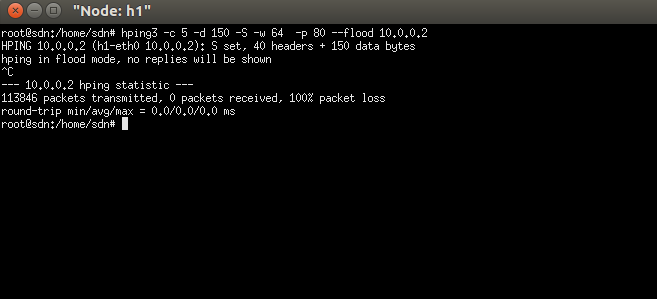
实验记录

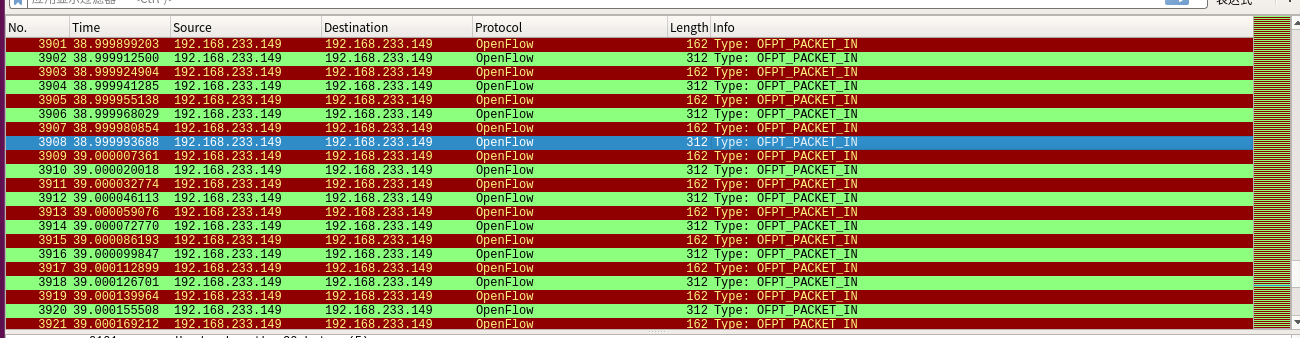
抓openflow包

首先用hping3发动syn攻击，产生大量半连接（没有第三次握手），得到第一次握手和第二次握手的包格式

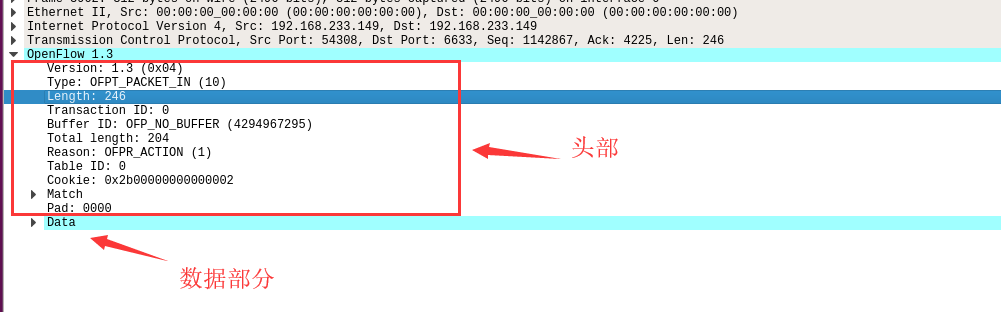
使用hping3工具发动syn攻击：



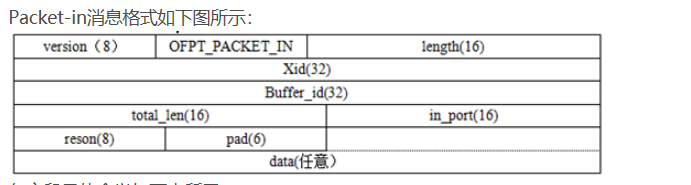
得到半连接：



第一次握手分析：



这里触发packetin的原因是转发到控制器



头部分析：

Version: openflow版本号

Type: 10代表PACKET\_IN的包号

Length：代表openflow包的长度

Transaction ID：packetin对应的事务id，所有的packetin都是一样的

Buffer\_id：表示OpenFlow交换机中保存的数据包的缓存id



数据部分分析：

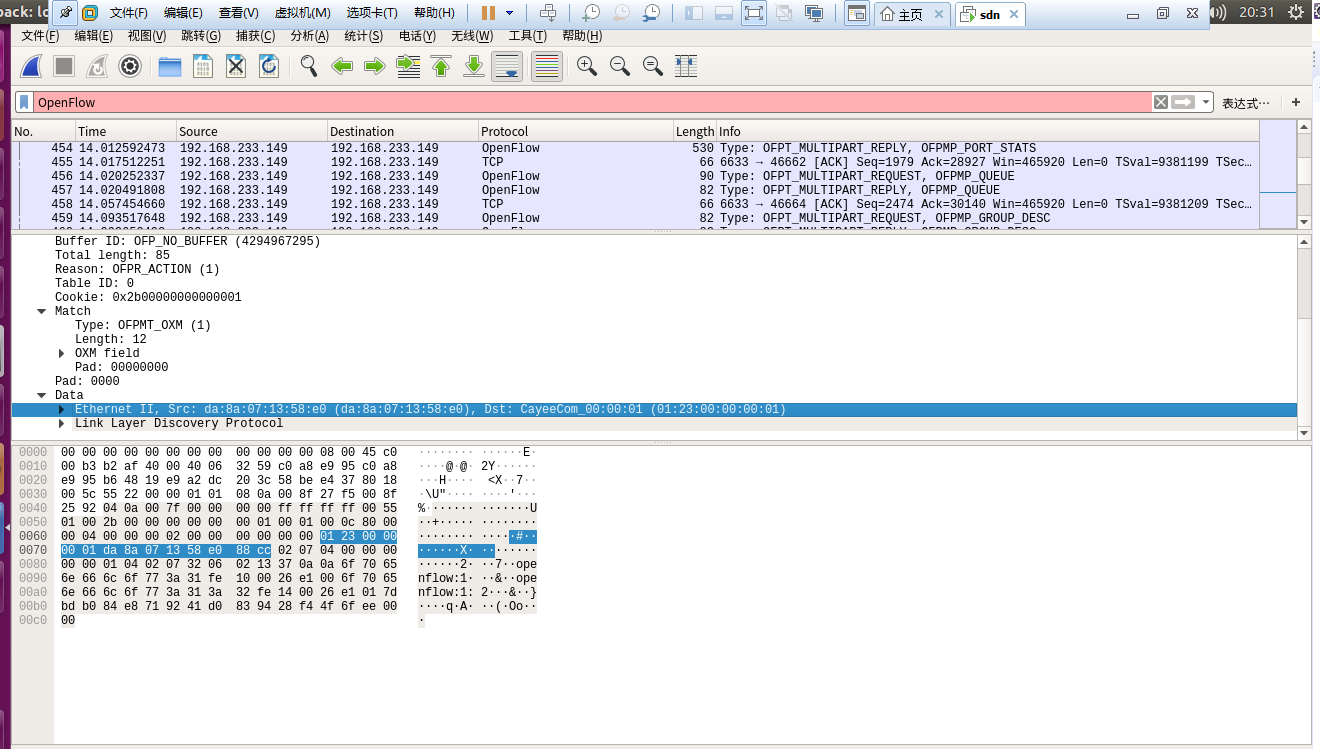
Destination: fa 7c 4d 7e 24 e9 [108,114) 目的mac

Source: ca 4a b2 54 20 ae [114,120) 源mac

Type:IPv4 08 00 [120,122)

|  |  |  |  |
| --- | --- | --- | --- |
| Ip层 | | | |
| 字段 | 十六进制数据 | 位置 | 含义 |
| Version： 4 |  |  | ip版本号 |
| Header Length：5 这两个是一起的 | 45 | [122,123) | 头部长度 |
| Total Length | 00 28 | [124,126) | 总长度 |
| Identification | 94 09 | [126,128) |  |
| Flags | 40 00 | [128,130) | 标志段 |
| Time to live | 40 | [130,131) | 存活时间 |
| Protocol | 06 | [131,132) | 所采用协议 |
| Header checksum | 92 c4 | [132,134) | 头部校验和 |
| Source | 0a 00 00 02 | [134,138) | 源ip |
| Destination | 0a 00 00 01 | [138,142) |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Transmission Control Protocol 层： | | | |
| 字段 | 十六进制数据 | 位置 | 含义 |
| Source Port | 00 50 | [142,144) | 源端口 |
| Destination Port | 09 d2 | [144,146) | 目的端口 |
| Stream index: |  |  |  |
| TCP Segment Len | 50 | [154,155) |  |
| Sequence number | 00 00 00 00 | [146,150) | 序列号 |
| Acknowledgement number | 11 a7 bd f4 | [150,154) | 确认号 |
| Header Length | 50 | [154,155) |  |
| Flags | 50 14 | [154,156) | 标志 |
| Window size value | 00 00 | [156,158) |  |
| [Calculated window size: 0] | 00 00 | [156,158) |  |
| Window size scaling factor | 00 00 | [156,158) |  |
| Checksum | c2 10 | [158,160) |  |
| Urgent point | 00 00 | [160,162) |  |



上面是抓到syn包

下面是对syn包的解析

打印payload字节数组:

num\_packetin++;

**char**[] hexArray = "0123456789ABCDEF".toCharArray();

**if**(num\_packetin == 30) {

**char**[] hexChars = **new** **char**[*payload*.length \* 2];

**for** ( **int** j = 0; j < *payload*.length; j++ ) {

**int** v = *payload*[j] & 0xFF;

hexChars[j \* 2] = hexArray[v >>> 4];

hexChars[j \* 2 + 1] = hexArray[v & 0x0F];

}

String ss = **new** String(hexChars);

**for**(**int** i=0;i<ss.length();i++) {

System.***out***.print(ss.charAt(i));

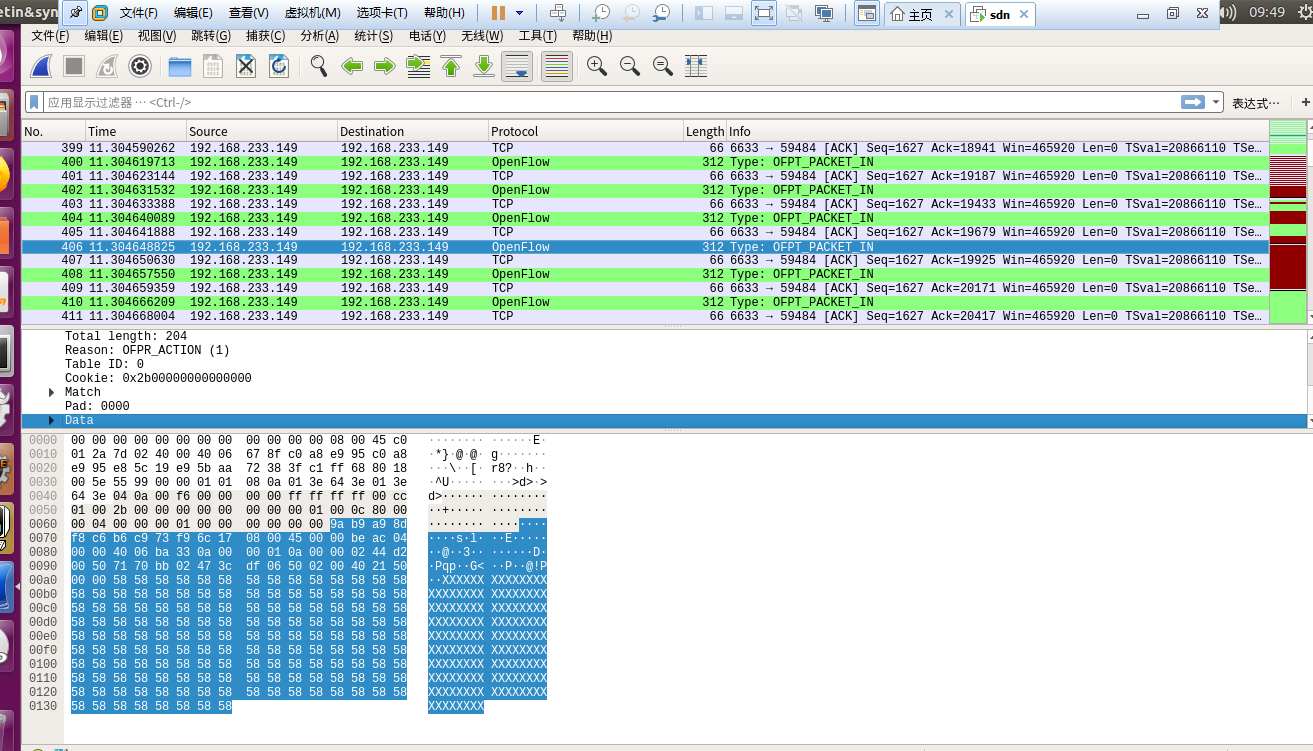
i++;

System.***out***.println(ss.charAt(i));

}

}

抓包：



终端打印：

9A B9 A9 8D F8 C6 B6 C9 73 F9 6C 17 08 00 45 00 00 BE AC 04 00 00 40 06 BA 33 0A 00 00 01 0A 00 00 02 44 D2 00 50 71 70 BB 02 47 3C DF 06 50 02 00 40 21 50 00 00 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 .........

|  |  |  |  |
| --- | --- | --- | --- |
| Ethernet Ⅱ | | | |
| Destination Mac | 9a b9 a9 8d f8 c6 | [0,6) | 目的mac |
| Source Mac | B6 c9 73 f9 6c 17 | [6,12) | 源mac |
| TYPE：IPv4(0x0800) | 08 00 | [12,14) | 类型 |
| IP | | | |
| Version &Header | 45 | [14,15) |  |
| Differentiated Service | 00 | [15,16) |  |
| Total Length:190 | 00 be | [16,18) | 总长度 |
| Identification | ac 04 | [18,20) |  |
| Flags | 00 00 | [20,22) |  |
| Time to live | 40 | [22,23) |  |
| Protocol:TCP（6） | 06 | [23,24) |  |
| Header checksum | ba 33 | [24,26) |  |
| Source:10.0.0.1 | 0a 00 00 01 | [26,30) | 源ip |
| Destination:10.0.0.2 | 0a 00 00 02 | [30,34) | 目的ip |
| Transmission Control Protocol | | | |
| Source Port:17618 | 44 d2 | [34,36) | 源端口 |
| Destination Port:80 | 00 50 | [36,38) | 目的端口 |
| TCP Segment Len:150 | 50 | [46,47) |  |
| Sequence number:0 | 71 70 bb 02 | [38,42) |  |
| Acknowledgement number | 47 3c df 06 | [42,46) |  |
| Header Length:20 b | 50 | [46,47) |  |
| Flags | 50 02 | [46,48) | 标志位 |

解析syn字段的值

num\_packetin++;

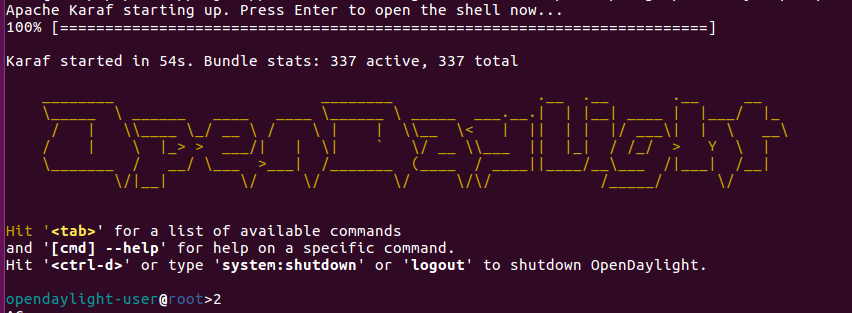
if(num\_packetin == 30) {

i\_syn = Integer.parseInt(syn);

result = i\_syn&2;

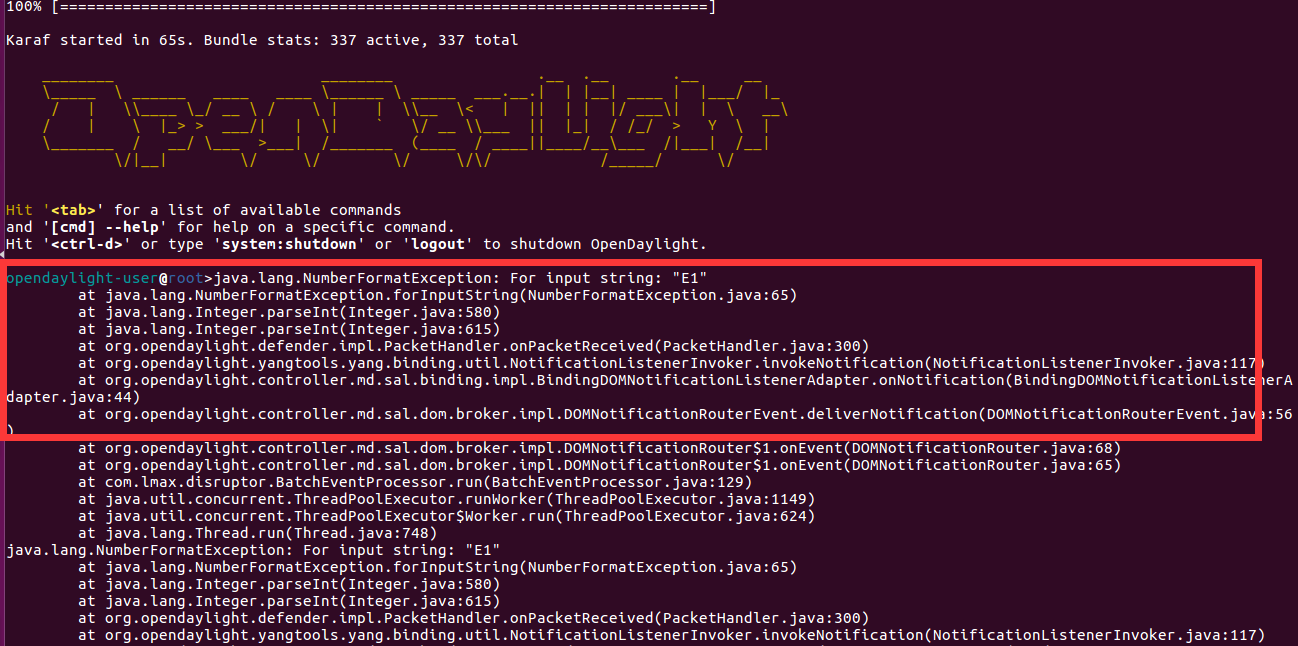
System.out.println(result);

}



检测到syn的值，是syn的包就打印2，不是就打印0

编译出错



这是我抛出的异常

除了这个异常还有一个数组越界的错误：

经过分析发现：

1. 抛出的第一个异常是字符串转整数的时候出现的，要是这个字符串是int型的，那转过来没有什么问题，但是若不是，比如e1，就不行了，所以先进行判断，是不是Int型，也就是可不可以转成Int型
2. 第二个错误是数组越界，经过对比wireshark，发现，packetin包并不都是一样的，有的长，有的短，所以我用解析syn [47,48)这个位置去解析所有的packetin，可能回出现数组越界，所以这里就要判断payload的长度，只有满足syn长度的才去解析它

// 解析syn

if(payload.length>=47) {

rawSyn = PacketParsing.extractSyn(payload);

syn = PacketParsing.rawSynToString(rawSyn);

if(isInt(syn)) {

try {

i\_syn = Integer.parseInt(syn.trim());

result = i\_syn&2;

if(result == 2) {

Lwm();

}

}catch (NumberFormatException e) {

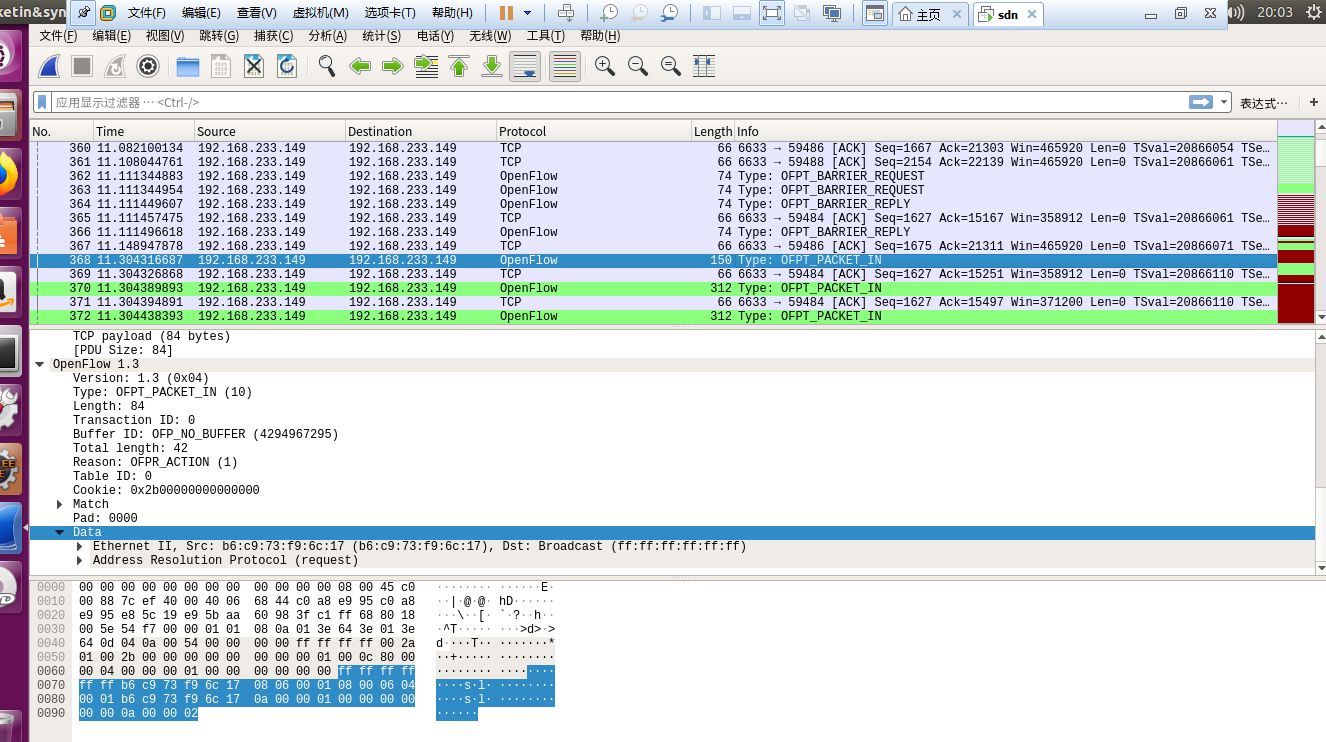
e.printStackTrace();

}

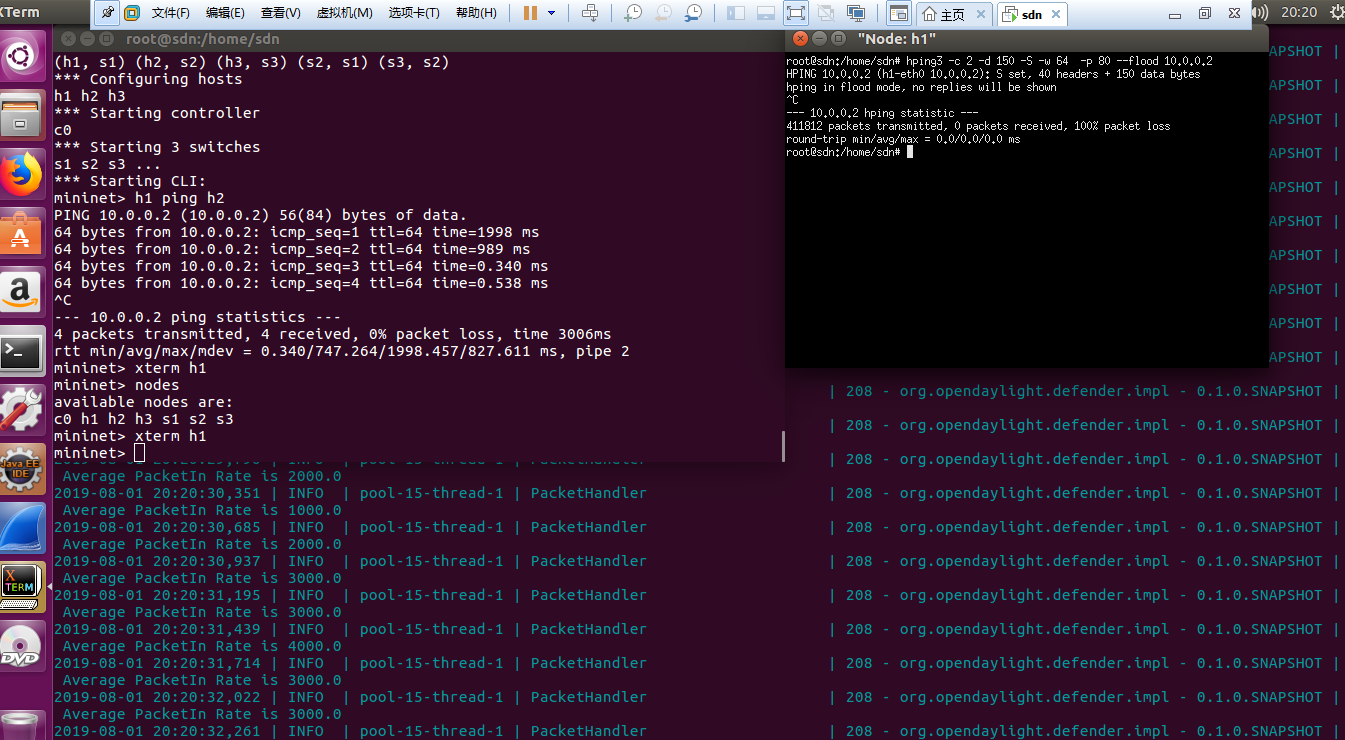
}

}

字节数组不够47的情况



发动h1 ping h2 会报错 syn攻击倒不会有错



看来得判断一下协议类型了