# 基于 jpcap 的网络嗅探器实验报告

孙晴-201818018670034

## 1.软件环境

下载并安装 JRE，下载并安装 winpcap，下载 jpcap 包，有两个文件，分别是 jpcap.dll 和 jpcap.jar，这两个文件分别放到 JAVA 安装路径/jre/bin 和/jre/lib/ext 下。然后正常使用 IntelliJ IDEA x64 新建项目就可以运行。

sun.jar 文件需要在命令行执行 **java -jar sun.jar** 才可以运行。

## 2.主要算法

Jpcap 实际上就是调用了 winpcap/libpcap，给 JAVA 语言提供了一个公共的接口。Jpcap 可以捕获的数据包有 IP 数据包（IPPacket 类）、数据链路层包(DatalinkPacket 类)、以太帧包(EthernetPacket 类）、以及 ICMP、TCP、UDP、ARP 包（ICMP 类、TCP 类、UDP 类、ARP 类）。

本实验只进行了对 ICMP、TCP、UDP、IP、ARP 数据包的捕获以及分析。

获得网卡信息：JpcapCaptor.getDeviceList()

打开网络接口：JpcapCaptor CAP = JpcapCaptor.openDevice();

捕获数据包：CAP.processPacket();

```
List<Packet> packetList = new ArrayList();
packetList.add(CAP.getPacket());
```

设置规则(用户选择)：CAP.setFilter(“UDP”,” true”)

Java 文件注释：

线程编程：jpcap_thread.java

获取网卡信息窗口：InterfacesWindows.java

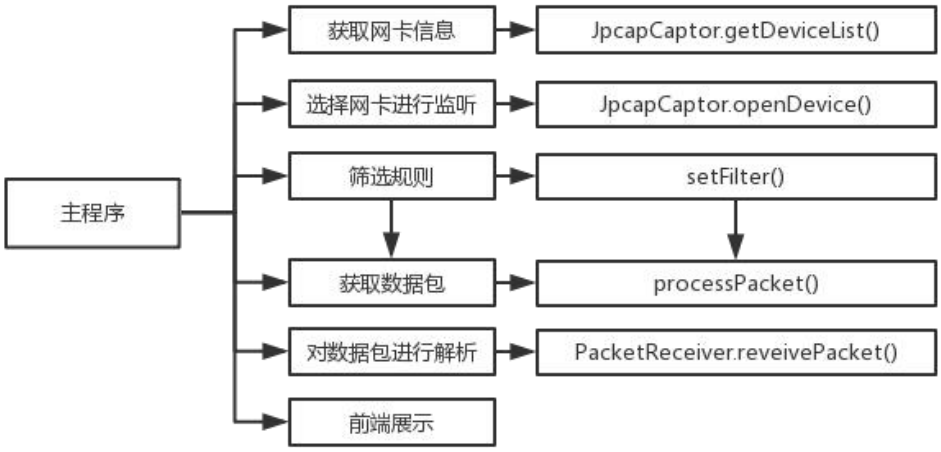
数据包信息分类：PacketContents.java

主要窗口界面操作以及抓取数据包等函数：Sniffer.java

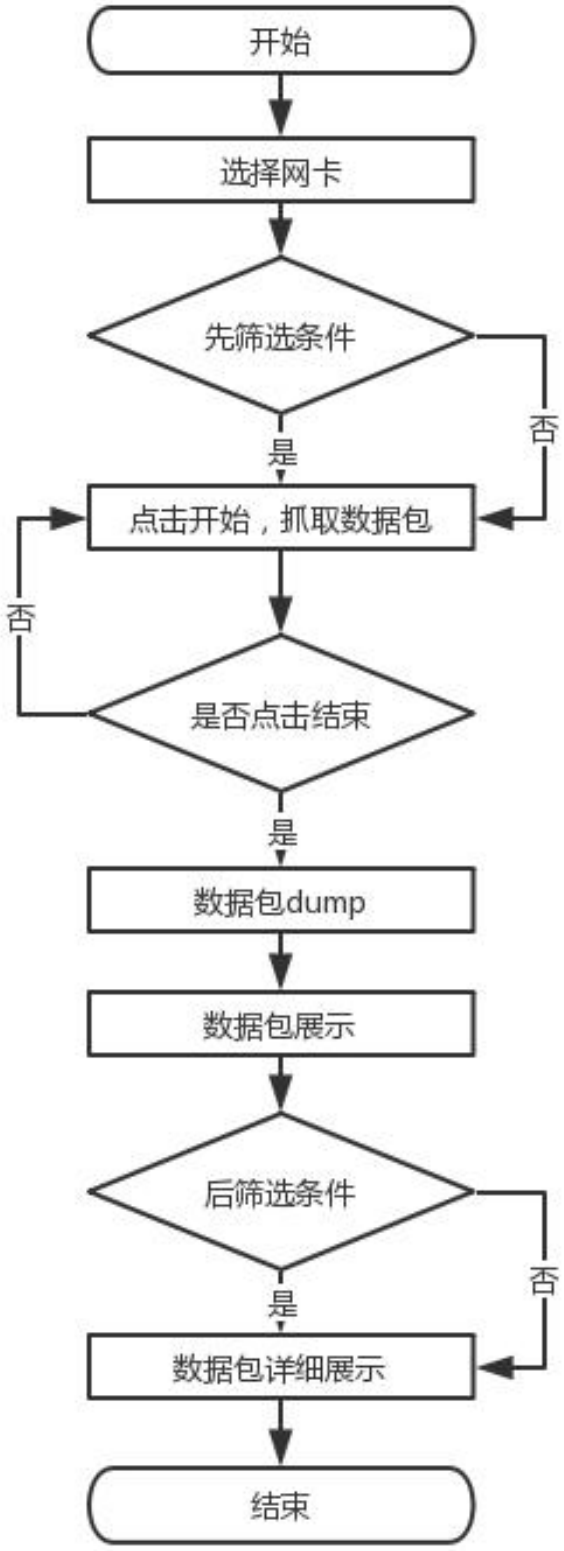
总结：

就是先进行网卡信息展示，然后用户选择网卡进行监听，接下来进行数据包的捕获，线程编程，并可以实时的展示，可以通过点击数据包条目查看数据包的详细信息，以及 16 进制文件。添加了捕获数据包前的筛选，以及捕获后的筛选操作。可以进行 dump 操作。

## 3.Jpcap 实现的 sniffer 基本框架




## 4.算法主要流程图


开始
选择网卡
先筛选条件
否
是
点击开始，抓取数据包
否
是否点击结束
是
数据包dump
数据包展示
后筛选条件
否
是
数据包详细展示
结束

## 5.遇到的问题以及解决办法

在配置环境时，无法配置成功，原因是 winpcap 以及 jpcap 以及 jre 都需要配置成 64 位的，然后就可以正常运行了。

Dump 总是出现问题，提示 javasvm 出错，将 svm 以及 javajre 都配置成统一版本才可以实现。

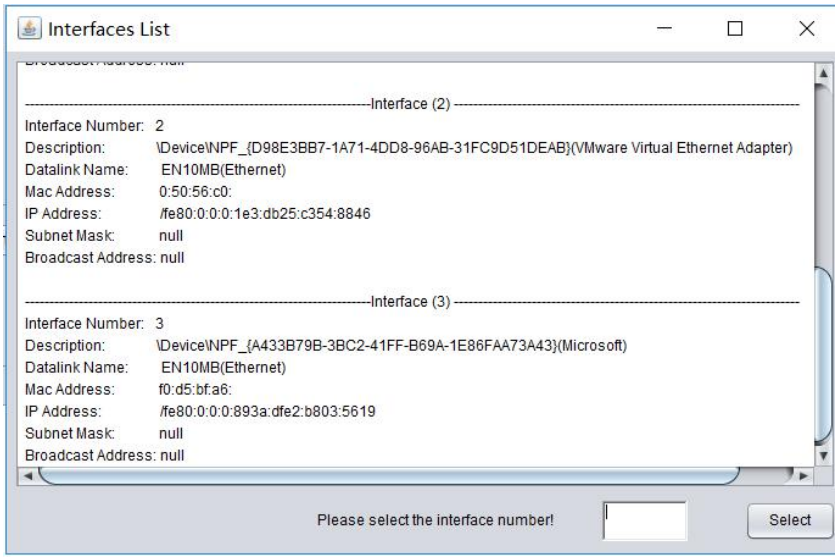
只有在电脑上存在 javasvm 环境的时候才可以双击 jar 文件直接打开，通过命令行 java -jre sun.jre 可以实现程序打开。

## 6.收获以及建议

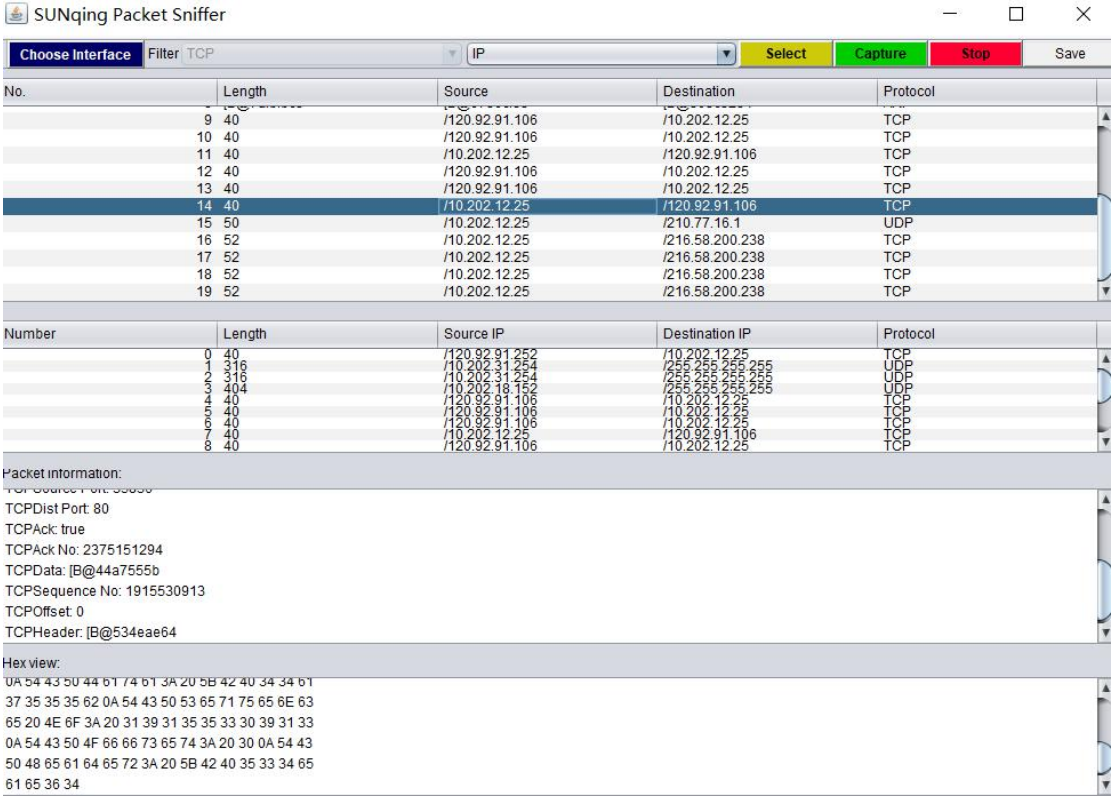
老师给了两周的时间进行实验，有基础并且做过的可以简单的时间，我在大学进行过类似的环境配置，所以这次有很多经验可以直接借鉴，因为 sniffer 的主要代码部分都是在前端展示部分，这次我选择了最简单的 GUI 来实现，其实感觉主要的程序算法代码量并不多，前端部分工作量太大。因为每周都要回所以及其他的作业，这次没有进行对 html 等包的精确分析，如果老师能要求不写前端，同等时间下，应该代码会更加优秀，功能也应该实现的更多一些。

## 7.程序截图，以及运行结果截图

网卡选择窗口。



运行时截图。


SUNqing Packet Sniffer
Choose Interface
Filter TCP
IP
Select
Capture
Stop
Save
No. Length Source Destination Protocol
9 40 /120.92.91.106 /10.202.12.25 TCP
10 40 /120.92.91.106 /10.202.12.25 TCP
11 40 /10.202.12.25 /120.92.91.106 TCP
12 40 /120.92.91.106 /10.202.12.25 TCP
13 40 /120.92.91.106 /10.202.12.25 TCP
14 40 /10.202.12.25 /120.92.91.106 TCP
15 50 /10.202.12.25 /210.77.16.1 UDP
16 52 /10.202.12.25 /216.58.200.238 TCP
17 52 /10.202.12.25 /216.58.200.238 TCP
18 52 /10.202.12.25 /216.58.200.238 TCP
19 52 /10.202.12.25 /216.58.200.238 TCP
Number Length Source IP Destination IP Protocol
0 40 /120.92.91.252 /10.202.12.25 TCP
1 316 /10.202.31.254 /255.255.255.255 UDP
2 316 /10.202.31.254 /255.255.255.255 UDP
3 404 /10.202.18.152 /255.255.255.255 UDP
4 40 /120.92.91.106 /10.202.12.25 TCP
5 40 /120.92.91.106 /10.202.12.25 TCP
6 40 /120.92.91.106 /10.202.12.25 TCP
7 40 /10.202.12.25 /120.92.91.106 TCP
8 40 /120.92.91.106 /10.202.12.25 TCP
Packet information:
TCPDist Port: 80
TCPAck: true
TCPAck No: 2375151294
TCPData: [B@44a7555b
TCPSequence No: 1915530913
TCPOffset: 0
TCPHeader: [B@534eae64
Hex view:
37 35 35 35 62 0A 54 43 50 53 65 71 75 65 6E 63
65 20 4E 6F 3A 20 31 39 31 35 35 33 30 39 31 33
0A 54 43 50 4F 66 66 73 65 74 3A 20 30 0A 54 43
50 48 65 61 64 65 72 3A 20 5B 42 40 35 33 34 65
61 65 36 34