

网络安全实验报告

实验目的

1. 掌握libnet数据包构造的原理
2. 编程实现基于libnet的数据包构造，结合前面的实验给出验证过程，能够对源码进行解释。

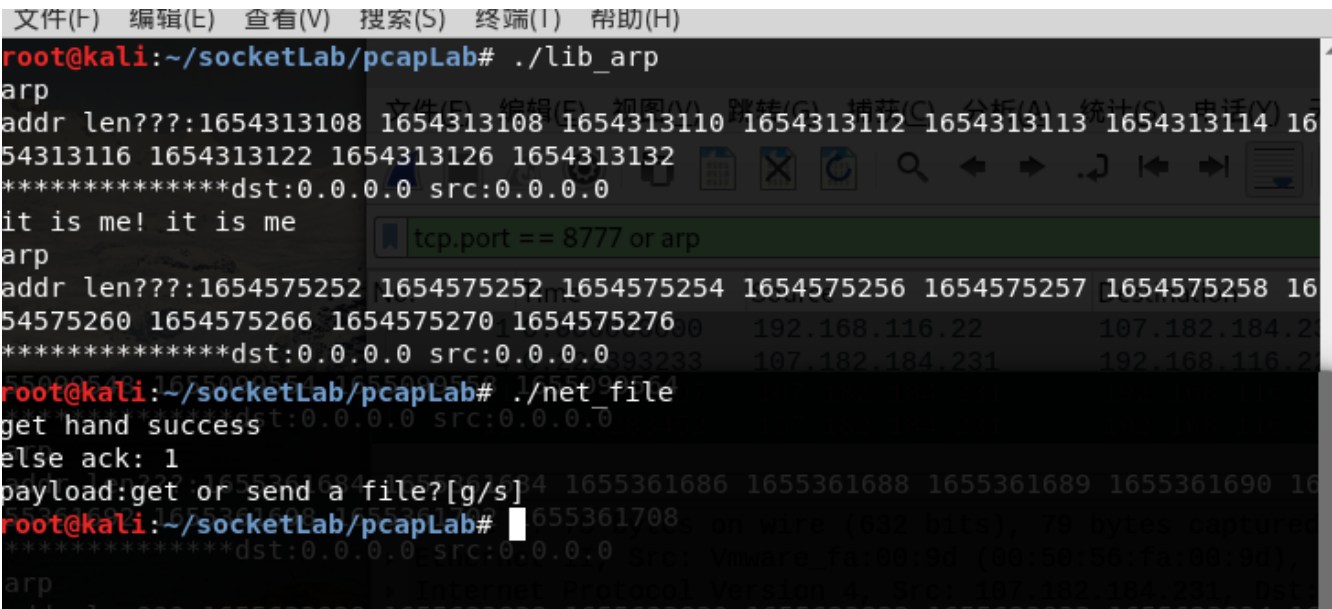
实验内容

用libnet API实现与远端socket server进行TCP三次握手。

- libnet_get_file.cpp（实际上并没有get file，只get了一个字符串）：
 - 使用libnet构造并发送tcp握手的 syn 和 ack包
 - 使用libpcap嗅探远端的tcp syn-ack包
- libnet_arp.cpp：
 - 监听ARP请求，伪造arp回复
 - 使用libnet构造发送tcp包没有使用系统的协议栈，不占用端口，故系统会对远端的tcp握手包回复rst。为了避免这一点，选择一个空闲的内网ip作为“本机ip”，当收到对这个ip的arp请求时，回复本机的mac地址。这样数据包会路由到本机，而系统不会发送rst。

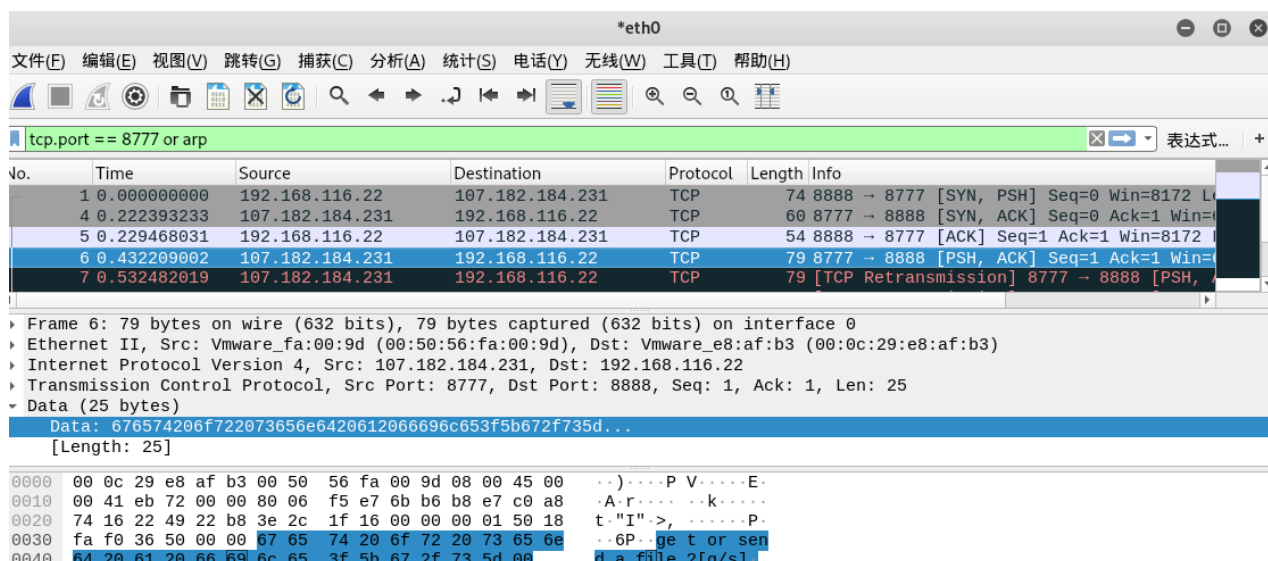
实验结果

- 命令行运行结果，成功获取server提示字符串



```
root@kali:~/socketLab/pcapLab# ./lib_arp
arp
addr len???:1654313108 1654313108 1654313110 1654313112 1654313113 1654313114 16
54313116 1654313122 1654313126 1654313132
*****dst:0.0.0.0 src:0.0.0.0
it is me! it is me
arp
addr len???:1654575252 1654575252 1654575254 1654575256 1654575257 1654575258 16
54575260 1654575266 1654575270 1654575276
*****dst:0.0.0.0 src:0.0.0.0
root@kali:~/socketLab/pcapLab# ./net_file
get hand success
else ack: 1
payload:get or send a file?[g/s]
root@kali:~/socketLab/pcapLab#
```

- 捕包软件截图，成功进行三次握手。



实验总结

- 在arp程序的编写中遇到了迷之bug，经过漫长的面向搜索引擎debug，发现是因为编译器对我的arp结构体进行了填充，6字节的mac地址数组变成了8字节。
 - 使用宏，把受到填充影响的地址偏移到正确位置解决了问题
- 熟悉了libnet，libpcap API的基本调用，加深了对TCP/IP协议的认识。