http://dx.doi.org/10.5392/JKCA.2016.16.10.467

SOHO환경을 위한 스마트 무선 침입 탐지 시스템 구현

Smart Wireless Intrusion Detection System Implementation for SOHO Environment

김철홍, 정임영
경북대학교 전자공학부

Cheol-Hong Kim(kikiyao@knu.ac.kr), Im Y. Jung(iyjung@ee.knu.ac.kr)

요약

정보통신기술의 발달을 기반으로 창업을 하는 영세 업체와 가택 근무 환경(Small Office Home Office: SOHO)이 늘고 있다. SOHO는 대부분 인터넷을 통해 업무를 처리하고 Wi-Fi의 이용이 일반화되어 있다. 무선통신환경은 쉽게 서비스거부(Denial of Service: DoS) 공격으로 통신에 장애가 생겨 업무환경 및 고객 서비스에 영향을 받을 수 있다. 이를 방어하기 위한 방안으로 무선침입탐지시스템(Wireless Intrusion Detection System: WIDS)이 있다. 그러나 시중 WIDS는 구축비용이 비싸며, 관리 부담이 높아 자본력과 인력이 낮은 SOHO환경에서 사용하기에 어려움이 따른다.

본 논문에서는 관리운영의 부담을 줄이고 간단히 공격 위협을 인식할 수 있는 SOHO 환경을 위한 Smart WIDS를 제안하고 구현하였다. Raspberry Pi2를 Smart WIDS의 연산장치로 사용하고, 안드로이드 스마트폰을 Smart WIDS의 연계 공격감지 알림 인터페이스로 제공한다. Smart WIDS는 IEEE 802.11 기반의 Masquerading DoS 와 Resource Depletion DoS을 감지하여, WI-FI의 Pre-shared Key(PSK) 획득과 중간자 공격 시도나 서비스 제공을 방해하는 공격을 감지할 수 있다. 또 안드로이드 스마트폰은 해당 WI-FI의 감지된 공격을 관리자에게 알리며, 관리의 편의성을 제공한다.

■ 중심어 : | SOHO Environment | DoS Attack | WIDS | IEEE 802.11 |

Abstract

With the development of information technology, Small office Home office(SOHO) is picking up. SOHO generally uses Wi-Fi. The wireless LAN environment using 802.11 protocol is easily affected by DoS attacks. To deal with these threats, there is Wireless Intrusion Detection System(WIDS). However, legacy products of WIDS cannot be easily used by SOHO because they are expensive and require management burden.

In this paper, Smart WIDS for SOHO is proposed and implemented on Raspberry Pi2. And, it provides the interface for attack detection notice to android smart phone. Smart WIDS detects Masquerading DoS and Resource Depletion DoS based on IEEE 802.11 so that we notice the attempt of cracking Pre-shared Key(PSK), Man-In-The-Middle(MITM), and service failure.

■ keyword : | SOHO | DoS | WIDS | IEEE 802.11 |

접수일자 : 2016년 07월 04일
수정일자 : 2016년 08월 08일
심사완료일 : 2016년 08월 28일
교신저자 : 정임영, e-mail : iyjung@ee.knu.ac.kr

I. 서 론

IT 발전에 따라, 편의상 WI-FI를 이용하는 환경이 증가 하고 있다. Small office/Small Home(SOHO)환경인 카페와 식당, 벤처 등에서도 WI-FI를 이용하여, 고객에게 서비스를 제공하거나 사용하고 있다. WLAN 환경에서 사생활 정보, 결제 정보, 회사의 기밀 정보들이 노출될 수 있다[1].

802.11 프레임 영역은 암호화가 되어 있지 않아, 공격자는 WLAN 환경에서 물리적으로 근거리에 위치하여, 802.11 프로토콜을 이용한 DoS 공격을 수행할 수 있다. 이 공격으로 WI-FI의 자원을 고갈시켜 원활한 서비스 제공을 방해할 수 있으며, WI-FI와 WI-FI의 서비스를 제공 받는 디바이스인 station간 통신을 차단시킬 수 있다. 더 나아가 WI-FI의 Pre-shared Key(PSK)획득을 위해서 공격을 할 수 있다. 이를 이용해 SOHO환경의 WI-FI를 이용하는 장치들에게 중간자 공격을 수행하여 데이터 스니핑, 위변조가 가능하다. 이로 인해 자산적 혹은 개인정보유출의 피해가 할 수 있다. 이와 같은 무선 공격을 방지하기 위한 시스템으로 Wireless Intrusion Detection System(WIDS)가 있다. WIDS는 무선 상의 공격을 탐지하여, 관리자에게 위협을 알려주는 시스템이다. 하지만 이와 같은 시스템은 비용이 비싸며, 지속적인 모니터링과 관리가 필요하기에 영세 업체에서 사용하기 어렵다[2].

한편, 상용 WIDS에 비하여 가볍고, 위협에 대해 즉각적인 알림을 제공하는 연구가 진행되었다[3-6]. 이와 같은 연구는 WIDS의 운영을 위해, 노트북과 비슷한 사양 기기의 지속적인 운영이 요구된다. 또, 공격 알림 방식에 추가 자원을 요구하는 방식이며, 관리자의 부재나 지속적이지 못한 모니터링에 대한 편의를 제공하지 못하고 있다. SOHO환경에서는 WIDS의 전담 모니터링을 위한 추가적인 인원을 배치하기 어려운 단점이 있다.

본 논문은, SOHO 환경에서는 WIDS의 구축비용을 보다 최소화할 수 있으며, 관리자의 외출 후 복귀와 같은 상황에서의 즉각적인 공격 알림과 관리의 편의를 제공하는 Smart WIDS를 구현하였다. Raspberry Pi2를 WIDS의 하드웨어로 사용하고, 안드로이드 스마트폰을 WIDS의 알림 인터페이스로 구현하였다. 즉, Smart WIDS와 안드로이드 스마트폰 간 블루투스 연결을 통하여 WIDS의 공격 탐지 내용은 안드로이드 폰으로 즉각 알릴 수 있다. 또 관리자의 안드로이드 스마트폰은 WI-FI의 SSID를 인식하여, 관리자의 복귀 상황 시에 WIDS와 자동적으로 연결한다.

그리고 Smart WIDS가 감지하는 공격은 IEEE 802.11 기반의 근거리 무선 공격인 Masquerading DoS 와 Resource Depletion DoS, Beacon flooding 이다. 이는 PSK 추출과 twin devil 공격으로 인한 중간자 공격에 대한 공격 시행 유무, 악의적인 통신 장애를 파악할 수 있다는 의미가 있다.

논문의 구성은 다음과 같다. 2장에서는 무선 네트워크 프로토콜에 유효한 공격법과 경량화 WIDS의 연구 상황을 분석하고 소개한다. 3장에서는 Smart WIDS을 설명하고, 4장에서는 Smart WIDS의 구현, 5장에서는 실험을 보인다. 6장에서는 결론과 앞으로 진행해야할 차후 과제를 기술한다.

2. 관련 연구

2.1 IEEE 802.11의 DoS 공격

WI-FI는 근거리 무선인터넷으로, IEEE에서 제정한 무선표준으로 IEEE802.11로 표기되는 WI-FI와 station 간의 무선 통신 규격이다. 이 프로토콜에서 정의 하고 있는 MAC 계층에서는 무선 환경에서 송수신할 데이터의 관리 정보를 제어한다. 이 MAC Frame Header에서 WI-FI 결합요청, 결합해제 설정을 할 수 있다. 이 부분은 사용자의 데이터 영역이 아니므로 암호화되지 않은 영역이다[5]. 따라서 공격자는 Wi-FI에서 근거리 통신이 가능한 공간에서 이 Frame Header를 조작하여 DoS 공격을 수행할 수 있다. DoS 공격은 크게 Masquerading DoS 와 Resource Depletion DoS, Media Access DoS로 구분된다. Masquerading DoS인 Deauthentication /Disassociation 공격은 MAC spoofing 공격, MITM 공격, WPA decryption 공격을 목적으로 Masquerading 공격을 수행할 수 있다[7]. 또, 개인 SOHO 환경에서는

키 설립을 위해 4-way handshake을 수행하게 된다[8]. 이 때, 공격자는 4-way handshake를 스니핑하여 SSID, station nonce, WI-FI nonce, WI-FI의 mac address, station의 mac address를 획득할 수 있다. 공격자는 이 값을 이용하여 dictionary attack 이나 rainbow table attack을 통해 PSK를 얻을 수 있다. 이 공격은 전문적인 지식 없이, aircrack-ng tool을 이용하여 쉽게 공격이 가능하다[9]. Resource Depletion DoS 공격은 WI-FI의 메모리나 프로세스와 같은 자원을 과사용케 하여, 정당한 사용자에게 서비스를 제공하는 것을 방해한다. 이 공격은 Probe request flood와 Authentication request flood, Association request flood등이 있다. 이런 공격을 이용하여 WI-FI의 자원고갈을 유도할 수 있다. 또 Media Access attack은 무선 매체를 공유하는 방식으로 사용되는 Distributed Coordination Function(DCF)에 대한 공격을 수행할 수 있다.

Smart WIDS는 위의 공격 중 Masquerading DoS 인 Deauthentication/Disassociation 와 Resource Depletion DoS 인 Prober request flood와 Authentication request flood, Reassociation Request flood, Association Request, EAPOL flood외에 추가로 Beacon flood 를 감지할 수 있다. 위 공격은 802.11영역에서 공격 툴을 이용하여 쉽게 실행할 수 있다. 또한 이 공격 자체로 서비스 장애를 초래할 수 있을 뿐만 아니라 PSK 획득이나 중간자 공격을 위해서 활용 될 수 있는 만큼 치명적일 수 있다. 그러므로 SOHO환경에서 대처가 가능해야 된다.

2.2 twin evil attack

공격자는 공격 대상인 WI-FI의 이름(SSID)과 동일한 WI-FI를 만든다. 공격자의 WI-FI는 Beacon의 신호강도를 높여 기존의 station이 오결합하기를 유도한다. 이와 같은 경우는 공격자가 공격 대상 WI-FI의 PSK를 알고 있는 경우나 공격대상이 되는 WI-FI가 암호화를 하지 않는 환경에 적용가능하다. 특히 카페, 공항, 식당과 같은 SOHO환경에 유효하다.

이 공격은 Deauthentication을 통한 PSK 획득 이후에 추가적으로 실행될 수 있다. 또는 공격자는 Rogue WI-FI를 강한 신호로 동작시키고, Deauthentication 공격을 통해 직접적으로 Rogue WI-FI로 station을 유도할 수 있다. 이를 통해 공격자는 Rogue WI-FI와 연결된 station의 정보를 획득하거나 위변조 할 수 있다.

이 공격은 802.11영역에서 Beacon flooding 과 Deauthentication/Disassociation 공격을 통해서 수행될 수 있다. 이는 Smart WIDS 의 탐지 대상에 속하므로 방비가 가능하다.

2.3 기존 경량화 WIDS

높은 가격과 관리의 문제를 가지고 있는 상용 WIDS에 대한 해결 방안으로 경량화 WIDS 연구가 있다. 노트북을 활용하여 무선침입 탐지를 수행하는 WIDS가 있고, Deauthentication attack 만을 탐지하여 경량화를 추구하는 WIDS가 있다[3-6]. 이런 WIDS들은 언제 발생 할지 모르는 공격에 대비하기 위해, 지속적으로 노트북을 해당 영역에 운영하여야 된다. 또한, 경량화 된 연구를 제안하고 있으나 노트북외의 더 낮은 성능에서의 운영이 드러나 있지 않다. SOHO 환경에서는 DoS 탐지에 장애가 없는 하드웨어에서 운영되는 것이 효율적이다. Raspberry Pi2 위에 IDS를 수행하기 위해 Snort를 운영하는 경우, 메모리 제한으로 Snort-rule를 사용하는 것에 한계가 있고 현저한 성능 저하가 생긴다[10]. 이런 문제를 해결하기 위해 저사양의 OS인 Arch Linux ARM을 운영하여 Snort를 동작케 하였다[10]. 또한, WIDS 자체의 DoS 공격에 초점이 맞춰져 있지 않고 유선 환경에서의 IDS 에 집중하고 있다. 또, 기존연구에서는 경고알람방식으로 사용자에게 상용 통신망을 이용한 SMS를 사용하여 위험을 알리는 방법이 있다[7]. 이는 SMS 전달을 위한 추가적인 자원이 요구된다.

3. Smart WIDS for SOHO

Smart WIDS IEEE 802.11에 기반의 공격을 탐지한다. 즉, fake WI-FI의 Beacon flooding과 Masquerading, Resource Depletion의 DoS공격을 감지한다. 이를 통해 Deauthentication/Deauthentication DoS 공격의 WI-FI와 station간 통신 차단을 파악 할 수 있으며, PSK 획득을 위한 공격자의 시도 여부를 유추 할 수 있다. 또 fake

WI-FI의 beacon flooding로 오결합 공격유무와 Resource Depletion DoS 공격으로 인한 자원 고갈 위협을 알게 된다.

Smart WIDS는 안드로이드 스마트폰의 알림 인터페이스로 블루투스 통신을 이용하므로 추가적인 입출력 장치나 상용데이터 망의 접속을 가지지 않는다. 또 Smart WIDS의 탐지 및 연산부를 Raspberry Pi2로 활용함으로서 가격이 저렴하며, 공격탐지만을 위해 운영하기에 용이하다.

즉, aircrack-NG, mdk3와 같은 공격 툴로 IEEE 802.11 프로토콜을 이용한 공격에 대비하기에 비교적 다른 복잡한 탐지 알고리즘 없이도 실질적으로 저사양의 하드웨어에 적합하다.

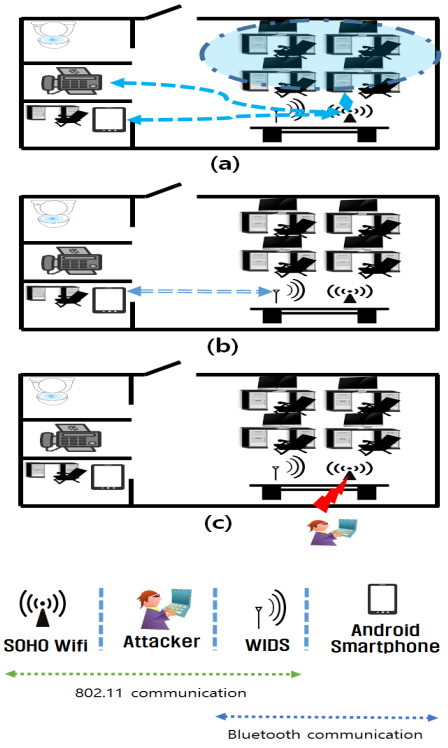

그림 1. Smart WIDS

Smart WIDS 환경 구성은 [그림 1] (a)같이 SOHO환경에서 동작하는 WI-FI와 station으로 구성되어 통신을 수행한다. 또한 관리자의 스마트폰은 WI-FI와 연결하여 제한 없이 인터넷을 활용할 수 있다. (b)와 같이 관리자의 안드로이드 스마트폰과 WIDS는 공격 탐지 알람이나 설정정보를 블루투스를 통해 송수신된다.

만약 (c) 와 같이 공격자의 WI-FI 공격을 Smart WIDS가 탐지한 경우, Smart WIDS는 등록된 블루투스 디바이스에 연결을 시도한다. 해당 관리자가 블루투스 통신 영역에 속할 경우, 관리자는 안드로이드 스마트폰으로 공격 메시지를 즉각 수신하게 된다. 이를 통해 관리자가 공격에 대응할 수 있도록 유도한다.

Smart WIDS와 안드로이드 스마트폰간의 인식 방식으로 안드로이드 스마트폰에서 WI-FI의 beacon인식 방법을 사용한다.

안드로이드 스마트폰에서는 감시 대상이 되는 SOHO WI-FI의 beacon을 수신시에 블루투스 로 연결을 시도한다. 이에 Smart WIDS는 안드로이드 스마트폰의 블루투스 통신이 수신 되면, 안드로이드 스마트폰이 통신영역 내에 있음을 알게 된다.

그리고 Smart WIDS는 WI-FI와 무선 연결을 가지지 않고 WI-FI와 station간의 패킷을 스니핑만 수행한다. 즉, Smart WIDS와 독립적으로 운영되어, IEEE 802.11 DoS 공격 대상이 되지 않는다. 이는 Smart WIDS가 WI-FI와 연결됨으로서 발생할 수 있는 취약점을 방지할 수 있다. 즉 , Smart WIDS 자체에 DoS 공격과 또한 공격자의 공격이 성공한 후, Smart WIDS 가 악성 WI-FI에 연결되는 것을 회피할 수 있다.

3.1 Smart WIDS의 공격 탐지

Smart WIDS의 공격 탐지영역은 802.11 프로토콜 영역에 속하며, WI-FI의 비밀번호(PSK)를 알지 못하는 공격자에 대한 DoS 공격을 탐지한다.

즉, WI-FI의 비밀번호를 모르는 공격자가 WI-FI의 802.11 프로토콜을 수행할 수 있는 통신 영역에 속하여 시행하는 DoS 공격이 탐지대상이 된다.

따라서, 802.11 프로토콜의 MAC 프레임의 Type, subType ToDs, FromDs의 비트를 확인한다. 특정 프레임이 일정 단위시간당 threshold값을 초과시 해당 DoS 공격으로 판별한다.

판별 가능한 DoS 공격은 Masquerading DoS 인 Deauthentication/Disassociation 과 Resource Flooding

DoS 공격인 Prober request flood와 Authentication request flood, Reassociation Request flood, Association Request, EAPOL flood가 된다.

WI-FI 모듈을 사용하여 저 사양 디바이스에 실시간으로 운영 될 수 있도록 간단한 구조의 DoS 공격 탐지를 수행한다. 특정 프레임이 지속적으로 송신 시, 10초의 단위 시간동안 특정 threshold을 넘기면 DoS attack으로 간주한다.

3.2 Smart WIDS의 안드로이드 스마트폰인식과 공격 알림

Smart WIDS와 스마트폰 간 블루투스 연결에 소모되는 전력을 최소화하기 위해 블루투스 통신을 이용하여 지속적으로 연결되진 않는다. 즉, WI-FI에 대한 공격이 탐지된 경우와 안드로이드 스마트폰에서 탐지하기를 원하는 WI-FI의 SSID를 넘겨줄 경우, 안드로이드 스마트폰이 mart WIDS에게 통신 가능 영역에 존재함을 알리는 경우에만 통신을 수행한다. 이러한 블루투스의 근거리 통신에 따른 특성으로 공격자의 위협이 탐지될 경우, 2가지의 상황이 존재한다. 관리자가 블루투스 통신영역에 속하는 경우와 그렇지 못한 경우로 나눌 수 있다.

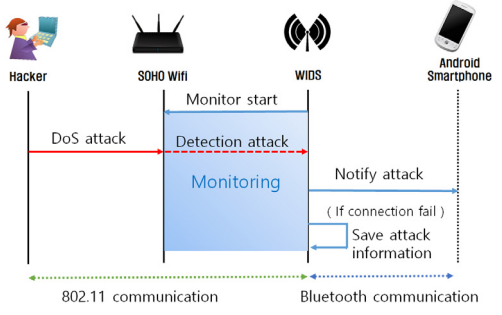

그림 2. 공격 알림 방식

[그림 2]에서 보이는 것과 같이, 관리자가 SOHO환경에 위치하는 경우, 즉 Smart WIDS의 블루투스 통신영역에 스마트폰이 위치한 상황에는 정상적으로 공격 탐지 메시지를 전달 할 수 있다. 반면 관리자가 SOHO환경에서 벗어날 경우, Smart WIDS에서 공격 탐지 후 연결 실패를 통해 스마트폰의 부재를 파악한다. 스마트폰이 통신영역에 돌아올 때 까지 공격 메시지를 저장한다.

이후 관리자의 안드로이드 스마트폰이 Smart WIDS 통신영역에 위치한 경우, 저장된 위협 내용을 전달한다. 이를 이용해 관리자의 부재 상황에서 위협이 계속 유지되는 지 파악케 한다. 즉 공격자의 공격이 성공하여 관리자의 station들이 악성 WI-FI에 연결되어 있는 지를 판단할 수 있게 한다.

[그림 3] (a)와 같이 SOHO WI-FI의 beacon을 수신할 수 없는 영역에서는 안드로이드 스마트폰은 블루투스의 수신 및 연결을 시도 하지 않는다.

(b)와 같이 WI-FI beacon을 수신할 수 있는 경우, 스마트폰은 지속적으로 connect을 시도한다.

(c) 와 같이 Smart WIDS 통신영역에 이르렀을 때, 스마트폰의 연결이 성공하게 되며, Smart WIDS는 스마트폰이 통신영역에 이르렀다는 것을 알게 된다. 이러한 스마트한 알람 방식으로 연결에 따른 자원 소모를 최소화 할 수 있다.

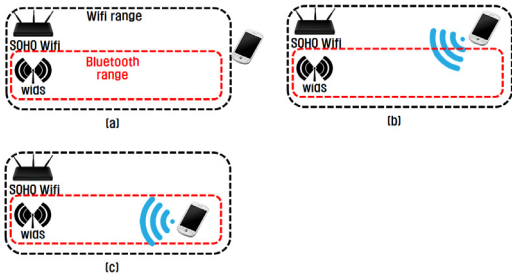

그림 3. Smart WIDS의 안드로이드 스마트폰 인식

4. Smart WIDS 구현

4.1 WIDS 구현

Smart WIDS는 Raspberry Pi2 와 WI-FI dongle로 ralink 5370 usb dongle, Bluetooth dongle로는 NEXT-204BT 로 구성된다. [그림 4] Smart WIDS와 같이 Raspberry Pi2에 WI-FI dongle과 Bluetooth dongle을 usb형식으로 연결하고 전원은 인가하여 하드웨어를 완성한다.

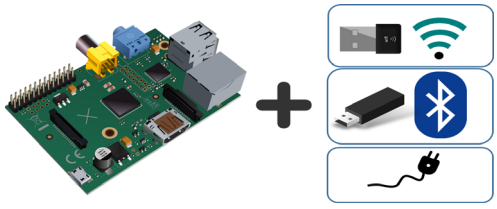

그림 4. Smart WIDS 구성

Raspberry Pi2는 quad-core ARM Cortex-A7 CPU 900MHz 으로 기존 노트북 대비 연산능력이 낮은 만큼, DoS 공격 시의 과도한 연산 작업을 회피하고자, 복잡한 처리 부분을 제거하고 단위 시간에서 특정 threshold값을 넘길 시 공격으로 간주한다. 경험적으로 4대의 PC와 1개의 태블릿, 3개의 스마트폰을 사용하는 경우에 공격 알림이 발생하는 경우가 없고, DoS 공격 툴인 Aircrack-NG 의 Deauthentication DoS 공격 수행 시 최소 64 패킷으로 전송되므로 본 실험에서는 10 패킷을 threshold 값으로 지정하였다. 이 threshold는 동적으로 변경이 가능하다.

100회 이상의 공격에서는 추가 연산을 하지 않는 심플한 방식을 사용한다. Smart WIDS에서 DoS 공격 탐지 소스는 xterm, tshark의 툴을 사용한다[11][12]. xterm은 tshark 프로세스를 호출하기 위해 사용된다. 또 tshark는 무선 패킷을 수신하기 위해 사용되는 프로세스이다.

Smart WIDS의 소프트웨어는 크게 WI-FI의 무선 공격 탐지 기능과 관리자의 안드로이드 스마트폰의 APP과 연동한 스마트 기능으로 구성된다.

```
Start
        monitor_mode( wlan0 )

        ( starting_tshark to save packets for 10 seconds)
        if ( packets exits ) then
                arrangement( file_tshark )
                if( int_deauthentication_packet is higher than int_threshold )
        then
                        ( fill flag of deauthentication )
                else if ( int_disassociation is higher than in_threshold ) then
                        ( fill flag of disassociation )
                end if
        end if
End
```

그림 5. DoS 탐지

Smart WIDS의 무선 공격 탐지 기능은 [그림 5] DoS 탐지 알고리즘에서 보이고 있다. xterm을 이용하여 별도의 프로세스를 생성하여 tshark를 실행한다. 이 프로세스에서는 모니터 모드로 변경시킨 무선 인터페이스와 tshark로 10초간 무선 환경에서의 패킷을 수집한다. 만약 수집된 패킷이 존재한다면, 지정한 SSID의 mac 주소를 참조하여 그 주소가 포함된 패킷의 802.11 프레임을 분석한다. 802.11의 MAC 프레임에서 Type와 SubType, ToDs, FromDs 의 bit를 확인하여 동일한 패킷의 수를 확인한다. 만약 이 패킷의 수가 지정한 threshold값 이상일 시에 DoS 공격으로 간주하게 된다.

스마트 기능은 관리자의 안드로이드 스마트폰으로 블루투스를 이용해 공격 탐지 메시지를 보내는 기능과 블루투스 통신 가능한 거리인지를 확인하는 존재 확인 기능을 가진다.

```
Start
        (setting Bluetooth of RFCOMM profile)
        if ( flag of android smart phone existence is "TRUE")
                try :
                        socket.connect( blue_addr, port )

                except :
                        ( fill flag of android smart phone existence with "FALSE")

                        if ( Is flag of deauthentication ) then
                                ( saving message of deauthentication )
                        else if ( Is flag of disassociation ) then
                                ( saving message of disassociation )
                        end if
                        socket.close()
                        End
                if ( Is flag of deauthentication ) then
                        sock.send(" detecting deauthentication ")
                else if ( Is flag of disassociation ) then
                        sock.send(" detecting disassociation ")
                end if

                socket.close()
        end if

End
```

그림 6. 알고리즘 - 안드로이드 폰으로 공격 알림

Smart WIDS 가 DoS를 감지한 경우, [그림 6]에 보이는 알고리즘으로 안드로이드 폰으로 공격알림 알고리즘이 동작된다. Smart WIDS는 관리자의 스마트폰이 존재 여부를 알리는 플래그를 확인한다. 만약 플래그에

서 관리자가 근처에 있다고 표시 되어 있다면, 사용자에게 공격 내용을 알리기 위해서 블루투스 RFCOMM 프로토콜로 연결을 시도 한다. 만약 연결이 성공된 경우라면, 직접 관리자의 안드로이드 스마트폰으로 메시지를 넘긴다. 반면 연결이 실패하면, 관리자의 안드로이드 스마트폰이 통신 영역, 즉 해당 SOHO환경에 존재하지 않는 것으로 간주한다. 해당 공격이 존재했다는 것을 나타내는 플래그를 저장하고, 탐지한 공격 메시지를 저장하게 된다.

또 Smart WIDS의 관리자의 안드로이드 스마트폰의 존재 확인 기능은 블루투스를 thread를 통해, 패킷 감지 및 연산에 병렬적으로 수행된다. Smart WIDS는 thread가 실행되어, 스마트폰의 연결을 기다린다. 만약 블루투스를 통해서, 관리자의 스마트폰으로부터 연결이 accept 된다면, Smart WIDS는 관리자가 SOHO영역에 존재한다는 플래그를 설정하게 된다. 이때, 탐지된 공격이 있다면, 관리자에게 메시지를 보내게 된다.

4.2 App 구현

안드로이드 스마트폰은 LG G5 로 모델명은 LG-F700을 사용한다.

구현 App은 [그림 7]과 같이 화면을 구성하는 Activity 와 안드로이드에서 백그라운로 동작하기 위한 Service, 부팅 방송을 수신하기 위한 BrocastReceiver로 구성된다.

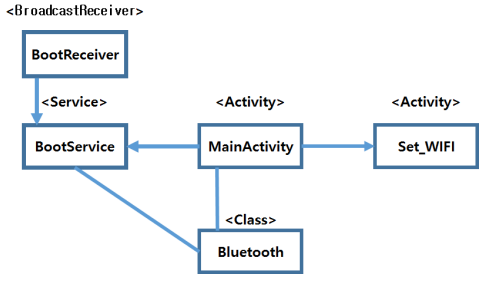

그림 7. App 구성

MainActivity와 Activity는 보호할 WI-FI의 SSID 설정 및 WIDS에게 설정 WI-FI의 SSID 전송을 관리한다. MainActivity는 BootService의 활성화 및 비활성화 시킨다. BootService는 Smart WIDS에게 안드로이드의 존재 유무를 알리며, 공격 정보 수신 기능을 수행한다. BootService와 MainActivity는 Bluetooth class를 활용하여 Smart WIDS와 블루투스로 통신을 수행한다. BootReceiver 은 안드로이드 부팅 시, 본 APP을 실행토록 부팅방송을 받는 BroadcastReceiver이다.

5. Smart WIDS 성능 실험

실험 위치는 경북대학교 3층 건물인 공대7호관에서 진행된다. 이 건물의 각 연구실과 강의실을 SOHO환경으로 가정한다.

5.1 Smart WIDS 패킷 탐지

건물의 구성은 [그림 8]과 같으며, 각 방의 너비는 약 3.6m, 길이 7.2m 로 되어 있다. 306호에서 Smart WIDS와 WI-FI 가 운영되며, 3m 떨어진 공간에 위치한다.

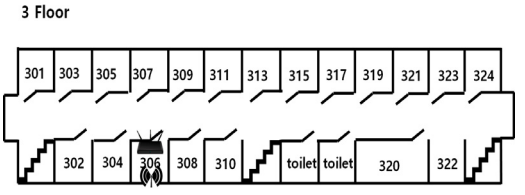

그림 8. 실험 건물 구성도

DoS attack 의 threshold 의 값을 10으로 지정하여, 10초 동안 송신한 Deauthentication 패킷을 측정하였다. [표 1]에서 횡축은 패킷 측정 횟수이고, 종축은 10초 동안 송신된 패킷 량이다. 표 내용은 측정 횟수별 송신된 패킷 탐지 량이다. X는 DoS로 판단되지 못한 경우로, 10초 동안 10개 이하의 패킷을 탐지한 경우다.

표 1. Deauthentication 패킷 측정량

		측정 횟수									
		1	2	3	4	5	6	7	8	9	10
패킷수	11	X	X	11	X	11	11	11	11	11	11
	12	X	X	11	X	X	12	12	11	11	12
	13	13	13	13	12	13	13	12	13	12	13
	14	14	14	13	12	15	15	13	14	14	15
	15	12	11	15	15	15	15	13	14	14	15

[표 1]에서 보이는 바와 같이, Deauthentication 공격 감지에서는 패킷을 11회 송신 시, 10회 중 7회를 탐지하였고, 12회 송신 시 6회 탐지 하였다. 13 패킷이상 송신 시에는 모두 공격 패킷으로 간주되었다. 패킷 13개와 14개, 15개 송신 시, 평균 12.7 개와 13개, 13.9개로 탐지되어 0.3 ~ 1.1 의 손실이 있다.

평균을 계산해보면, 지정 threshold의 2개 이상 시에 모든 패킷을 탐지하는 것을 보인다. aircrack-ng툴을 사용하여[9], 1회 공격 시의 송신 패킷인 32패킷을 비교해 보면 충분히 공격을 탐지해 낼 수 있는 것을 보인다.

5.2 Smart WIDS 공격 알림 시간

관리자 부재 시에 공격자에 의해 공격이 수행된 후, 관리자가 SOHO 영역에 돌아온 상황에서의 스마트폰으로의 공격 알람 시간을 측정한다. 실험은 [그림9] 와 같이 306호를 SOHO환경이라 가정하고 Smart WIDS에서 1m 떨어진 (A)와 SOHO의 입구로 들어오는 상황으로 Smart WIDS와의 거리가 6m 지점인 (B)를 입구로 가정한다. 또, SOHO 사무실의 외부인 (C)지점에서 10m 떨어진 위치로 시간을 측정한다.

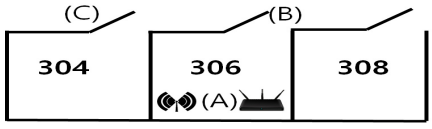

그림 9. 공격 알림 실험 환경

실험 결과는 [그림 10]에서 보이고 있다. 1m, 6m 의 위치인 경우, 스마트폰이 공격알람 수신 시간이 6초 내이다. 벽면이 없는 공개된 영역에서의 수신 시간은 큰 차이 없이 공격 알람을 받을 수 있는 것을 확인 할 수 있다. 반면 10m에 위치한 경우, 최소 20초에서 78초 내로 수신 시간의 차가 큰 것을 보이고 있다. 위의 결과로 Smart WIDS는 벽과 같은 무선 통신의 장애가 없는 6m 내의 공간에서는 원활한 공격 알람을 제공받을 수 있으며, 벽과 같은 제한된 공간에서 역시 2분 내로 공격 알람을 수신 받을 수 있다는 것을 보인다. 즉 관리자가 SOHO 영역에서는 6초 내로 빠르게 공격 알람을 받는 것을 보인다.

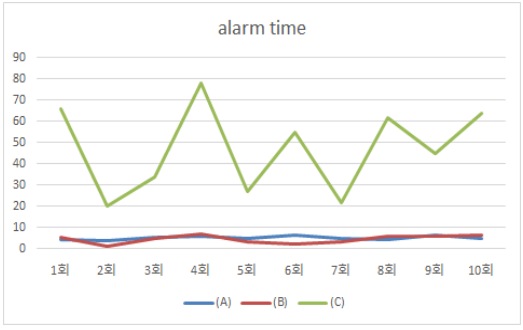

그림 10. 공격 알림 수신 시간

5.3 Smart WIDS 와 smart phone 간 연결 영역

[그림 11]의 동그라미 친 영역에서 안드로이드 스마트폰으로 WI-FI 신호를 측정하며, Smart WIDS와의 블루투스 연결 여부를 확인한다. WI-FI 신호 수신 장비는 구현된 안드로이드 스마트폰을 사용한다. 또, 신호 세기는 각 방에서 2분 간격으로 수신된 Beacon의 신호 세기를 5회 측정하였다.

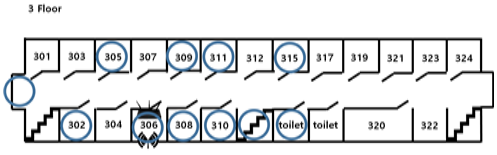

그림 11. 공격 알림 실험 환경

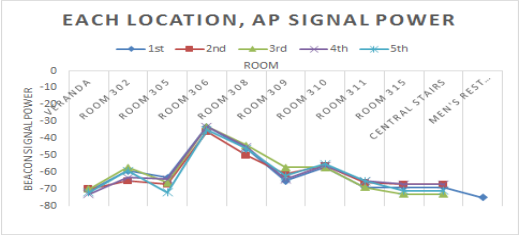

그림 12. 3층 장소별, 신호 세기.

[그림 12]은 각 장소별 WI-FI beacon 신호 세기를 보이고 있다. 세로축은 Beacon 신호 세기로 dBm 단위이며, 가로축은 신호를 측정한 각 방이다. WI-FI가 동작하는 306호에서 멀어질수록 WI-FI신호는 줄어들며, 그 이상에서의 신호는 smart phone이 감지할 수 없다. 추가로 블루투스와 Smart WIDS와의 연결 여부를 확인해보면, -70dBm 이하에서 블루투스 연결이 불가능하였다. 위의 실험결과를 토대로 WI-FI 기준 약 10m 이내에서 Smart WIDS와 블루투스 간의 정상적인 식별이 이루어진다는 것을 보이고 있다. 즉 SOHO환경의 벽면으로 가려진 공간에서, 스마트폰을 가진 관리자가 10m내에 위치 할 시 공격 여부를 확인 할 수 있다는 것을 보인다.

6. 결론 및 차후과제

다양하고 많은 장소에서 WI-FI를 이용한 무선 네트워크 서비스를 제공하고 있다. 반면 이러한 WI-FI를 통한 무선 네트워크는 IEEE 802.11 프로토콜에 대한 무선 공격에 취약하다. 이런 무선 공격을 방어하기 위한 방안으로 WIDS를 활용한다. WIDS는 재정적, 관리적인 제한으로 인해 대기업과 같은 지원이 가능한 경우에 활용되고 있다. 반면, SOHO 환경에서는 재정적, 관리의 한계로 WIDS를 활용하기에 적합하지 않다. 즉 저렴한 무선 공격 탐지가 되어야 하며, WI-FI 제공자의 관리상의 편의가 제공되어야한다. 타 무선 랜 침입탐지 경량 시스템은 기존 WIDS에 비해 노트북에서 수행될 수 있는 경량화를 이루었으나, 모든 SOHO환경에서 장기간 노트북이 동작하여야 하고, 탐지 결과에 대한 주의가 요구된다.

본 논문에서는 실질적으로 SOHO 및 영세업체에서 WI-FI를 보호할 수 있는 Smart WIDS를 구현하였다. 저렴한 디바이스와 스마트폰을 활용하여 구축비용을 최소화 하였다. 또, 안드로이드 스마트폰을 Smart WIDS의 인터페이스로 구현하여 관리의 편의성과 즉각적인 공격 알람을 제공한다. 또, 관리자의 외출 복귀후의 상황에서 Smart WIDS와 안드로이드 스마트폰간의 효율적인 인식 방법을 제공하였다.

본 논문에서는 DoS 공격을 탐지하기 위한 방식으로 임의의 threshold 값을 지정하여 사용하였다. 이와 같은 경우는 다양한 통신 상황을 수렴하지 못하는 단점을 가지고 있다. 따라서 Raspberry Pi2와 같은 저사양 기기에서 무리 없이 동작할 수 있는 능동적으로 threshold 값을 조절할 수 있는 연구가 계속 진행 중이다.

참 고 문 헌

[1] C. D. Mano and A. Striegel, “Resolving WPA limitations in SOHO and open public wireless networks,” IEEE Wireless Communications and Networking Conference (WCNC 2006), pp.617-622, 2006.

[2] J. Park, M. Park, and S. Jung, “A whitelist-based scheme for detecting and preventing unauthorized AP access using mobile device,” Journal of Korean Institute of Communications and Information Sciences, Vol.38, pp.632-640, 2013.

[3] J. Kim, A. Kim, J. Yuk, and H. Jung, “A Study on Wireless Intrusion Prevention System based on Snort,” International Journal of Software Engineering and Its Applications, Vol.9, pp.1-12, 2015.

[4] H. Kim, S. Kim, H. Lee, and H. Jung, “Light-weight System Design & Implementation for Wireless Intrusion Detection System,” Journal of the Korea Institute of Information and Communication

Engineering, Vol.18, pp.602-608, 2014.
[5] N. Baharudin, F. H. M. Ali, M. Y. Darus, and N. Awang, "Wireless Intruder Detection System (WIDS) in Detecting De-Authentication and Disassociation Attacks in IEEE 802.11," International Conference on IT Convergence and Security (ICITCS), pp.1-5, 2015.
[6] W. Hsieh, C. Lo, J. Lee, and L. Huang, "The implementation of a proactive wireless intrusion detection system," International Conference on Computer and Information Technology (CIT'04), pp.581-586, 2004.
[7] G. Chen, H. Yao, and Z. Wang, "An intelligent WLAN intrusion prevention system based on signature detection and plan recognition," International Conference on Future Networks (ICFN'10), pp.168-172, 2010.
[8] C. He and J. C. Mitchell, "Analysis of the 802.11 i 4-Way Handshake," ACM workshop on Wireless security, pp.43-50, 2004.
[9] https://www.aircrack-ng.org/index.html.
[10] Ar Kar Kyaw, Yuzhu Chen, and Justin Joseph, "Pi-IDS: evaluation of open-source intrusion detection systems on Raspberry Pi 2," IEEE International Conference on Information Security and Cyber Forensics (InfoSec), pp.165-170, 2015.
[11] http://invisible-island.net/xterm/
[12] https://www.wireshark.org/

저 자 소 개

김 철 홍(Cheol-Hong Kim) 정회원

- 2013년 2월 : 영남대학교 전자공학부(공학사)
- 2016년 8월 : 경북대학교 전자공학부(공학석사)

<관심분야> : 네트워크, 무선 보안

정 임 영(Im Y. Jung) 정회원

- 1993년 2월 : 포항공과대학교 화학과 이학사
- 1999년 2월 : 서울대학교 전산과학과 이학사
- 2001년 2월 : 서울대학교 컴퓨터공학부 공학석사
- 2010년 8월 : 서울대학교 컴퓨터공학부 공학박사

<관심분야> : IoT 보안, 데이터 및 시스템 보안, 스토리지 시스템, 분산컴퓨팅시스템, 클라우드 컴퓨팅