파이썬을 이용한 무선 공유기 DoS 공격

지병규, 문봉교
동국대학교 컴퓨터공학과
e-mail : gub122@naver.com

DoS Attack at Access Point Using Python

Byoung-Kyu Ji, Bong-Kyo Moon
Dept of Computer Engineering, Dongguk University

요 약

최근 무선 네트워크의 발달로 카페나 대학교, 지하철 등 대부분 공간에서 무선 공유기를 통해 WIFI 서비스가 제공되고 있다. 무선 공유기는 IEEE의 표준 프로토콜인 802.11을 사용하는데, 이 프로토콜의 상용화된 버전인 802.11n의 관리 프레임은 암호화되어 있지 않아 쉽게 악용될 수 있다. 이는 Rogue AP, 비밀번호 크래킹 등 다른 공격으로의 발판이 될 수 있으며 이에 따라 보안에 위협이 될 수 있다. 본 논문에서는 파이썬을 이용해 이러한 취약점에 대해 공격하는 과정을 분석하고 이를 통해 발생할 수 있는 취약점과 대책에 대해 논의할 것이다.

1. 서론

IEEE의 802.11[1]은 무선 LAN을 위한 컴퓨터 무선 네트워크에 사용되는 기술이다. 초기 버전인 802.11a 에서부터 최근에 배포된 802.11ay까지 대역폭의 증가와 변조 방식의 변경 등 많은 변화가 있었다. 현재 대한민국에는 802.11n이 상용화되어 있고, WEP, WPA/PSK, WPA2/PSK 등의 암호화를 제공하여 보안을 유지한다. 하지만 802.11n의 관리 프레임은 암호화되어 있지 않아 쉽게 악용될 수 있다.

또한, 802.11n을 이용하는 무선 공유기를 통해 WIFI 서비스를 제공하는 대부분 업체에서 이런 취약점에 대해 인식하고 예방하기 힘들어서 이를 근본적으로 방지해주는 프로토콜이 필요하다.

802.11w[2]에서 이를 막기 위해 보호된 관리 프레임을 제공하지만, 아직 국내에 상용화되어있지 않아서 관리 프레임을 이용한 DoS 공격을 현실적으로 막을 방법이 없다.

그런데 관리 프레임을 통한 DoS 공격으로 인해 가용성에 대한 위협뿐만 아니라 이를 통해서 추가적인 취약점이 생긴다. 무선 공유기 암호가 유출된다면 개인정보에 위협이 될 수 있다. 본 논문에서는 파이썬으로 Scapy를 이용하여 DoS 공격을 시연하고 이를 통해 생기는 추가적인 취약점과 대책에 대해서 논의해 볼 것이다.

2. 연구의 배경

2.1. 인증 해제 프레임

802.11의 프레임은 크게 3가지로 분류할 수 있는데, 데이터, 관리, 제어 프레임이다. 관리 프레임 중에서 인증 해제 프레임은 무선 공유기가 정책을 변경하거나 사용자가 연결을 끊을 때 발생하는 프레임으로 이를 악의적으로 공격 대상에게 보내게 된다면 DoS 공격을 유발하게 된다. 이는 단순히 DoS 공격뿐만 아니라 그다음의 비밀번호 크래킹이나 Rogue AP 공격 등으로 이어져 개인정보 유출에 위협을 줄 수 있다.

2.2. Scapy

Scapy[3]는 사용자가 네트워크 패킷을 스니핑, 위조, 송신할 수 있게 하는 네트워크 관련 기능을 가진 파이썬 기반의 라이브러리이다. Scapy를 이용하면 어떤 802.11 패킷도 캡쳐해서 볼 수 있으므로[4] 이 정보를 분석하고 패킷을 위조하여 송신함으로써 다양한 공격에 활용될 수 있다. 본 논문에서는 비컨 프레임을 캡쳐하고, 그 속에 들어있는 무선 공유기의

MAC 주소인 BSSID를 근원지로, 브로드 캐스트를 목적지로하고 인증 해제 패킷으로 위조하여 DoS 공격을 실행한다.

3. 연구의 내용

3.1. DoS 공격

3.1.1. 공격 환경

Host OS인 Windows 10에서 VMWare를 설치하여 Debian 리눅스 기반의 모의 공격/테스트에 특화된 Kali 리눅스를 사용했다.[5] VMWare에서도 무선 네트워크를 인식하게 하려고 외장형 무선 네트워크 어댑터 삼성 무선 동글이 WIS12ABGNX를 사용했으며 Python 2.7.12, Scapy 2.3.2 버전을 사용했다.

3.1.2. 무선 어댑터 모드

DoS 공격을 실행하기 위해 먼저 공격 대상을 설정해야 하는데, 공격 대상은 공격자가 자신의 위치에서 비컨 프레임을 수신할 수 있게 되는 모든 무선 공유기가 될 수 있다. 비컨 프레임을 수신하기 위해서는 먼저 무선 네트워크 어댑터의 모드를 모니터 모드로 설정해야 한다.[6] 리눅스 환경에서는 <그림 1>과 같이 ifconfig, iwconfig 명령어로 모니터 모드로 변경할 수 있다.

```
root@kali:~# iwconfig
lo        no wireless extensions.

wlan0     IEEE 802.11abgn  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off

eth0      no wireless extensions.

root@kali:~# ifconfig wlan0 down
root@kali:~# iwconfig wlan0 mode monitor
root@kali:~# ifconfig wlan0 up
root@kali:~# iwconfig
lo        no wireless extensions.

wlan0     IEEE 802.11abgn  Mode:Monitor  Frequency:2.412 GHz  Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off

eth0      no wireless extensions.

root@kali:~#
```

<그림 1> 무선 네트워크 어댑터 모드 변경

3.1.3. 공격 소스 분석

먼저, 패킷을 스니핑할 무선 네트워크 어댑터의 인터페이스와 스니핑할 패킷의 수를 입력받는다. 그리고 여러 채널에 대해 스니핑하기 위해서 iwconfig 명령어의 channel 옵션으로 인터페이스의 채널을 바꾼다. sniff 함수로 채널마다 앞서 입력받은 패킷의 수만큼 스니핑하여 PacketHandler 함수를 통해 중복되지 않은 BSSID가 발견된 경우 ap_li 리스트에 삽입하고 콘솔 창에 띄워준다. 한 채널에서 입력받은 패킷의 수만큼 스니핑 한 경우 공격을 시도할 무선 공유기와 인증 해제 패킷의 수를 입력받고, deauth_ap 함수를 통해 목적지를 브로드 캐스트로, 근원지를 발견한 무선 공유기의 MAC 주소로 하고 인증 해제 패킷으로 위조한 패킷을 송신해서 공격을 시도한다. 그리고

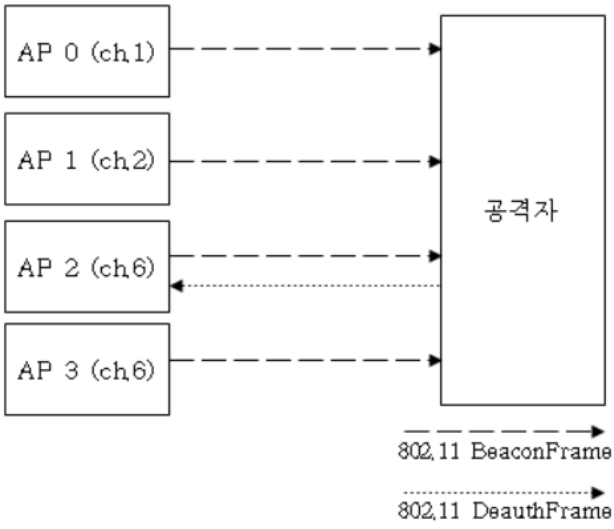
```
from scapy.all import *
import os
ap_list = []
no_ap=0
broadcast="ff:ff:ff:ff:ff:ff"
def PacketHandler(pkt):
        global no_ap
        if pkt.haslayer(Dot11):
                if pkt.type==0 and pkt.subtype==8 :
                        if pkt.addr2 not in ap_list:
                                ap_list.append(pkt.addr2)
                                print "%d AP MAC:%s with SSID: %s" %(no_ap,pkt.addr2,pkt.info)
                                no_ap=no_ap+1
def deauth_ap(bssid,number):
        packet=RadioTap()/Dot11(type=0,subtype=12,addr1=broadcast,addr2=bssid,addr3=bssid)/Dot11Deauth(reason=7)
        for n in range(number):
                sendp(packet)

def main():
        interface=raw_input("Name of interface : ")
        cnt=input("Number of packet to sniff : ")
        for channel in range(1,14):
                os.system("iwconfig "+interface+" channel "+str(channel))
                print "Snifiing on channel"+str(channel)
                sniff(iface=interface,prn=PacketHandler,count=cnt)
                choice_ap=input("AP to attack : ")
                attack_no=input("number of packet to attack : ")
                print("Attack to "+ap_list[choice_ap])
                deauth_ap(ap_list[choice_ap],attack_no)

if __name__== "__main__":
        main()
```

<그림 2> 소스 코드[7]

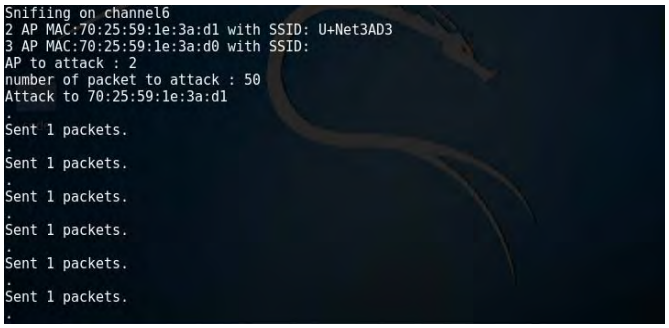
802.11n의 일부 채널(1-14)에 대해 위의 과정을 반복 실행한다. 공격의 개요도는 <그림 3>과 같다.



<그림 3> 공격의 개요도

3.1.4. DoS 공격 결과

가정용 무선 공유기인 U+Net3AD3을 대상으로 <그림 2>의 소스코드를 실행해서 공격을 시도해 보았고, 실행 결과는 <그림 4>와 같았다. 그리고 인증 해제 패킷을 송신하는 동안에는 해당 무선 공유기를 통해 네트워크 서비스를 이용하던 모든 단말기가 연결이 끊기는 것을 확인할 수 있었다.



<그림 4> 실행 화면

3.2. 인증 해제 공격을 통한 취약점

WPA/PSK, WPA2/PSK 방식의 암호화를 사용하는 무선 공유기에서 DoS 공격을 통해 인증이 해제된 단말기는 재 인증을 위해 무선 공유기에 PSK와 함께 연결을 4-way handshake 방식[8]으로 요청한다. 이 때, 패킷을 계속 스니핑하고 있던 공격자는 airodump-ng와 같은 툴로 4-way handshake를 캡쳐할 수 있고 이는 PSK를 알아내기 위한 사전 공격에 악용될 수 있다. 또한, 재인증을 위해 연결을 요청한 다른 무선 공유기가 Rogue AP인 경우에도 PSK가 노출되어 개인 정보가 노출될 수 있다.

4. 결론

4.1. 취약점

802.11n의 관리 프레임이 암호화되어있지 않아 누구나 쉽게 패킷을 조작하여 악용할 수 있다. 관리 프레임 중에서 인증 해제 프레임을 악용하여 목적지를 브로드 캐스트, 근원지를 공격하려는 무선 공유기의 MAC 주소로 한 후에 패킷을 위조하여 송신한다면 패킷이 송신되는 동안 주변의 모든 단말기의 연결이 끊어진다. 이는 단순히 DoS 공격으로 가용성에 대한 위협뿐만 아니라 WPA/PSK, WPA2/PSK 방식에서는 PSK에 대한 사전 공격, Rogue AP의 효과적인 공격을 위한 도구가 될 수 있다.

4.2. 해결 방안

보호된 관리 프레임을 제공하는 IEEE 802.11w 프로토콜을 지원하는 무선 공유기를 사용하는 경우 <그림 2>의 소스코드를 통한 DoS 공격은 막을 수 있지만 802.11w에서 유니캐스트 방식과 달리 브로드캐스트 또는 멀티캐스트를 사용하여 관리 프레임을 전송할 때는 프레임 자체를 암호화하지 않고 MIC(Message Integrity Code)를 추가하여 전송하기 때문에[9] 공격자가 관리 프레임을 도청하여 MIC를 탈취하면 DoS 공격을 막을 수 없다.

그리고 관리 프레임이 브로드캐스트로 전송되었을 때 인증 해제 패킷을 무시하는 경우에도 <그림 2>의 소스코드를 통한 DoS 공격을 막을 수 있지만, 이 경우에는 특정 단말기를 대상으로 하는 DoS 공격을 막을 수 없다. 따라서 802.11w를 사용하되 브로드캐스트로 전송된 인증 해제 패킷을 무시한다면 무선 공유기 DoS 공격을 막을 수 있을 것이다.

Acknowledgement

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 SW중심대학지원 사업의 연구결과로 수행되었음 (R7116-16-1014)

참고문헌

[1]https://ko.wikipedia.org/wiki/IEEE_802.11
[2]https://en.wikipedia.org/wiki/IEEE_802.11w-2009
[3]https://phaethon.github.io/scapy/api/introduction.html#about-scapy
[4]Varun Chopra, Sushil Mehra (2014). Cracking and Hardening Hidden SSID Mechanism in 802.11 using PYTHON. IJAIS, Volume 7, Number 11
[5]https://www.offensive-security.com/kali-linux-vmware-virtualbox-image-download/
[6]Aigerim Ismukhamedova, Yelena Satimova, Andrei Nikiforov, Natalia Miloslavskaya (2016). Practical Studying of Wi-Fi Network vulnerability. IEEE (DIPDMWC), 227-232.
[7]https://github.com/JiByungKyu/deauth_wifi/blob/master/sniff.py
[8]https://en.wikipedia.org/wiki/IEEE_802.11i-2004
[9]송왕은, 정수환 (2016). IEEE 802.11w 무선 보안 표준 기술. 한국통신학회지(정보와통신), 33(3), 74-79.