

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

一、 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

二、 实验原理：

利用实验 2 中掌握的 wireshark 数据包嗅探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

三、 实验环境：

Linux 操作系统；

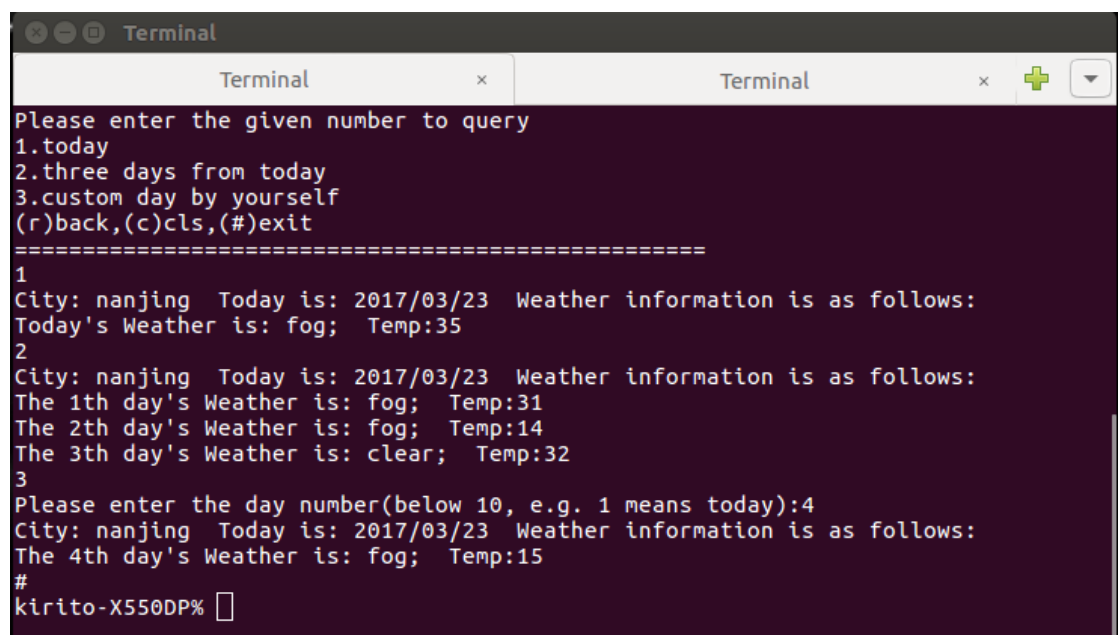
Wireshark；

四、 实验过程：

1. 运行客户端程序，同时打开 wireshark 监听，抓获客户端和服务端之间交换的报文；
2. 对报文的编码进行分析，掌握报文格式；
3. 一旦分析完毕，使用 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；

五、 实验结果：

1. 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；
2. 首先打开 wireshark，然后再运行客户端，在客户端中进行如下操作：



```
Terminal
Please enter the given number to query
1.today
2.three days from today
3.custom day by yourself
(r)back,(c)cls,(#)exit
=====
1
City: nanjing Today is: 2017/03/23 Weather information is as follows:
Today's Weather is: fog; Temp:35
2
City: nanjing Today is: 2017/03/23 Weather information is as follows:
The 1th day's Weather is: fog; Temp:31
The 2th day's Weather is: fog; Temp:14
The 3th day's Weather is: clear; Temp:32
3
Please enter the day number(below 10, e.g. 1 means today):4
City: nanjing Today is: 2017/03/23 Weather information is as follows:
The 4th day's Weather is: fog; Temp:15
#
kirito-X550DP% 
```

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；

3. 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）：

```

00000000 01 00 6e 61 6e 6a 69 6e 67 00 00 00 00 00 00 00 ..nanjin g.....
00000010 00 00 00 00 00 00 00 00 .....
00000000 01 00 6e 61 6e 6a 69 6e 67 00 00 00 00 00 00 00 ..nanjin g.....
00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000017 02 01 6e 61 6e 6a 69 6e 67 00 00 00 00 00 00 00 ..nanjin g.....
00000027 00 00 00 00 00 00 00 01 .....
0000004D 03 41 6e 61 6e 6a 69 6e 67 00 00 00 00 00 00 00 .Ananjin g.....
0000005D 00 00 00 00 00 00 00 07 e1 03 17 01 04 23 00 00 00 ..... #...
0000006D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000007D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000008D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000002E 02 02 6e 61 6e 6a 69 6e 67 00 00 00 00 00 00 00 ..nanjin g.....
0000003E 00 00 00 00 00 00 00 03 .....
0000009A 03 42 6e 61 6e 6a 69 6e 67 00 00 00 00 00 00 00 .Bnanjin g.....
000000AA 00 00 00 00 00 00 00 07 e1 03 17 03 04 1f 04 0e 01 .....
000000BA 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000000CA 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000000DA 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000045 02 01 6e 61 6e 6a 69 6e 67 00 00 00 00 00 00 00 ..nanjin g.....
00000055 00 00 00 00 00 00 00 04 .....
000000E7 03 41 6e 61 6e 6a 69 6e 67 00 00 00 00 00 00 00 .Ananjin g.....
000000F7 00 00 00 00 00 00 00 07 e1 03 17 04 04 0f 00 00 00 .....
00000107 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000117 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000127 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

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：

```
struct recv_weather{
    char idefiner[2];
    char city[20];
    char date[4];
    char days;
    struct weather_inform inform[25];
};
```

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。

六、 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