**计算机网络协议开发实验lab3**

**实验报告**

**计算机科学与技术系**

姓名：刘博

学号：141220065

1. **实验目的：**

理解协议的逆向分析方法并掌握客户端套接字编程

1. **实验原理：**

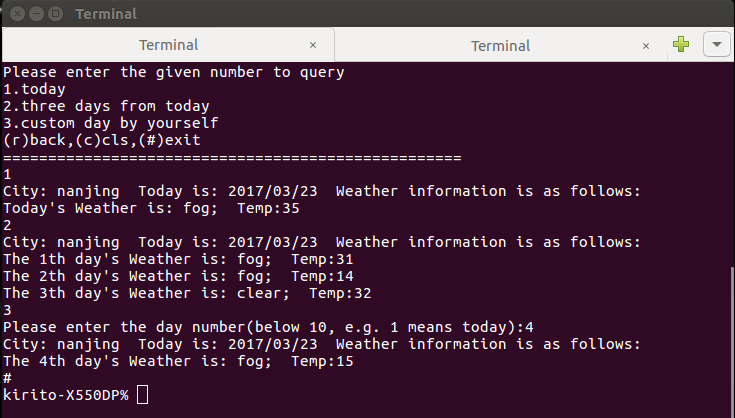
利用实验2中掌握的wireshark数据包探嗅技术逆向分析一个协议，并利用套接字编程技术重新实现该协议的客户端

1. **实验环境：**

Linux操作系统；

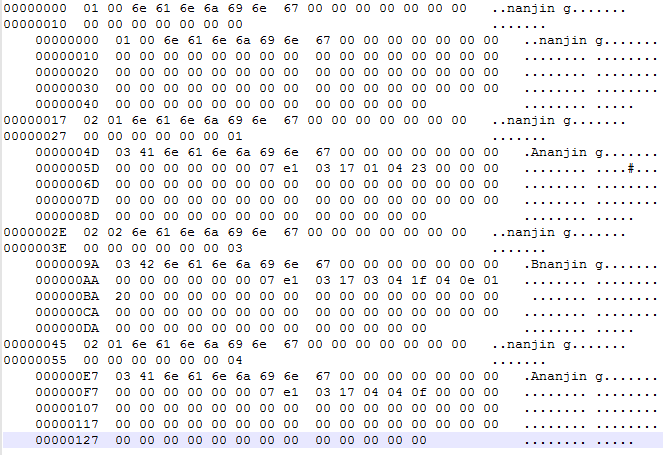
Wireshark；

1. **实验过程：**
2. 运行客户端程序，同时打开wireshark监听，抓获客户端和服务器之间交换的报文；
3. 对报文的编码进行分析，掌握报文格式；
4. 一旦分析完毕，便用C语言在Linux系统中进行套接字编程，利用同样的协议报文发送给服务器，并接受报文解析；
5. **实验结果：**
6. 利用wireshark监听客户端的报文，由于我们熟知的协议为TCP协议。所以我们在wireshark中截取TCP报文进行监听；
7. 首先打开wireshark，然后再运行客户端，在客户端中进行如下操作：



可以看到我输入了nanjing，然后分别输入1，2，3，并且在3号菜单中输入了4；得到了服务器反馈如上；

1. 此时停止wireshark，并抓取TCP流进行保存：得到如下结果（保存在tcp文件中）：



可以看到发送报文中，当我输入南京时，报文头部为：01 00 nanjing 00 …… 00

即协议格式为：01 00 城市 00 ……. 00

再观察收到的报文中，当服务器收到发送报文后，回复的报文头部为：01 00 nanjing 00 …….. 00

即协议格式为：01 00 城市 00 ……. 00

当我进入城市day选择界面后：

输入1后，发送报文为：02 01 nanjing 00 …….. 01

即协议格式为： 02 01 城市 00 ……. 01

然后接受到1的服务器回复的报文为：03 41 nanjing 00 ….. 00 07 e1 03 17 01 04 23 00 …….. 00

可以分析到：07 e1 03 17为2017.03.17

01 为分割

04 23为气候和温度的编码

经过多次尝试后，得到气候的编码如下：

00 shower

01 clear

02 cloudy

03 rain

04 fog

输入2后，发送报文为：02 02 nanjing 00 …….. 03

即协议格式为：02 02 城市 00 ……. 03（初步分析03是天数）

然后接受到2的服务器回复的报文为：03 42 nanjing 00 ……. 00 07 e1 03 17 03 04 1f 04 0e 01 20 00 ………. 00

可以分析到：07 e1 03 17 为2017.03.17

03为分割（初步分析03是天数）

04 1f

04 0e

01 20

分别为3组气候和温度的编码

输入3后，客户端会继续接受一个额外的参数（第几天），并把这个参数放入报文中：

报文为：02 01 nanjing 00 ……… 04

可以看出这个报文头部和输入1的报文头部很像，只有最后一个字节不同，所以最后一个字节应该是第几天的参数

当服务器接收到该报文后，回复的报文如下：

03 41 nanjing 00 ……. 00 07 e1 03 17 04 04 0f 00 ……… 00

可以看出报文格式为：03 41 城市 00 ……… 00

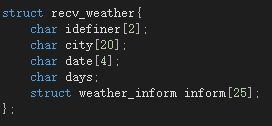
07 e1 03 17 为时间2017 03 17

04 为天数

04 0f为气候和温度的编码

4．分析完成后，我们就可以根据分析出的报文格式，进行套接字编程仿造其发送的报文。

其中我们可以建立如下的数据结构：



表示我们会受到的数据包的信息；

然后我们根据实验手册的附录，首先建立socket套接字，然后connect to IP地址，就可以向服务器发送请求报文（send）函数，然后我们通过接受（recv）函数接受到回复的报文并放入上述结构中，然后按照其中的每一个成员进行输出即可。

1. **反思：**

在上述实验程序中，我发现其实原程序使用了一个存储城市的公共数组，导致每次城市数组没有清除上一次的结果然后重写，所以会产生城市名过长时，第二次的城市名中仍会含有第一次城市名的部分

我发现对于标准输入函数scanf，对于换行符的读取不是很准确。会导致下一次的读入存在问题，所以每次我都把scanf和getchar合并使用