단점도 존재한다.

혹자는 여기서 '공격자가 게이트웨이를 경유하지 않고 직접 802.11 frame을 공격대상의 NIC로 주입하면 방어 모듈을 우회하지 않을까?'라는 의문을 가질 수 있을 것이다. 그러나 이것은 매우 어렵다. 802.11 프로토콜 스펙에 따르면 무선랜 단말과 AP가 주고받는 모든 프레임에는 AID 라는 세션ID와 순서번호가 할당되며 이는 NIC의 Firmware상에서 처리된다. AP의 도움 없이 직접 무선 프레임을 주입 시키려면 공격자는 NIC를 완벽하게 제어 할 수 있어야 하며 빠른 속도로 변하는 순서번호를 올바르게 맞출 수 있어야 하므로 현실적으로 어렵다. 한 가지 문제는 네트워크 내부 단말끼리 통신할 때 일반적으로 송신자가 ARP를 통해 직접 링크계층 목적지 주소(여기서는 MAC주소)를 얻고, 패킷을 보낸다는 것이다[4]. 스위치는 이러한 프레임에 대해서 러닝을 한 뒤에 상위계층으로 넘기지 않고 직접 전달한다. 이는 무선랜 상에서도 동일한데, AP의 디바이스 드라이버 레벨(또는 그 바로 상위 레벨)에서 이 기능을 대행한다. 우리가 제시하는 방어 모듈은 게이트웨이의 커널 상에 위치하므로 방어 모듈구성을 위해서는 링크계층 스위치를 수정해야 하며, 이것은 방어 시스템의 추가로 인한 전체 시스템 성능의 저하를 유발한다. 이러한 성능저하를 개선할 수 있는 방법은 고민해봐야 할 문제이다.

결론적으로 지금까지의 내용처럼 트랜스포트 계층 정보조작에 기반 하여 방어 모듈을 구현하는 방식은 IP Spoofing을 차단하는 것보다 효과적이다. IP Spoofing의 경우 정당한 시스템이 필요로 할 수도 있고(Mobile IP등) 공격자가 네트워크 외부를 경유하여 패킷을 보내면 차단하기 어려운 점이 있으며 IPv6로의 전환 시 시스템이 바뀌어야 하는 문제가 있기 때문이다.

5. 방어 시스템 구현 및 성능분석 결과

본 논문에서 제시하고자 하는 방어 시스템을 구현하기 위해서는 먼저 링크계층의 스위칭방식을 제어 할 수 있어야 하고 네트워크 체인에 후킹을 걸어서 상태기반 방화벽을 구현해야 한다. 본 논문의 실험에 사용된 무선 AP는 오픈소스 무선라우터 운영체제인 OpenWrt 가 포팅된 Buffalo WZR-HP-G300NH-AP 모델로서 Atheros 사의 NIC가 내장되어 있고 802.11 프레임의 스위칭은 NIC 디바이스 드라이버가 담당한다. 따라서 우리는 공개소스인 Atheros NIC 의 디바이스 드라이버 코드를 분석하고 기술문서들을 참고하며 해당 디바이스 드라이버의 코드를 수정하여 모든 프레임을 방어모듈이 검사할 수 있도록 하였다[5,6]. 방어 모듈 구현에서 기존의 유명한 리눅스기반 방화벽인 'iptables'가 존재하지만 이것은 상태기반으로 동작 할 수 없는 문제가 있으므로 사용할 수 없다. 본 논문의 실험에서는 리눅스기반의 방화벽 개발용 인터페이스로 유명한 'netfilter'를 사용하여

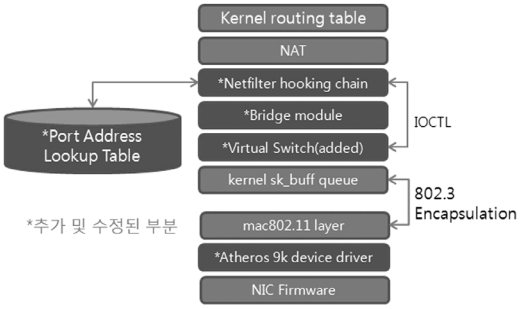

그림 7 방어 시스템이 추가된 OpenWrt 커널

위에서 언급한 내용을 구현하는 상태기반 방화벽을 개발 하였다.

그림 7은 이러한 기능이 추가되어 수정된 OpenWrt 커널을 표현한 것이다. 수정된 Atheros 디바이스 드라이버는 LAN 내부 스위칭을 하지 않고 모든 패킷을 커널 버퍼로 올린다. 커널로 올라온 패킷은 netfilter 후킹체인에 등록된 방어모듈의 콜백함수를 통해 전달된다. 방어 모듈은 DNS 패킷을 필터링한 뒤 내장된 Port Address Lookup Table을 참조하며 패킷을 변/복조 한다. Virtual Switch 모듈은 LAN 내부를 향하는 패킷을 ioctl 을 통해 방어 모듈로부터 검사를 받고 스위칭 해 준다. 일반적인 라우터라면 모든 패킷을 라우팅 해줘야 하지만 실험에 사용된 공유기의 커널은 우리가 임의로 수정한 사설 네트워크간의 패킷흐름을 라우팅 해주지 못하였기에 따로 구현 하였다. 그림에서 ioctl을 통해 2중으로 방어 모듈에게 검사를 받는 이유는 Virtual Switch 가 사용하는 SOCK_PACKET 소켓이 netfilter를 거치기 이전의 패킷 사본을 받기 때문이다(실험을 통해 확인함). 마지막으로 Bridge module은 이더넷과 무선링크간의 연결을 차단하도록 수정되었다(본 논문에서는 무선랜만 사용하므로). 보안 모듈은 커널모듈로서 'insmod' 명령으로 등록되며 상태기반 방화벽으로 동작한다.

우리는 방어 시스템이 기존 시스템의 성능에 얼마나 차이를 주는지에 대한 실험을 하기 위해서 ICMP Request 응답모듈을 구현하여 네트워크 스택상의 여러 군데에 위치시키고 ping 으로 실험하였다. 표 2는 이 실험에 대한 결과이다.

4가지 실험은 동일한 이더넷 환경에서 수행 되었다. 실험 결과에 따르면 패킷처리 속도가 방화벽에 의해 0.7ms 정도 지연되는 것을 알 수 있다. 가상 스위치의 경우 DNS 트래픽만 보안모듈을 통과시키고 나머지는 LLC 계층으로 직접 스위칭 하는데, 커널/유저 메모리전환 때문에 디바이스 드라이버상의 스위칭보다 1.3ms 정도 시간을 소요함을 알 수 있다. 그러나 가상스위치가 사용되는 상황(LAN 내부 단말간의 통신)은 상대적으로

표 2 보안모듈 성능실험

ICMP 응답 위치	RTT(ms) min/avg/max/mdev
디바이스 드라이버	0.141/0.145/0.155/0.005
커널 IP 레이어	0.277/0.303/0.391/0.028
보안모듈을 거친 커널	1.066/1.084/1.210/0.056
가상스위치	1.296/1.454/6.364/0.809

트래픽이 적기 때문에 큰 문제가 되지 않을 것이다. 방어 시스템을 갖춘 뒤 DNS 위조 공격을 앞전과 동일하게 다시 수행한 결과 DHCP 네트워크 참여부터 LAN 내부 통신 및 정상적인 인터넷 사용에 아무 지장 없이 모든 위조응답이 차단되는 것을 확인 하였다.

6. 결 론

본 논문에서는 Wi-Fi 무선 환경의 패킷 스니핑에 기반 한 DNS 위조 공격이 위협적임을 보이고 이를 효과적으로 감지하여 방어할 수 있는 방법을 제시하였다. 우리는 직접 제작한 해킹 툴을 사용하여 Wi-Fi 환경에서 DNS 위조 공격을 구현 하였고 방어 시스템에 대한 제안 방법의 효과를 검증하기 위하여 OpenWrt 임베디드 리눅스 및 Atheros 디바이스 드라이버를 수정하고 시스템을 구축한 뒤 여러 가지 환경에서 실험하고 성공적으로 공격이 차단되는 것을 확인 하였다.

결과적으로 본 논문에서 제시하는 방법을 사용할 경우 표 2의 실험결과가 보여주듯이 패킷 처리속도를 다소 희생한다는 비용이 발생 하지만, 무선랜 환경에서의 공격이 방어하기 어려우며 위협적이라는 점을 감안하면 본 논문에서 제시한 방어 시스템은 기존의 DNS 시스템을 수정할 필요 없이 상대적으로 적은 비용으로 효과적인 보안성을 제공 할 것으로 기대한다.

참 고 문 헌

[1] Pablo Brenner, "A Technical Tutorial on the IEEE 802.11 Protocol," "BreezeCom," 1997.
[2] Laselva Daniela, "The IEEE 802.11 Medium Access Control (MAC)," 2004.
[3] Wireless LAN Medium Access Control (MAC) and Physical Layer(PHY) Specifications, "IEEE Computer Society Sponsored by the LAN/MAN Standard Committee," pp.150-151, 2007.
[4] James F. Kurose, "Computer Networking A Top-Down Approach," "Pearson," pp.488-510, 2008.
[5] Ankit Jain, "Linux Networking Subsystem -2.4.18," 2002.
[6] Rami Rosen, "Wireless Linux Kernel Networking -advanced topics," "Haifux," 2009.