

网络攻防技术基础课程实验报告

|  |  |
| --- | --- |
| 实验名称： | 实验三  网络嗅探编程实现 |
| 实验日期： | 2024-3-26 |
| 实验地点： | 西部片区4号楼301 |
|  | |
| 学号： | 33920212204567 |
| 姓名： | 任宇 |
| 专业年级： | 软工2021级 |
| 学年学期： | 2023-2024学年第二学期 |

1. **实验目的**

* 熟悉常用网络嗅探编程使用的库，如Scapy、libpcap库等。

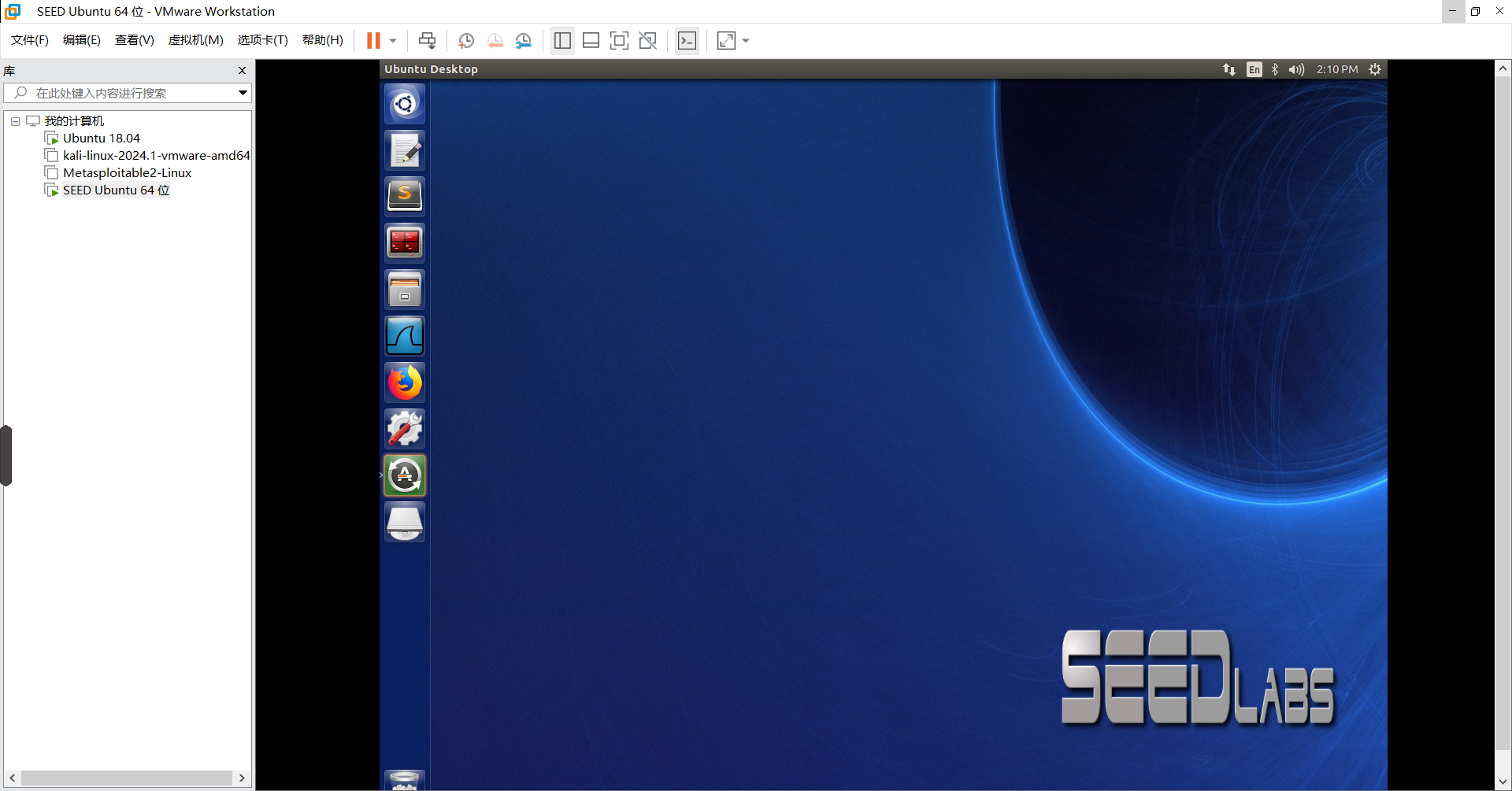
1. **实验用到的软件或工具**

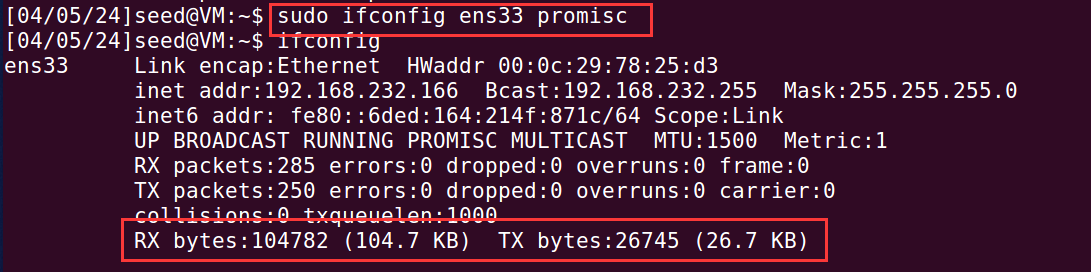
虚拟机软件：VMware Workstation Pro 17

虚拟机系统：SEED Ubuntu-16.04-32bit、Ubuntu18.04

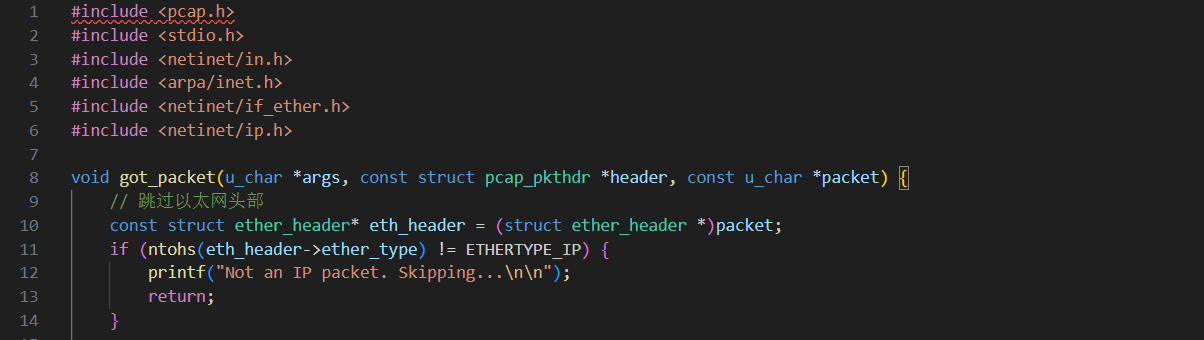
网络环境：两个虚拟机环境配置在同一NAT网络中

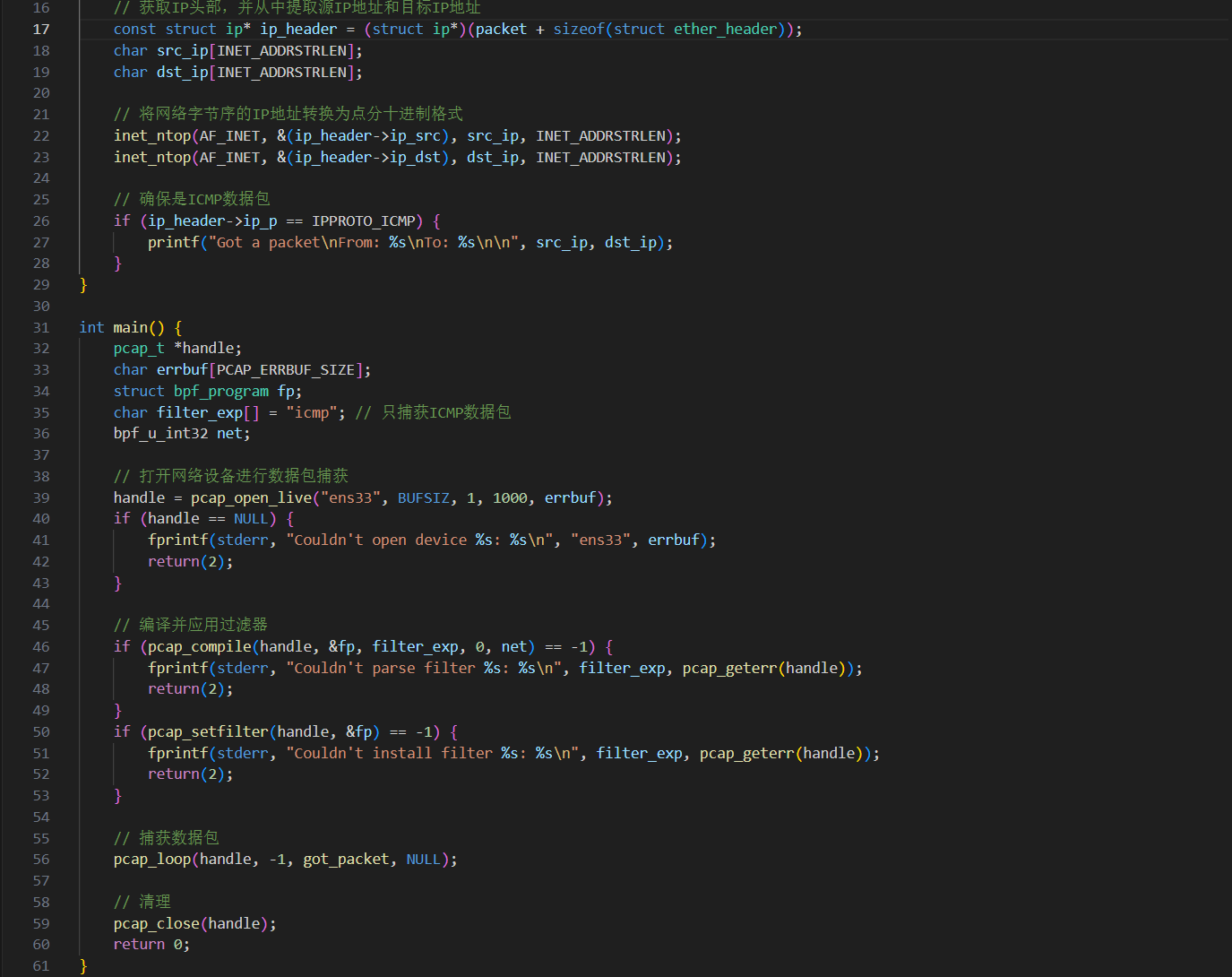
1. **实验过程**
2. **参考sniff.c，编写网络包嗅探程序：**
3. 虚拟机软件中打开SEED镜像，网卡设置为混杂模式：



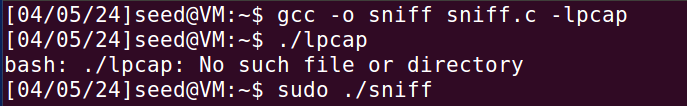


1. Libpcap实现靶机的ICMP嗅探，程序如图：

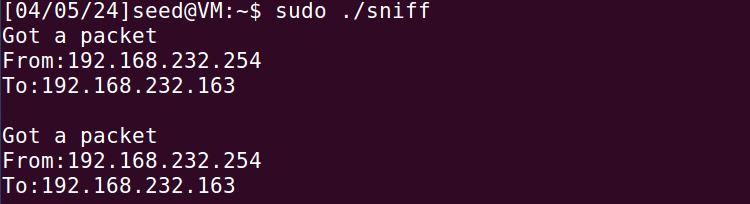




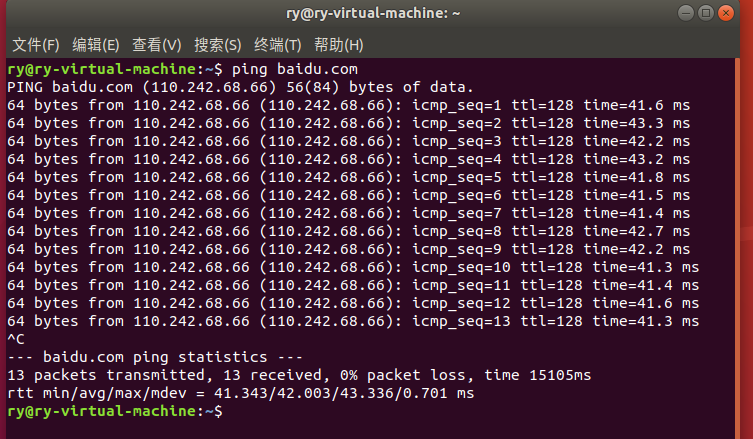
1. 编译C代码：gcc –o sniff sniff.c –lpcap：



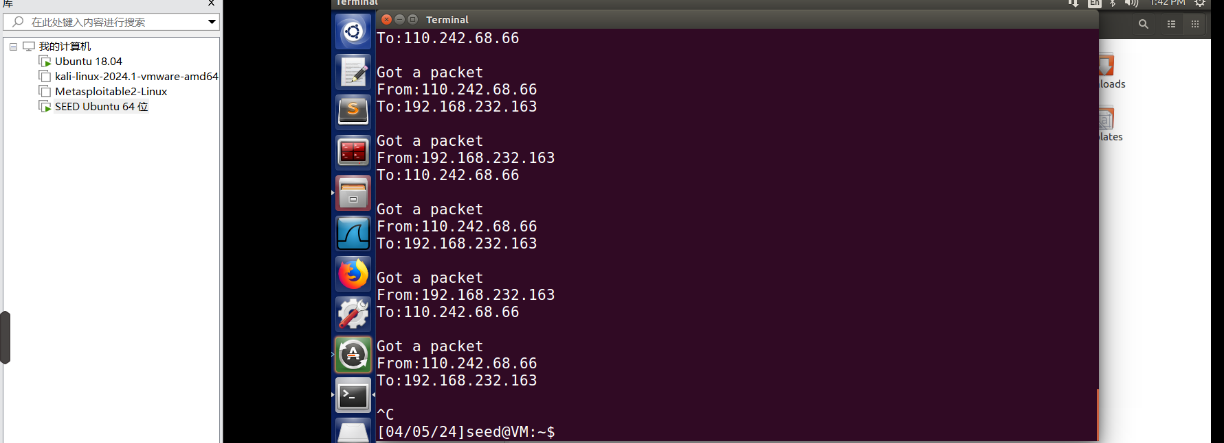
1. 开始执行抓包：sudo ./sniff：



1. 开启靶机，执行ping命令：



1. 观察攻击机的结果，如下图所示，可见成功捕获并分析出靶机发出和接收的ICMP包：

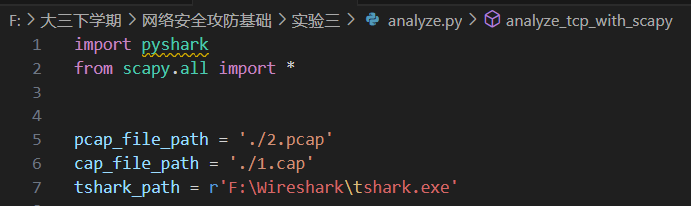


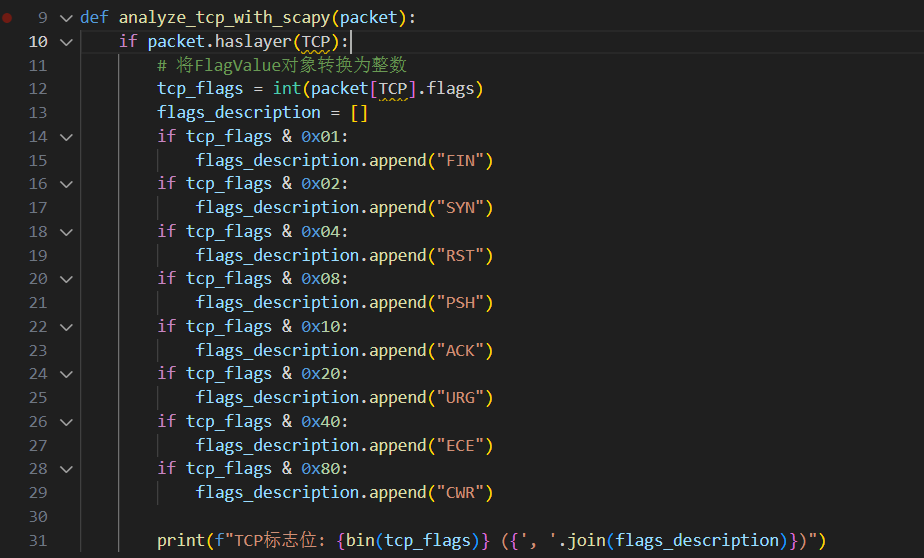
1. **简易流量分析系统：**

对给定的协议流量报文能够解析出相关协议类型及相关字段，比如：http URL、Header等等：

1. 使用python语言以及pyshark、scapy两个库实现题目要求：

设计思路：利用pyshark进行基本的网络数据包解析，获取数据包的IP、TCP/UDP、HTTP等层的信息。当遇到特定协议（如TCP或DHCP）时，调用scapy进行更深入的分析，比如解析TCP标志位和DHCP选项。这样可以结合两个工具的优点：pyshark用于高效的初步分析，而scapy用于更复杂或定制化的数据包处理，从而实现一个全面的网络数据包分析工具，代码实现如下：

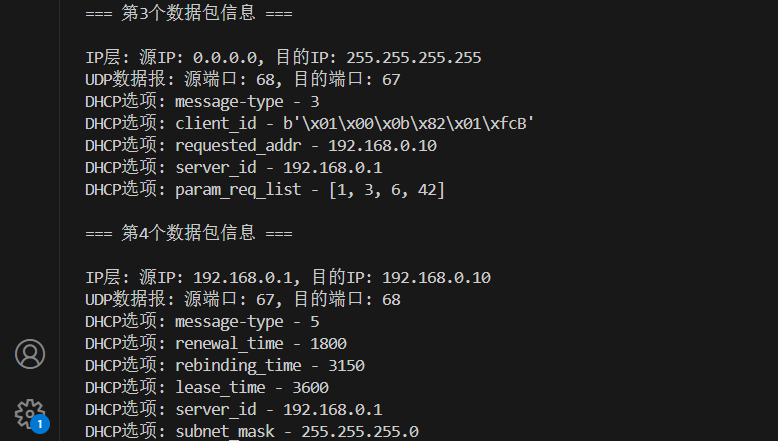






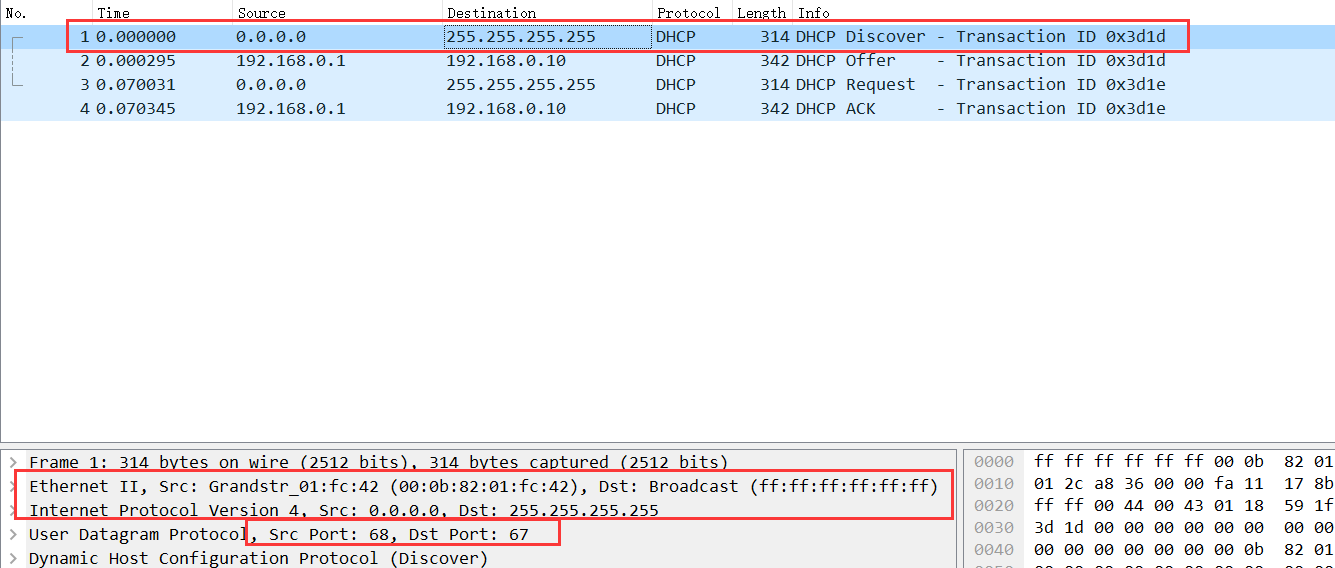
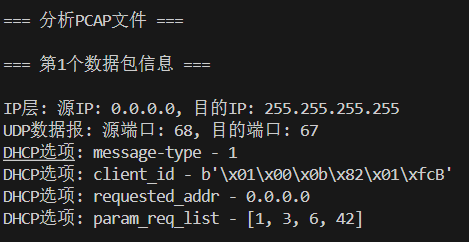
1. 解析教学网站上的测试网络包文件：





使用Wireshark打开两个网络包，检验分析是否正确：

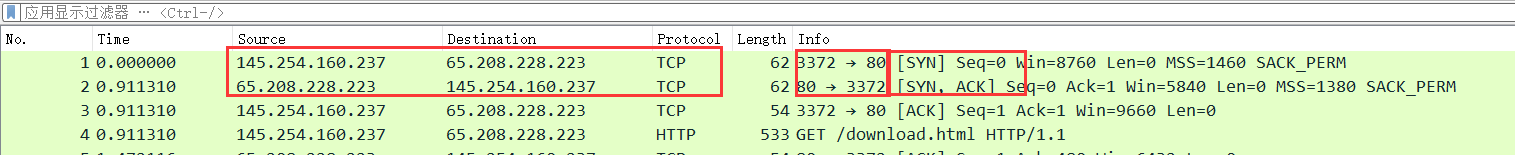
* 对于.pcap文件，可见四条数据都被解析，选取第一条与wireshark上的解析对照，可见解析正确。



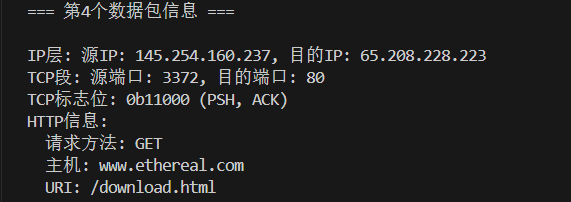
* 对于.cap文件，全部43条数据都被解析，选取部分数据包与wireshark上的解析对照，可见解析正确。

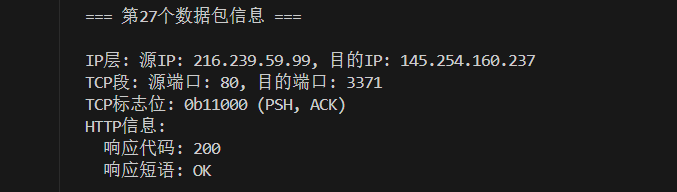
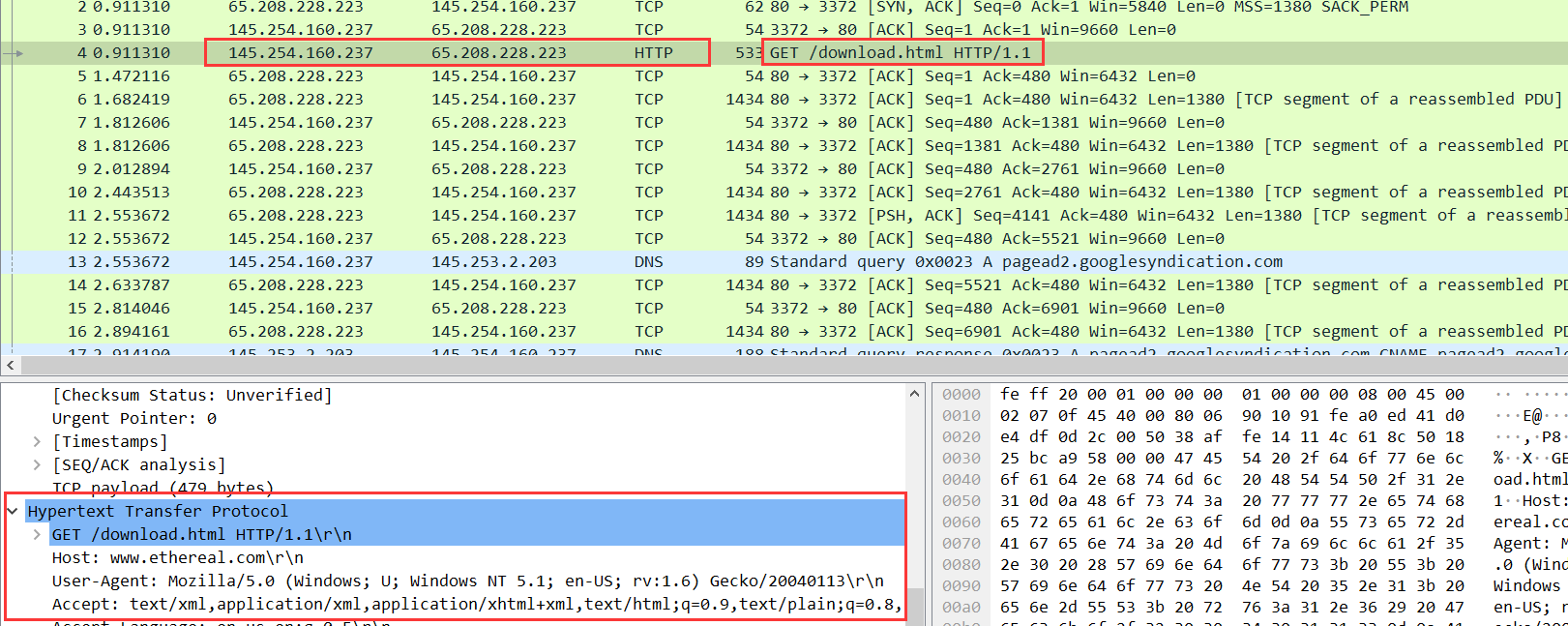
TCP报文：

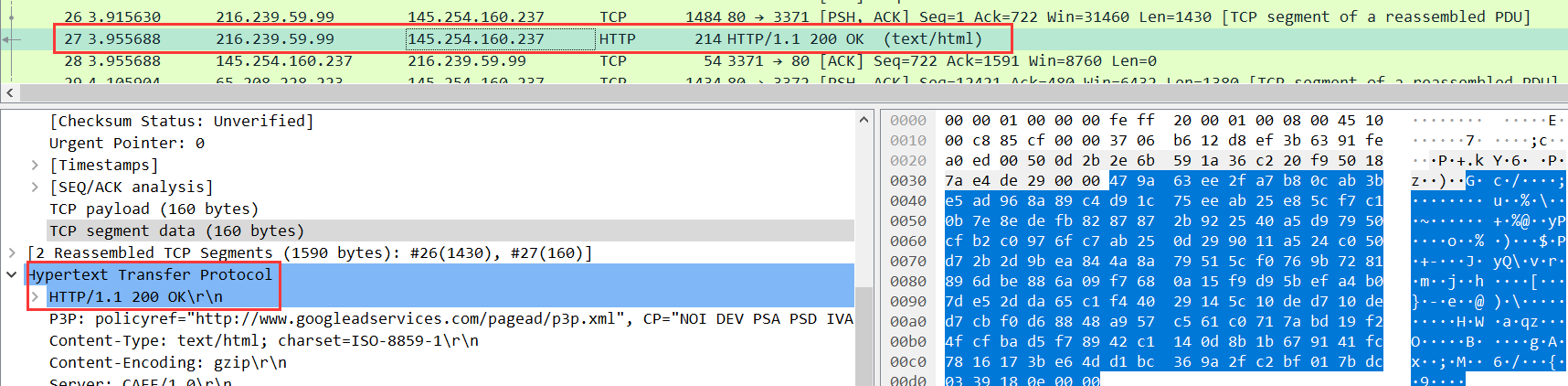




HTTP报文：







1. **实验思考及心得**

通过这次实验，我深入了解和熟悉了两个强大的网络嗅探和数据包分析库：Scapy和libpcap。通过编写网络嗅探程序，我实践了网络数据包捕获的基本原理和方法。特别是在实现简易流量分析系统的过程中，我学会了如何结合使用pyshark和Scapy两个库，利用各自的优势来对网络流量进行深入分析。这不仅提升了我的编程技能，也加深了我对网络协议和网络安全的理解。