网络协议安全实验报告

黄予　2013011363　计 34

1. 任务描述

处于同一局域网的三台主机（可以使用虚拟机），其中一台主机为攻击机。在攻击机上利用 Scapy 伪造数据包，对另外两台靶机进行 ARP 欺骗,实现窃听靶机之间的会话，在实现 ARP 欺骗的基础上，进一步实现中间人攻击。需完成以下两个要求：

1. 使用 Scapy 实现窃听另外两台靶机的会话。例如窃听并提取另外两台靶机之间 FTP 或者 HTTP 会话的登录账号。
2. 对另外两台靶机进行中间人攻击，实现对会话进行篡改。例如对靶机间的 HTTP 会话进行注入，修改 HTTP 响应。

2. 环境配置

2.1 攻击机 A

硬件：MacBook Pro 笔记本（2015 年中）
操作系统：虚拟机 Ubuntu 15.10
网络模式：桥接

2.2 靶机 B

硬件：Dell Vostro 5460 笔记本
操作系统：windows 7

2.3 靶机 C

硬件：SAMSUNG SM-A5000 手机
操作系统：Android 5.0.2

2.4 路由器

硬件：TL-MR22U 路由器
工作模式：无线路由模式

3. 实验过程

3.1 初始配置

攻击机 A、靶机 B、靶机全部连接至路由器，ip 分配使用 DHCP 协议。查看各个主机的 ip 地址等信息如下：

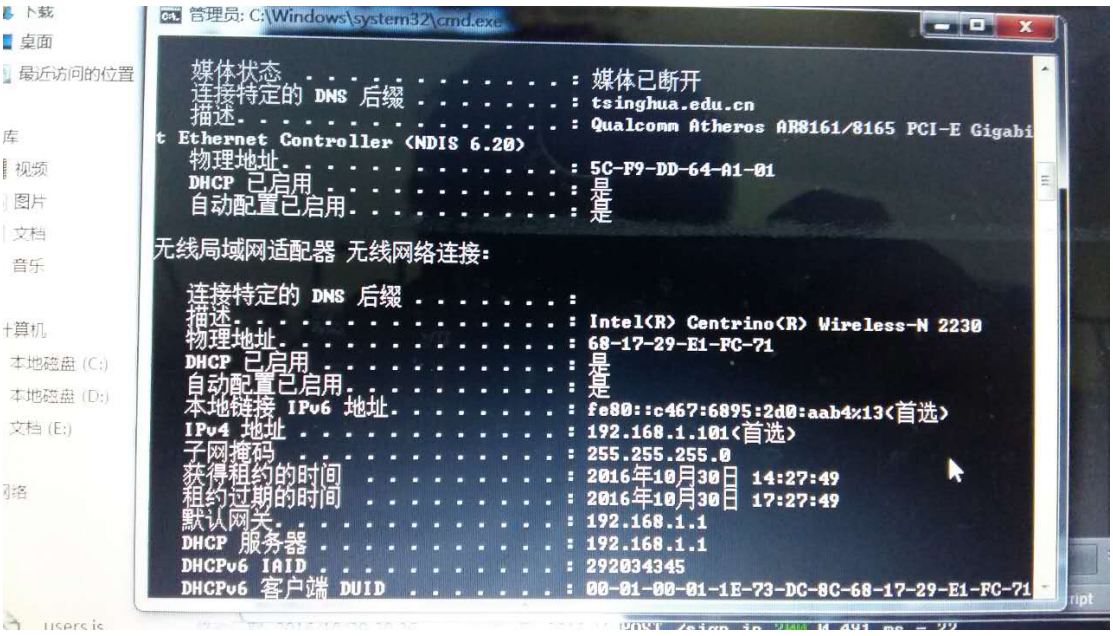
1. 攻击机 A：

```
hy@hy-virtual-machine:~/work/web security/lab1/testScapy$ ifconfig
eno16777736 Link encap:以太网  硬件地址 00:0c:29:a3:22:a6
          inet 地址:192.168.1.104  广播:192.168.1.255  掩码:255.255.255.0
          inet6 地址: fe80::20c:29ff:fea3:22a6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
          接收数据包:315801 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:172458 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节:378956912 (378.9 MB)  发送字节:62950650 (62.9 MB)

lo        Link encap:本地环回
          inet 地址:127.0.0.1  掩码:255.0.0.0
          inet6 地址: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  跃点数:1
          接收数据包:4902 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:4902 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:0
          接收字节:480903 (480.9 KB)  发送字节:480903 (480.9 KB)
```

IP 地址：192.168.1.104
MAC 地址：00:0c:29:a3:22:a6
网卡名：eno16777736

2. 靶机 B：



IP 地址：192.168.1.101
MAC 地址：5c:f9:dd:64:a1:01

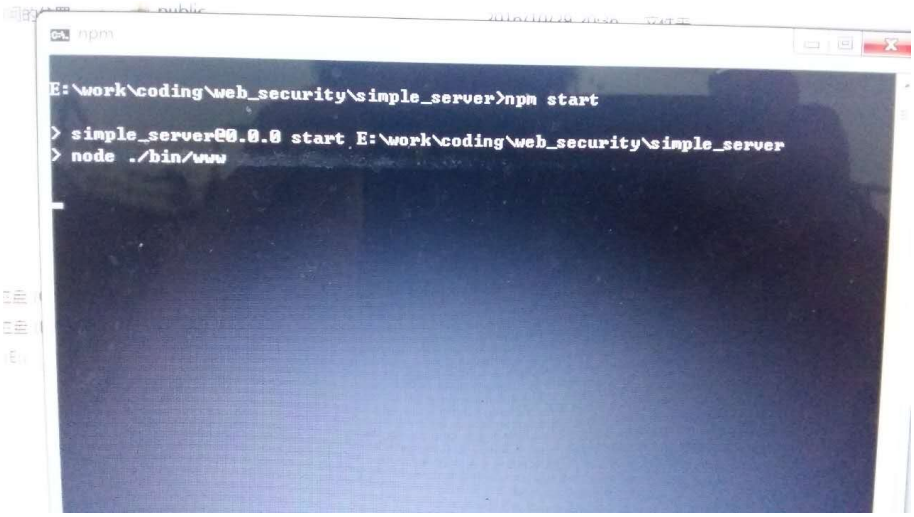
3. 靶机 C：

IP 地址：192.168.1.102

MAC 地址：a8:7c:01:5b:f6:c8

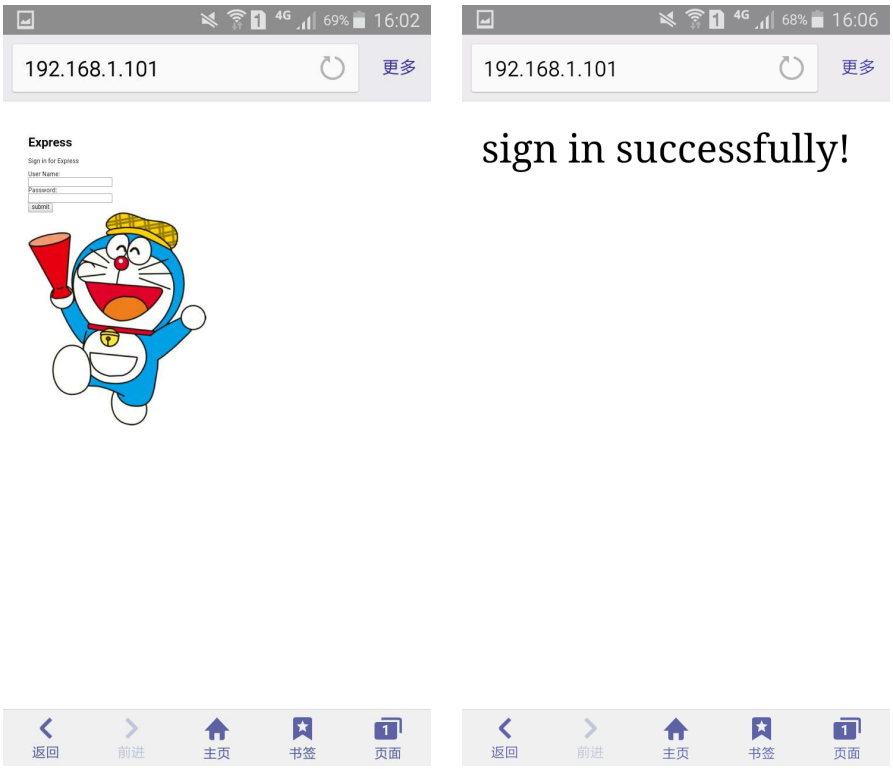
3.2 攻击前的表现

3.2.1 靶机 B 启动 http 服务



使用 node.js，套用 express 框架，端口使用 80。

3.2.2 靶机 C 访问靶机 B 的 http 服务



url 为 192.168.1.101 的 http 页面很简单，仅仅有一个 post 表单和一个图片，填写表单后点击“submit”按钮会跳转到 192.168.1.101/sign_in 页面，该页面仅有一句话：“sign in successfully!”。

3.3 攻击后的表现

3.3.1 攻击原理

攻击机 A（192.168.1.100）通过 scapy 每隔 0.5s 向靶机 C（192.168.1.102）发送 ARP 包，内容为：IP 为 192.168.1.101 的主机（靶机 B）的 mac 地址是 00:0c:29:a3:22:a6（攻击机 A），从而污染靶机 C 的 ARP 缓存表。对靶机 C 而言，攻击机 A 已经伪装成了靶机 B。当靶机 C 访问靶机 B 的 http 服务时，其流量会流经攻击机 A，这时在攻击机 A 上使用 mitmproxy 设置代理，窃取并修改经过流量中的信息。

3.3.2 源代码说明

1. ARP.py：不断向靶机 C 发送错误的 ARP 包。
2. attack.py：窃取靶机 C 向靶机 B 提交的 form 表单的信息；将靶机 B 的 http 响

应中的图片替换为指定图片。

3.3.3 配置攻击机 A

1. 在 ARP.py 写入攻击机 A 的网卡名，以及配置 IP

```
35  iface = 'eno16777736'
36  #iface = 'en0'                          #网卡
37
38  psrc = getLocIP(iface)          #获取本机IP地址（实际上未用到）
39  hwsrc = getLocMAC(iface)        #获取本机mac地址
40
41  pdst = '192.168.1.102'          #将该ARP包发给IP为pdst的主机
42  gpsrc = '192.168.1.101'         #攻击者伪装成的ip
```

如图所示，网卡名为 eno16777736（攻击机 A），pdst 设为 192.168.1.102（靶机 C），gpsrc 设为 192.168.1.101（靶机 B）。

2. 设置攻击机 A 的 IP 转发以及端口映射

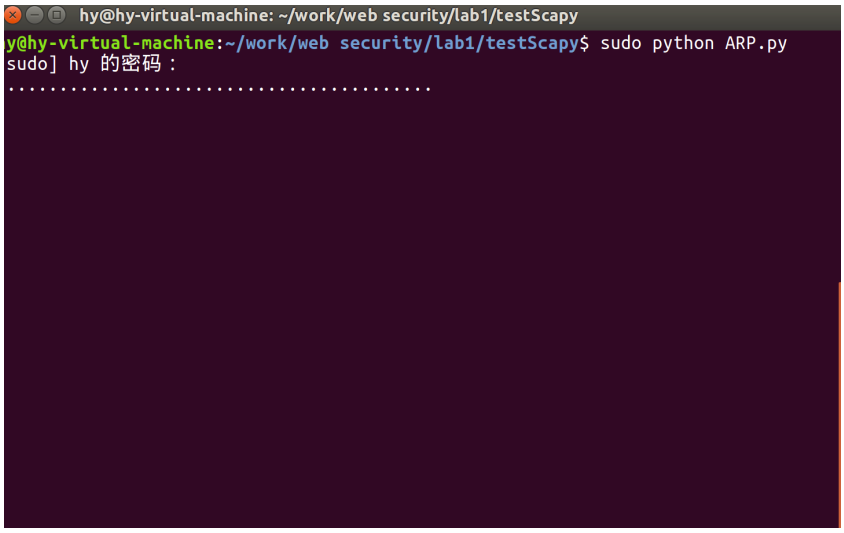
在终端输入如下命令：

```
sudo sysctl -w net.ipv4.ip_forward=1
sudo iptables -t nat -F
sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-ports 8080
```

以上命令的含义分别为：

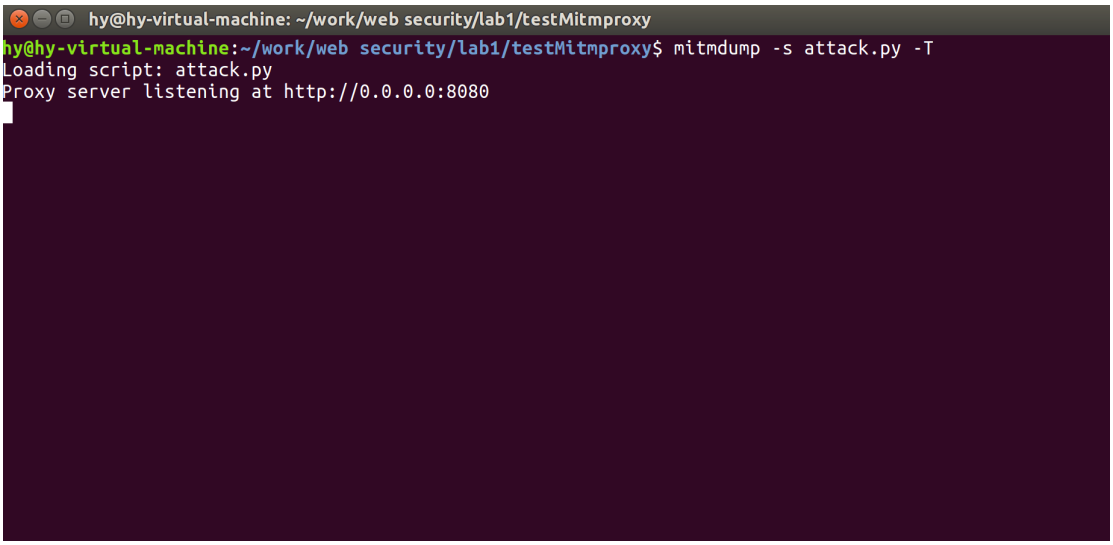
开启 IP 转发
清空 nat 表
将 http 端口 80 到 8080 的映射，因为 mitmproxy 的默认代理端口为 8080

3. 运行 ARP.py



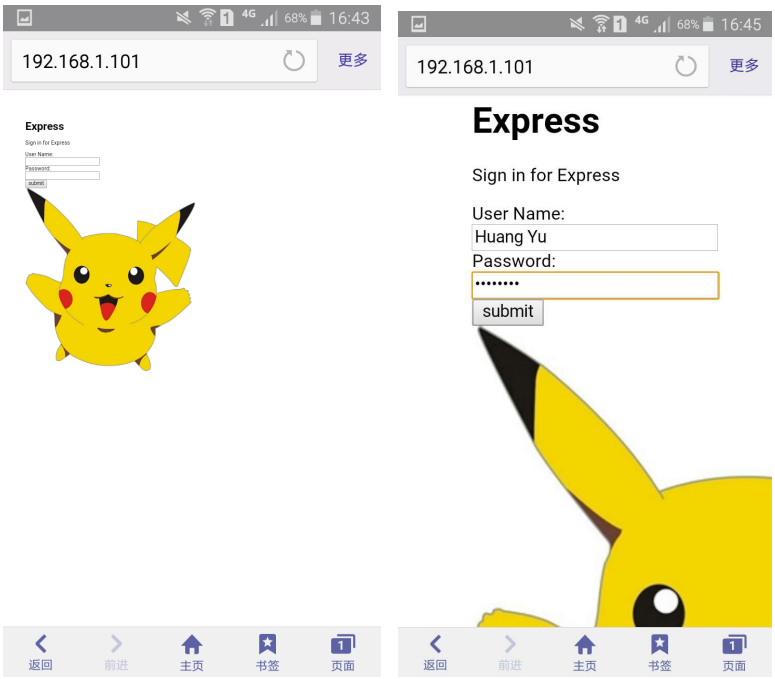
4. 以 attack.py 为脚本，运行 mitmdump 透明代理

另外开启一个新的命令行终端，执行命令：mitmdump –s attack.py -T

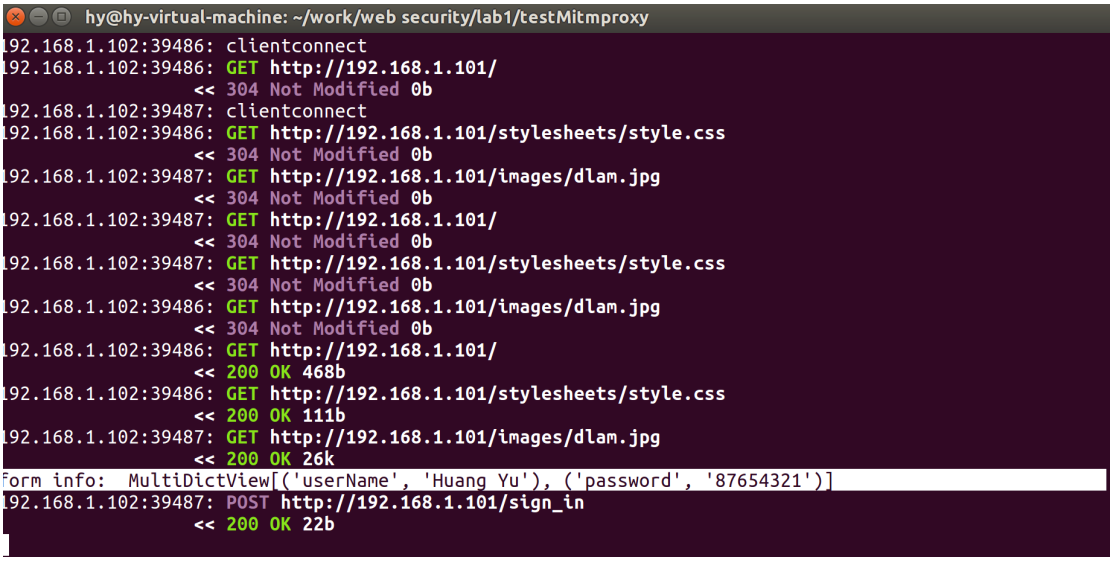


attack.py 脚本的作用为

5. 靶机 C 访问靶机 B



可见靶机 C 的页面的图片已被替换为另一张皮卡丘图片，填写表单后点击 submit，攻击机 A 的终端如下所示：



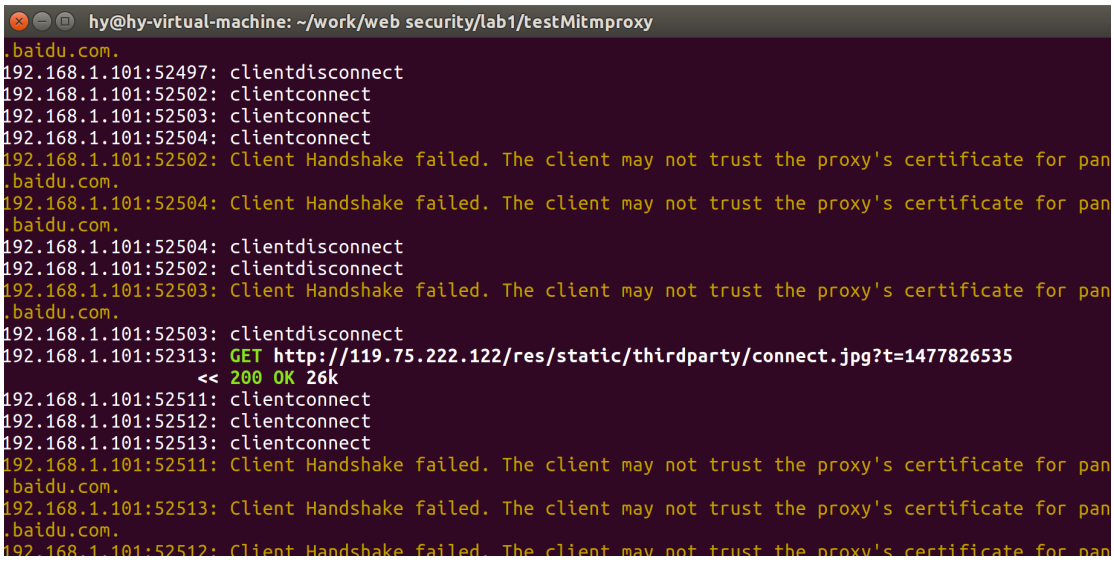
注意到高亮部分，攻击机 A 已经窃取到表单中的用户名与密码信息。至此，两个实验要求均已完成，即窃听 http 会话的账号与篡改 http 响应。

注：靶机 C 应注意清除缓存后再访问

3.4 拓展：冒充网关

```
37  psrc = getLocIP(iface)          #获取本机IP地址（实际上未用到）
38  hwsrc = getLocMAC(iface)        #获取本机mac地址
39
40  pdst = '192.168.1.102'          #将该ARP包发给IP为pdst的主机
41  gpsrc = '192.168.1.1'           #攻击者伪装成的ip
```

将 gpsrc 改为 192.168.1.1 后，使用手机访问外网，例如 www.qq.com，发现手机断网，攻击机终端的显示如下图所示：



可见有代理不被信任的问题，看来利用 ARP 欺骗窃听局域网下的其他主机没有

想象中的容易，但是让目标至少断网是很容易做到的。

4. 实验收获

1. 对 ARP 欺骗攻击理解更加深入
2. 实际环境下（3.4 拓展）的攻击有一些困难，还需要进一步学习