

计算机网络协议开发实验报告

201220064 吴浩然

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、实验内容

对抓包文件的分析得出：客户端发送的数据包大小为33字节，第一个字节说明该包查询城市还是查询天气，第二个字节说明该包查询天气的类型，之后30个字节存储城市名，最后一字节代表查询的日期，若查询天气的类型为0x02，则代表查询未来对应天数。

服务器端发送的数据包大小为127字节，第一个字节指示该包返回城市、天气、查询城市失败或查询天气失败，第二个字节在查询天气成功时代表返回的是一天或多天天气，之后30字节为城市名，后跟2字节的年份、1字节的月份和具体日期，之后的1字节在查询一天天气时代表具体哪一天，查询多天天气时代表查询的天数，最后的90字节，每两个字节分别代表了天气和温度。

使用结构体表示如下，其中weather使用1个字节的枚举类型来表示天气，condition结构体包含1个字节的天气和1个字节的温度，使用结构体函数来对数据包字段的读取进行简化：

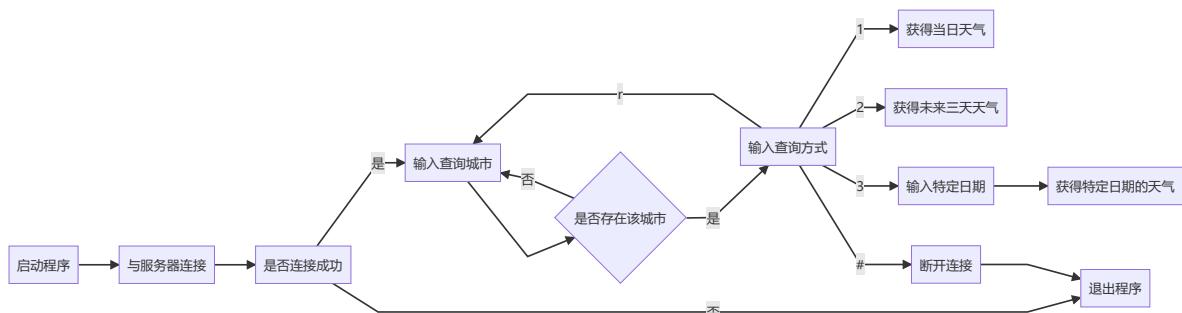
```
1 struct sendpkt {
2     uint8_t stype;
3     uint8_t qtype;
4     char cname[30];
5     uint8_t number;
6 }
7
8 enum weather : uint8_t
9 {
10     overcast = 0x00,
11     sunny,
12     cloudy,
13     rain,
14     fog,
15     rainstorm,
16     thunderstorm,
17     breeze,
18     sand_st0rm
19 };
20
21 struct condition
22 {
23     weather wea;
24     uint8_t temp;
25 };
26
27 struct recvpkt {
28     uint8_t field1;
29     uint8_t field2;
30     char cname[30];
```

```

31     uint16_t year;
32     uint8_t mon;
33     uint8_t day;
34     uint8_t num;
35     condition con[45];
36
37     recvpkt() { memset(this, 0, sizeof(*this)); }
38     bool isCitypkt() { return field1 == 0x01; }
39     bool isWorngpkt() { return field1 == 0x02; }
40     bool isweapkt() { return field1 == 0x03; }
41     bool isNoinfopkt() { return field1 == 0x04; }
42     char getType() { return (char)field2; }
43     int getYear() { return ntohs(year); }
44     int getMon() { return mon; }
45     int getDay() { return day; }
46 };

```

程序的简要设计流程图如下：



因为需要同时监视标准输入和套接字输入，所以使用select系统调用进行阻塞，使用FD_ISSET函数来对套接字的输入和标准输入进行分别处理。在设计查询流程时，因为整个流程包含在一个大循环中，所以设置整型变量menulevel来指示到达流程的哪一步骤，使用布尔变量menushow来在每个状态指示是否需要显示菜单。显示菜单的函数包含在每次循环的开始。如下所示：

```

1 void showMenu(int lv, bool need) {
2     if(!need) return;
3     if(lv == 1) showMainMenu();
4     else if(lv == 2) showSubMenu();
5     ...
6 }
7
8 int main(int argc, char** argv) {

```

```

9     int menulevel = 1;
10    bool menushow = true;
11    ...
12    while (1) {
13        showMenu(menulevel, menushow);
14        int rc = select(sockfd + 1, &recvmask, NULL, NULL, NULL);
15        ...
16    }
17 }

```

在进行标准输入后，对输入进行判断，分别处理，逻辑如下所示：

```

1  if(FD_ISSET(0,&recvmask) && wgets(query,MAXLINE) != NULL) {
2      if(strncmp(query,"#",1) == 0) {
3          ...//退出
4      }
5      else if(strncmp(query,"c",1) == 0) {
6          ...//清屏
7      }
8      else if(menulevel == 1)
9      {
10         ...//查询城市
11         sendpkt pcity = sendpkt(0x01, 0x00, query, 0x00);
12         int sdsuc = send(sockfd, &pcity, sizeof(pcity), 0);
13         ...
14     }
15     else if(menulevel == 2 && strncmp(query,"r",1) == 0)
16     {
17         ...//返回到menulevel为1
18     }
19     else if(menulevel == 2 && strncmp(query,"1",1) == 0)
20     {
21         ...//查询当日
22         int sdsuc = send(sockfd, &ptoday, sizeof(ptoday), 0);
23     }
24     else if(menulevel == 2 && strncmp(query,"2",1) == 0)
25     {
26         ...//查询未来3天
27         int sdsuc = send(sockfd, &ptoday, sizeof(ptoday), 0);
28     }
29     else if(menulevel == 2 && strncmp(query,"3",1) == 0)
30     {
31         ...//查询特定日期
32     }
33     ...//其他情况及错误处理

```

对接受到的数据包的处理为根据包的类型输出相应内容。

三、实验结果

1. 实现了查询是否有输入城市天气的功能，根据包的大小限制，已对输入字符串的长度进行检查。

```
Welcome to NJUCS Weather Forecast Program!
Please input City Name in Chinese pinyin(e.g. nanjing or beijing)
(c)cls,(#)exit
nanjing
```

```
Welcome to NJUCS Weather Forecast Program!
Please input City Name in Chinese pinyin(e.g. nanjing or beijing)
(c)cls,(#)exit
sqiwsqwsqkwjskqwjskqwjskqjskqjwsjqwsjqwjsqwjsqwjsjqwswqsjksjwqksj
your input must be less than 30 words, please try again:
```

2. 实现了查询当天天气的功能。

```
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 :2023/03/21    Weather information is as follows:
Today's Weather is: overcast; Temp:0
```

3. 实现了查询未来3天天气的功能

```
2
City :nanjing    Today is :2023/03/21    Weather information is as follows:
The 1th day's Weather is: fog; Temp:4
The 2th day's Weather is: sand_st0rm; Temp:34
The 3th day's Weather is: sand_st0rm; Temp:28
```

4. 实现了查询特定日期天气的功能，对错误的日期范围及其他输入进行了检查

```
Please enter the given number to query
1.today
2.three days from today
3.custom day by yourself
(r)back,(c)cls,(#)exit
3
Please enter the day number(below 10,e.g. 1 means today):
5
City :nanjing    Today is :2023/03/21    Weather information is as follows:
The 5th day's Weather is: overcast; Temp:0
```

```
3
Please enter the day number(below 10,e.g. 1 means today)
1000
input error, please try again:
```

5. 当查询不到城市或天气时，给出文字反馈。

6. 对错误的输入进行了检查。

```
Please enter the given number to query
1.today
2.three days from today
3.custom day by yourself
(r)back,(c)cls,(#)exit
ddqwdwqd
input error, please try again:
█
```

四、flag

1. 查询天气时遇到如下结果:

```
1
City: nanjing Today is: 2023/03/21 Weather information is as follows:
Today's Weather is: Please guess: ZmxhZ3tzYW5kX3N0MHJtfQ==; Temp:34
```

观察形式, 立即想到可能是base编码, 使用Base64编码解码可得:

ZmxhZ3tzYW5kX3N0MHJtfQ==

☐ 解密为UTF-8字节流

flag{sand_st0rm}

flag为sand_st0rm, 为第九种天气。

2. 观察服务器发送的数据包, 可得如下形式数据:

0080	00 00 00 00 00 00 00 00	00 00 00 00 00 2e 2e 2d
0090	2e 20 2e 2e 20 2d 2e 20	2d 2e 2e 20 2d 2d 20 2e	.. - . - . - . - .

观察形式, 可能为摩尔斯电码, 解码可得:

..-.. -.. -.. -..

分割

长

➡ 编码

↩ 解码

FINDME

flag为FINDME

五、实验中遇到的问题及解决方案

1. 使用C语言对结构体的处理不够方便，比较麻烦，对字段的读取需要传参。

解决方案：换用C++，使用结构体函数，在维持结构体不变的同时简化了对字段的读取和处理。

六、实验的启示和建议

系统级的套接字封装的相当优秀，大大简化了传输层连接协议的使用过程。

附：共用时间7小时