南开大学

Nankai University

南　　开　　大　　学

网　络　空　间　安　全　学　院

网络技术与应用

实验 3：通过编程获取 IP 地址与 MAC 地址的对应关系

姓名：郑盛东

学号：2010917

年级：2020 级

专业：信息安全、法学双学位班

指导教师：张建忠、徐敬东

2023 年 11 月 13 日

目录

一、 实验内容说明

(一) 通过编程获取 IP 地址与 MAC 地址的对应关系

实验要求如下:

1. 在 IP 数据报捕获与分析编程实验的基础上，学习 NPcap 的数据包发送方法。
2. 通过 NPcap 编程，获取 IP 地址与 MAC 地址的映射关系。
3. 程序要具有输入 IP 地址，显示输入 IP 地址与获取的 MAC 地址对应关系界面。界面可以是命令行界面，也可以是图形界面，但应以简单明了的方式在屏幕上显示。
4. 编写的程序应结构清晰，具有较好的可读性。

二、 实验准备

(一) 下载、配置 NPcap

需要下载 NPcap，注意，必须下载 NPcap SDK，它提供必要的函数库。使用 VS2019 时，需要进行必要的配置。

1. 添加 pcap.h 包含文件，即 include ”pcap.h”
2. 添加包含文件目录，包含 npcap sdk 提供的函数
3. 添加库文件目录，在附加库目录下添加 npcap sdk 提供的函数

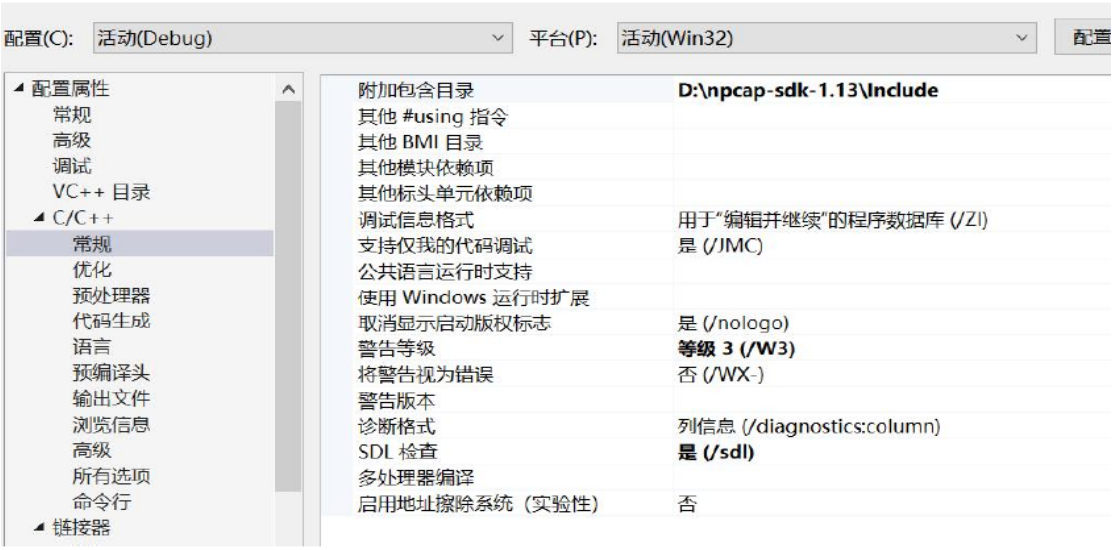

图 1: 添加包含文件

4. 添加链接时使用的库文件

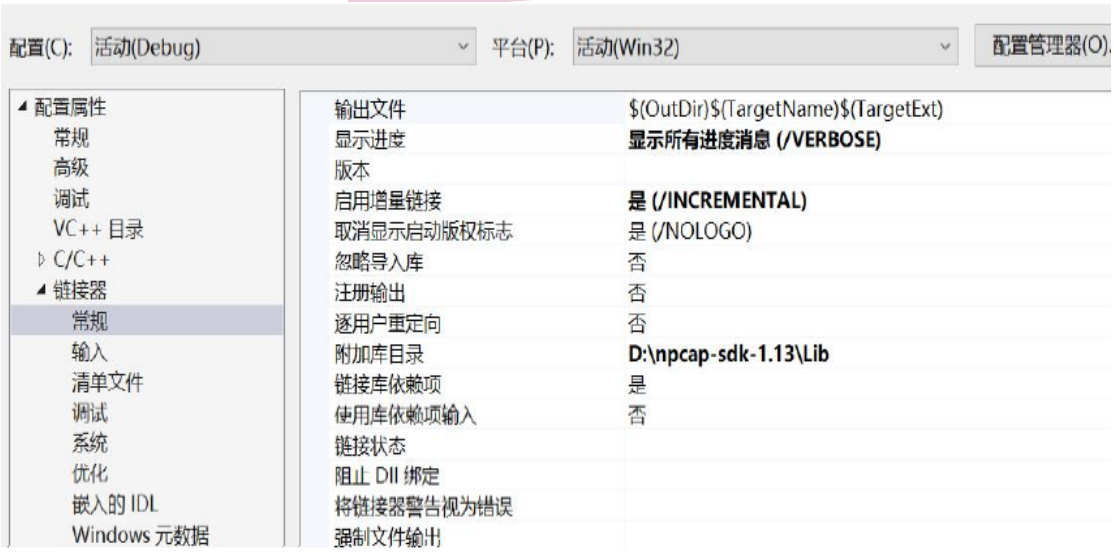

图 2: 添加库文件

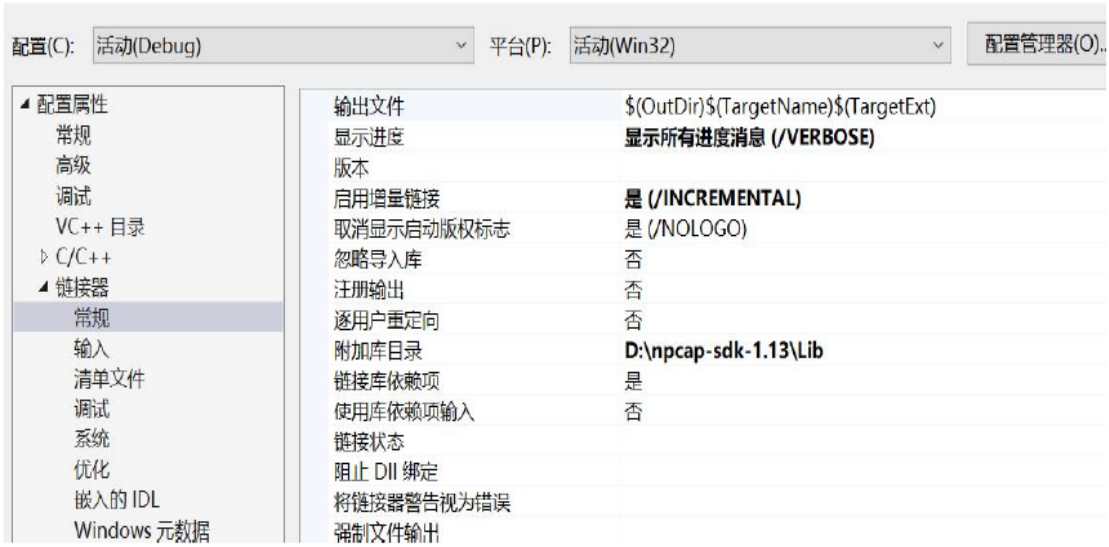

图 3: 添加链接文件

(二) 学习相关协议细节

1. 老师课上给出类定义模版

2. 学习基本过程定义，参考 https://dandelioncloud.cn/article/details/1560542885714305025

3. 学习基本过程对应结果展示,参考 https://blog.csdn.net/lyshark_csdn/article/details/126688509

4. 学习 ARP 协议，参考 https://blog.csdn.net/dinghuan6053/article/details/101878376

4. 参考 RTFM。

三、 实验过程

(一) 实验核心思路

本次实验参考《计算机网络技术与应用》进行实验设计，主要分为三步：获取本机的 IP 和 MAC 的映射关系、向以太网下其他主机发送 ARP 请求数据包、捕获 ARP 响应数据包。

1. 获取本机的 IP 和 MAC 的映射关系

(1) 需要获取本机的接口设备 [lab2 完成]。获取本机安装的网络接口和接口上绑定的 IP 地址：利用 WinPcap 提供的 cap_findalldevs_ex() 函数获取本机的接口设备列表，从而获得本机网络接口及其接口上绑定的 IP 地址。

(2) 向本地网卡发送 ARP 请求。本地主机伪造一个远端主机，采取广播的方式发送一个 ARP 请求报文，该请求报文请求本机网络接口上绑定的 IP 地址与 MAC 地址的对应关系。本地主机一旦获取该 ARP 请求，做出响应。[使用 WinPcap 的 pcap_sendpacket() 函数实现数据包的发送]

(3) 捕获本机的 ARP 响应。针对目的网卡进行捕获，对报文内容进行筛选：FrameType-0806 是 ARP 协议，FrameType-0002 是 ARP 响应，响应 IP 等于请求 IP，从而得到本机网络接口卡的 MAC 地址。

2. 向以太网下其他主机发送 ARP 请求数据包

得到本机网络接口的 MAC 地址和其上绑定的 IP 地址后，使用上一步获取到的主机网卡的 IP、MAC 地址，重新组装 ARP 数据包，模拟本地网卡发送 ARP 请求数据包，请求以太网中其他主机的 IP 地址与 MAC 地址的对应关系。使用 WinPcap 的 pcap_sendpacket() 函数发送数据包。

3. 捕获 ARP 响应数据包

针对目的网卡进行捕获，对报文内容进行筛选：FrameType-0806 是 ARP 协议，FrameType-0002 是 ARP 响应，响应 IP 等于请求 IP，从而得到本机网络接口卡的 MAC 地址。

函数调用逻辑如下：

针对本地网卡：伪造洪泛 ARP 请求数据包，捕获网卡的 ARP 响应数据包，解析获得 MAC 地址。

针对其他主机：通过本地 IP 和 MAC 对应关系，伪造对其他主机的 ARP 请求数据包，捕获 ARP 响应数据包，解析获得其他主机 IP 和 MAC 对应关系。

(二) ARP 协议

通过查阅资料，在以太网协议中，统一局域网内的主机之间要想通信必须知道彼此的 MAC 地址，而 TCP/IP 协议中，网络层和传输层只关心目标主机的 IP 地址，这就导致在以太网中使用 IP 协议时，数据链路层的以太网协议接到上层 IP 协议提供的数据中，只包含目的主机的 IP 地址。于是需要一种方法，根据目的主机的 IP 地址，获得其 MAC 地址。这就是 ARP 协议要做的事情。所谓地址解析（address resolution）就是主机在发送帧前将目标 IP 地址转换成目标 MAC 地址的过程。

当发送主机与目的主机不在同一局域网时，即便知道对方的 MAC 地址，两者也不能直接通信，必须经过路由转发才可以。所以此时，发送主机通过 ARP 协议获得的将不是目的主机的真实 MAC 地址，而是一台可以通往局域网外的路由器的 MAC 地址。于是此后发送主机发往目的主机的所有帧，都将发往该路由器，通过它向外发送。这种情况称为委托 ARP 或 ARP 代理（ARP Proxy）。在点对点链路中不使用 ARP，实际上在点对点网络中也不使用 MAC 地址，因为在此类网络中分别已经获取了对端的 IP 地址。

ARP 工作流程如下：

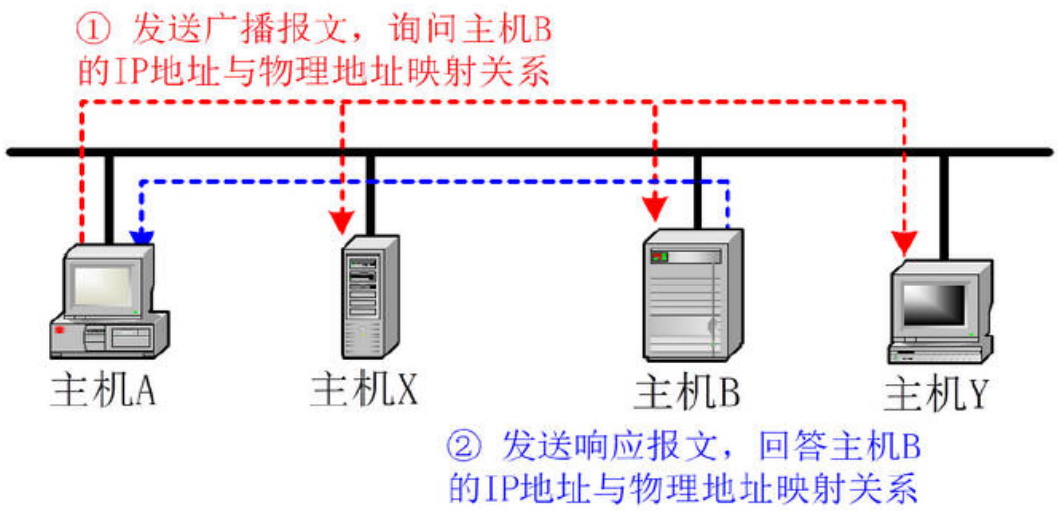

图 4: ARP 工作流程

ARP 报文格式及意义如下：

图 5: ARP 报文格式

- 硬件类型：以太网接口类型为1
- 协议类型：IP协议类型为080016
- 操作：ARP请求为1，ARP应答为2
- 硬件地址长度：MAC地址长度为6B
- 协议地址长度：IP地址长度为4B
- 源MAC地址：发送方的MAC地址
- 源IP地址：发送方的IP地址
- 目的MAC地址：ARP请求中该字段没有意义；ARP响应中为接收方的MAC地址
- 目的IP地址：ARP请求中为请求解析的IP地址；ARP响应中为接收方的IP地址

图 6: ARP 各字段意义

(三) 实验代码分析

1. 设备获取

设备获取同上一次实验。我们获得网卡列表后，选择其中一个网卡。同时获得该网卡的 IP 地址。最终取得的设备，存放在 alldevs 中。adddevs 中每个元素都是 pcap_if_t 结构。

```
pcap_findalldevs_ex(PCAP_SRC_IF_STRING, //获取本机的接口设备
                    NULL, //无需认证
                    &alldevs,//指向设备列表首部
                    errbuf//出错信息保存缓冲区
          )
```

2. 获取本机 MAC 地址

首先根据 ARP 报文格式要求设计 ARP 报文:

```
typedef struct ARPFrame_t {//IP首部
        FrameHeader_t FrameHeader;
        WORD HardwareType;
        WORD ProtocolType;
        BYTE HLen;
        BYTE PLen;
        WORD Operation;
        BYTE SendHa[6];
        DWORD SendIP;
        BYTE RecvHa[6];
        DWORD RecvIP;
}ARPFrame_t;
```

为了获得选取网卡的 MAC，需要先发送 ARP 请求，进而捕获目的网卡的响应，从而获得其 MAC 地址。发送 ARP 请求，使用了 WinPcap 提供的 pcap_sendpacket() 函数。

其中，pcap_sendpacket() 函数有三个参数，p 指定了该函数需要通过哪块网卡发送数据包，该参数为一个指向 pcap_t 结构的指针，一般是调用 pcap_open() 函数成功后的返回值。buf 指向了需要发送的数据包，而 size 指定发送数据包的大小。对于第一个参数，我们使用指定获取映射关系的目标网卡，第二个参数是 ARP 报文。因此我们需要先完成 ARP 报文的编写。

```
int pcap_sendpacket {
    pcap_t * p ,
    u_char buf ,
    int size
}
```

由于 ARP 请求是广播的，因而我们要将帧首的目的 MAC 设置为广播地址，即 0xffffffffffff，源 MAC 随意设置，帧类型赋值 0x0806，即 ARP 类型。ARP 帧中，硬件类型设置为 0x0001，即以太网；协议类型设置为 0x0800，即 IP，硬件地址长度与协议地址长度设置为 6 和 4，操作设置为 0x0001，即 ARP 请求。ARP 报文中的源 mac、ip 地址可以任意设置，这样我们发送 ARP 请求的主机实际模拟的是一个远程主机，而目的 mac 地址设置为 0，目的 IP 地址设置为请求的 IP 地址，在本实验，我们设置为请求网卡所对应的 IP 地址。

接着在循环中，先是调用 pcap_next_ex() 函数，再调用前文提到的 pcap_sendpacket() 函数，捕获网络数据包，再借助预先定义好的 ARPFrame，实现捕获的数据包转换为 ARP 报文格式，从而方便后续的重要信息提取。

捕获报文后，需要对捕获到的数据包进行筛选，过滤条件为，帧类型为 ARP 且操作类型为 ARP 响应，SendIP 为发送的数据包中的 RecvIP（即目的主机的 IP 地址）。具体代码如下，在捕获后，通过设置数据包过滤条件，保证捕获到目标报文:

```
while (1)
        {
                //抓包
                pcap_pkthdr* pkt_header;
                const u_char* pkt_data;
                int capture = pcap_next_ex(p, &pkt_header, &pkt_data);//抓包
                //发送ARP
                if (pcap_sendpacket(p, (u_char*)&ARPFrame, sizeof(ARPFrame_t)
                    ) != 0) {
                        //cout << "本地模拟发送数据包失败！" << endl;
                        //pcap_freealldevs(alldevs);
                        //return 0;
                }
                if (capture == 1)
                {
                        ARPFrame_t* ARP_response1 = (ARPFrame_t*)pkt_data;//
                            转成定义好的数据结构，方便读取信息
                        //0806是ARP协议，0002是ARP响应，响应IP是请求IP
                        if ((ntohs(ARP_response1->FrameHeader.FrameType) == 0
                            x0806) && (ntohs(ARP_response1->Operation) == 0
                            x0002) && (ARP_response1->SendIP == ARPFrame.
                            RecvIP))
                        {
                                cout << "所选设备的MAC地址为：";

                                for (int i = 0; i < 6; i++)
                                {
                                        printf("%02x.", ARP_response1->
                                            FrameHeader.SrcMAC[i]);
                                        device_mac[i] = ARP_response1->
                                            FrameHeader.SrcMAC[i];
                                }
                                break;
                        }
                }
        }
```

3. 获取目的主机 MAC 地址

思路与获取本地 MAC 地址一致，从伪造主机进行 ARP 请求转变为模拟本地主机进行 ARP 请求。DesMAC 设置为广播地址；SrcMAC 和 SendHa 为前面操作获取的网卡 MAC 地址；SendIP

为网卡的 IP 地址，用上面得到的 ip 设置；RecvIP 设置为目的主机的 IP 地址，由用户输入；其他关于类型的设置均与上面相同。

```
…
ARPFrame_again.FrameHeader.SrcMAC[i] = device_mac[i];
ARPFrame_again.SendHa[i] = device_mac[i];
ARPFrame_again.SendIP = inet_addr(device_ip);
…
while(1){
    抓包…
    解析…
}
```

(四) 实验结果截图

1. 本地 IP 和 MAC 信息

首先通过 CMD 查找本地 IP 和 MAC 地址，从而针对 IP 地址进行伪造 ARP 请求，并捕获、解析对应 MAC 地址

```
无线局域网适配器 WLAN:

   连接特定的 DNS 后缀 . . . . . . . :
   描述. . . . . . . . . . . . . . . : Intel(R) Wi-Fi 6 AX200 160MHz
   物理地址. . . . . . . . . . . . . : 78-2B-46-51-17-24
   DHCP 已启用 . . . . . . . . . . . : 是
   自动配置已启用. . . . . . . . . . : 是
   IPv6 地址 . . . . . . . . . . . . : 2001:250:401:6570:ceb8:3571:81f:b575(首选)
   临时 IPv6 地址. . . . . . . . . . : 2001:250:401:6570:dc85:c057:95ce:6875(首选)
   本地链接 IPv6 地址. . . . . . . . : fe80::4dc2:6638:5b19:2c92%19(首选)
   IPv4 地址 . . . . . . . . . . . . : 10.136.17.244(首选)
   子网掩码 . . . . . . . . . . . . : 255.255.128.0
   获得租约的时间  . . . . . . . . . : 2023年11月13日 14:03:29
   租约过期的时间  . . . . . . . . . : 2023年11月13日 20:03:29
   默认网关. . . . . . . . . . . . . : fe80::865b:12ff:fe5e:360b%19
                                       10.136.0.1
   DHCP 服务器 . . . . . . . . . . . : 10.136.0.1
   DHCPv6 IAID . . . . . . . . . . . : 158870342
   DHCPv6 客户端 DUID  . . . . . . . : 00-01-00-01-2C-D8-60-9D-78-2B-46-51-17-24
   DNS 服务器  . . . . . . . . . . . : 222.30.45.41
                                       202.113.16.41
   TCPIP 上的 NetBIOS  . . . . . . . : 已启用
```

图 7: 本地 ip 和 MAC 地址

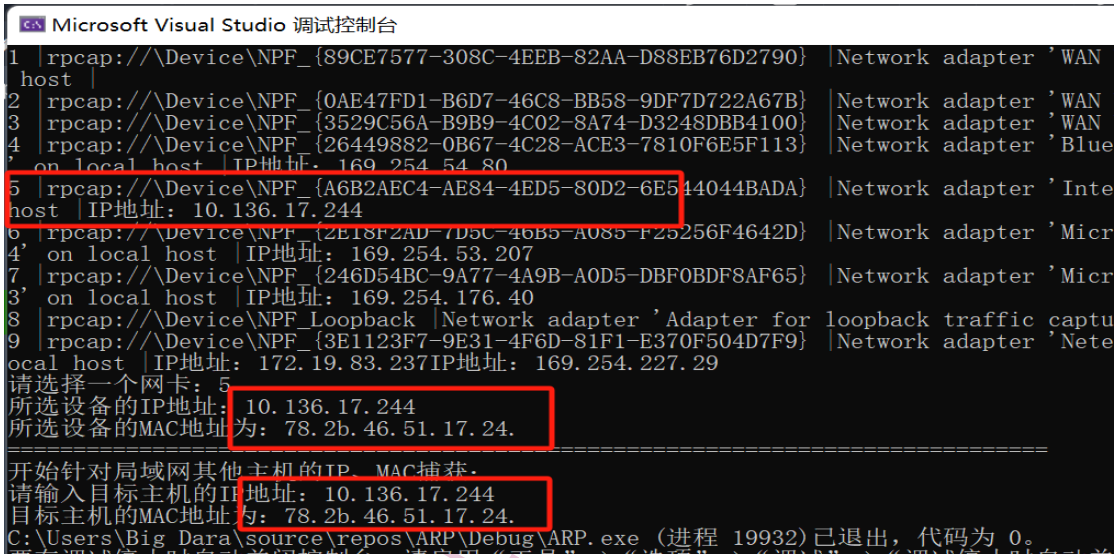
Microsoft Visual Studio 调试控制台
1 |rpcap://\Device\NPF_{89CE7577-308C-4EEB-82AA-D88EB76D2790} |Network adapter 'WAN
host |
2 |rpcap://\Device\NPF_{0AE47FD1-B6D7-46C8-BB58-9DF7D722A67B} |Network adapter 'WAN
3 |rpcap://\Device\NPF_{3529C56A-B9B9-4C02-8A74-D3248DBB4100} |Network adapter 'WAN
4 |rpcap://\Device\NPF_{26449882-0B67-4C28-ACE3-7810F6E5F113} |Network adapter 'Blue
' on local host |IP地址: 169.254.54.80
5 |rpcap://\Device\NPF_{A6B2AEC4-AE84-4ED5-80D2-6E544044BADA} |Network adapter 'Inte
host |IP地址: 10.136.17.244
6 |rpcap://\Device\NPF_{2E18F2AD-7D5C-46B5-A085-F25256F4642D} |Network adapter 'Micr
4' on local host |IP地址: 169.254.53.207
7 |rpcap://\Device\NPF_{246D54BC-9A77-4A9B-A0D5-DBF0BDF8AF65} |Network adapter 'Micr
3' on local host |IP地址: 169.254.176.40
8 |rpcap://\Device\NPF_Loopback |Network adapter 'Adapter for loopback traffic captu
9 |rpcap://\Device\NPF_{3E1123F7-9E31-4F6D-81F1-E370F504D7F9} |Network adapter 'Nete
ocal host |IP地址: 172.19.83.237IP地址: 169.254.227.29
请选择一个网卡: 5
所选设备的IP地址: 10.136.17.244
所选设备的MAC地址为: 78.2b.46.51.17.24.
开始针对局域网其他主机的IP、MAC捕获:
请输入目标主机的IP地址: 10.136.17.244
目标主机的MAC地址为: 78.2b.46.51.17.24.
C:\Users\Big Dara\source\repos\ARP\Debug\ARP.exe (进程 19932)已退出, 代码为 0。

图 8: 本地模拟 ARP

2. 同一局域网下其他主机的 IP 和 MAC 信息

将本地电脑与其他电脑通过 802.11 协议连接到同一局域网下，模拟 ARP 请求数据包，捕获到的 MAC 地址与目的主机 MAC 地址一致，验证成功。

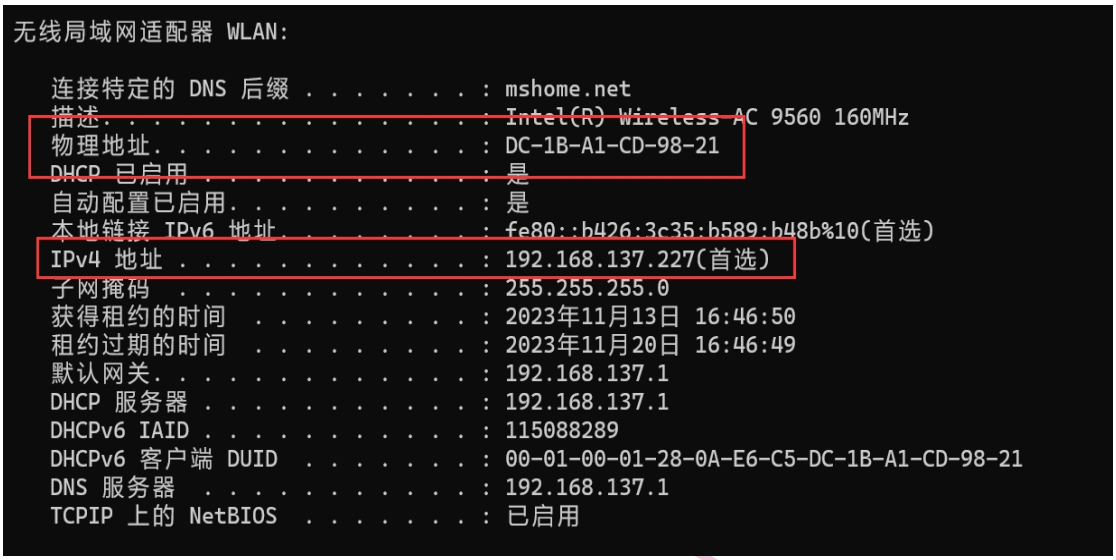

图 9: 其他主机 IP 和 MAC config

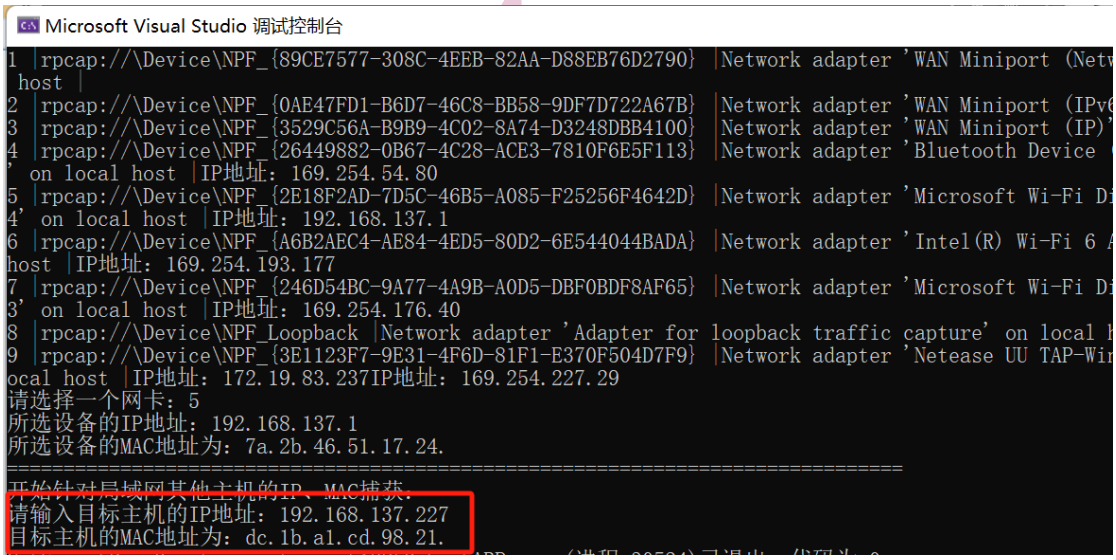

图 10: 捕获其他主机 MAC 地址

四、 总结

通过本次实验，使用 NPcap 模拟 ARP 数据包发送并捕获，了解相关过程调用过程调用。通过阅读 RTFM 增加了对网络接口、协议的了解。