

网络安全实验报告

题 目 捕包软件的使用与实现

专 业 网络安全

学 号 1170300728

学 生 汤添凝

指 导 教 师 王彦

**一、实验目的**

理解捕包程序捕包过程，可以自己编程捕包并从数据包中解析出需要的信息。

# 二、实验内容

1. winpcapt或libpcap的安装
2. winpcapt或libpcap编程过程，编写程序捕获实验一的数据包。还原四元组（源目的IP源目的端口），和文件内容，写到一个文件里（默认文件），如果需要也可输出到特定文件（通过配置文件或命令行参数决定，屏幕也是特定文件之一）（还记得PV么？）

# 三、实验过程

**利用 libpcap 编写捕包软件实验基本信息：**

实验环境：Ubuntu16.04 x64

编程语言：Python语言

## 1. 需求分析

本程序需要运用 libpcap 来捕获本机数据包，并获取数据包中的四元组，将其展示给

用户。

程序功能：

(1)捕获本机数据包（可以自定义过滤条件）；

(2)逐层解析数据包，获得 IPv4 数据包的源 ip、目的 ip、源端口、目的端口；

(3)将上述四元组写入文件（每次运行程序都新生成一个文件）。

## 2. 数据结构设计

由于是逐层解析以太网数据帧，所以需要准备至少三种数据结构：以太网数据帧头、

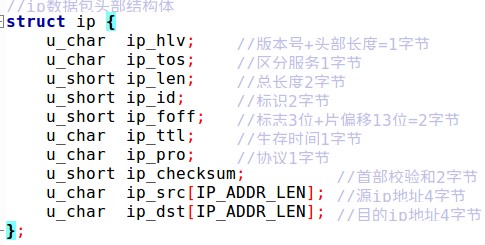
IPv4 数据报头、传输层报文头。具体如下：

数据结构的定义原则：1 字节数据定义为 u\_char，2 字节数据定义为 u\_short，其他 2 的倍数字节的数据（MAC 地址和 IP 地址）定义为 u\_char 数组（TCP 的序列号和 ack 定义为 u\_int，因为它们的表现形式就是一个数字，但地址我们通常是一个一个字节分开解析的）。

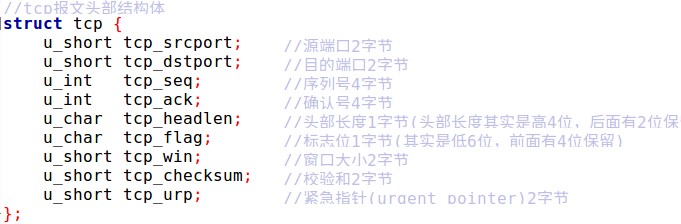
1. 以太网数据帧头



1. IPv4 数据报头



1. TCP 报文头



1. UDP 报文头



## 3. 主函数设计

第一步：使用 sniffer = pcap.pcap(name=get\_first\_iface())函数来获取网络设备。

第二步：循环从sniffer中提取packet\_time, packet\_data，以便进行包分析。

第三步：如果有过滤条件的话（作为程序运行参数读入），设置过滤条件。

第四步：生成本次捕包的 txt 文件，逐条写入包内容。

## 4. 运行

使用下列命令运行：

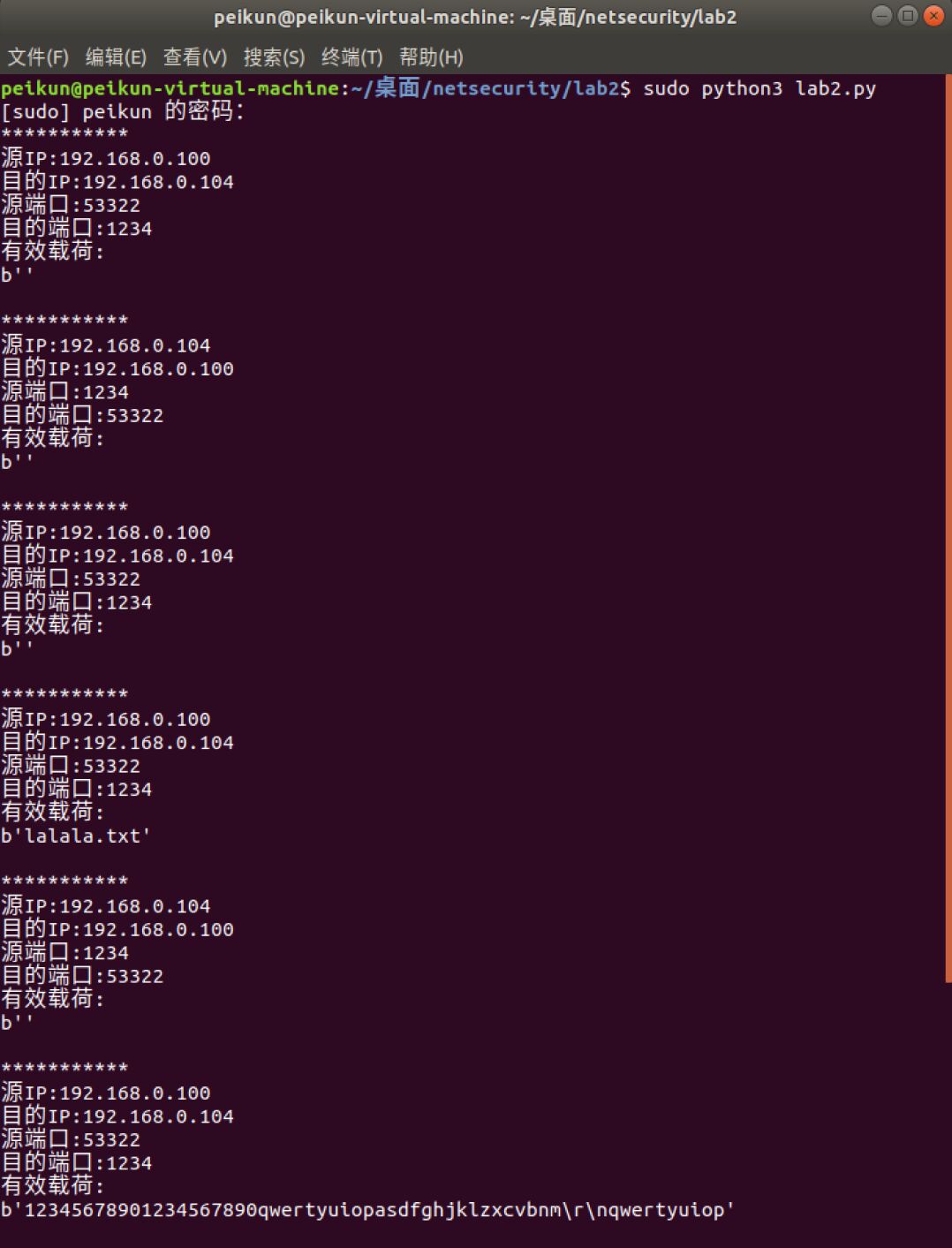
**sudo** python lab2.py

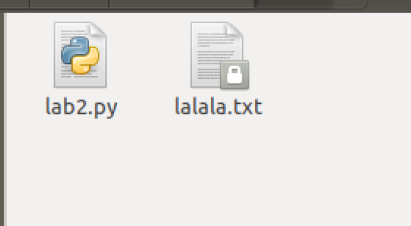
捕包程序在运行时需要 root 权限，否则无法正常打开。

# 四、实验结果

1. **利用 libpcap 编写捕包软件**

以下为捕包时文件中部分输出：





# 五、心得体会

本次实验内容精炼到位，有一定难度与复杂度，但在同学以及个人自学的多重努力下较为轻松地完成了实验。

本次实验为“补包实现”，以直接的实际应用，使我们巩固了socket网络编程，教会了我们如何用libpacap捕获数据包并分析，效果出众。在实验逐渐进行的同时，开拓了计算思维，以至于在测试效果的时候能够轻松地发现失误的部分并加以修正。我想这大概就是身为“计算机科学与技术学院”的学生相比其他专业学生对于计算机认识更深入的地方之一了。