哈爾濱工業大學（深圳）

Harbin Institute of Technology, Shenzhen

Advanced Computer Networking

Experiment Report

Group Members:

姓名　庄舒漫　　学号 22S15118

姓名　唐栋　　学号 22S151060

报告日期: 2022.06.09

1.实验概述

实验任务

只实现了 project1

write a program that can capture the coming packets and revert the webpage which you select to test by using the captured data.

实验步骤

Capture the packets

Reconstruct http message

Revert the webpage with captured data

开发平台

Os: win10

编译工具：mingw64,npcap,python3.9,cmake

可视化平台：QT5

2.实验过程

1.Npcap 配置

在 vscode 平台将相应的 npcap 动态链接库和函数头部文件放入工程文件夹

使用语句

```
include_directories(Include)
include_directories(Lib/x64)
link_directories(Lib/x64)
enable_testing()
add_executable(NPCAP main.cpp)

target_link_libraries(NPCAP wpcap.lib Packet.lib)
target_link_libraries(NPCAP ws2_32.lib)
```

注意动态链接要在 add_executable 语句后，而加载静态库 include/link 要在该语句之前

Make 后就会生成 win 平台的可执行文件

```
NPCAP.exe
```

2.根据 npcap 函数捕包

1. pcap_findalldevs

NpCap 提供了 pcap_findalldevs_ex 和 pcap_findalldevs 函数来获取计算机上的网络接口设备的列表；此函数会为传入的 pcap_if_t 赋值——该类型是一个表示了设备列表的链表头；每一个这样的节点都包含了 name 和 description 域来描述设备。

除此之外，pcap_if_t 结构体还包含了一个 pcap_addr 结构体；后者包含了一个地址列表、一个掩码列表、一个广播地址列表和一个目的地址的列表；此外，pcap_findalldevs_ex 还能返回远程适配器信息和一个位于所给的本地文件夹的 pcap 文件列表。

2. pcap_open

用来打开一个适配器，实际调用的是 pcap_open_live；它接受五个参数：

name：适配器的名称（GUID）

snaplen：制定要捕获数据包中的哪些部分。在一些操作系统中 (比如 xBSD 和 Win32)，驱动可以被配置成只捕获数据包的初始化部分：这样可以减少应用程序间复制数据的量，从而提高捕获效率；本次实验中，将值定为 65535，比能遇到的最大的 MTU 还要大，因此总能收到完整的数据包。

flags：主要的意义是其中包含的混杂模式开关；一般情况下，适配器只接收发给它自己的数据包， 而那些在其他机器之间通讯的数据包，将会被丢弃。但混杂模式将会捕获所有的数据包——因为我们需要捕获其他适配器的数据包，所以需要打开这个开关。

to_ms：指定读取数据的超时时间，以毫秒计；在适配器上使用其他 API 进行读取操作的时候，这些函数会在这里设定的时间内响应——即使没有数据包或者捕获失败了；在统计模式下，to_ms 还可以用来定义统计的时间间隔：设置为 0 说明没有超时——如果没有数据包到达，则永远不返回；对应的还有 -1：读操作立刻返回。

errbuf：用于存储错误信息字符串的缓冲区

该函数返回一个 pcap_t 类型的 handle。

3. pcap_loop

API 函数 pcap_loop 和 pcap_dispatch 都用来在打开的适配器中捕获数据包；但是前者会已知捕获直到捕获到的数据包数量达到要求数量，而后者在到达了前面 API 设定的超时时间之后就会返回（尽管这得不到保证）；前者会在一小段时间内阻塞网络的应用，故一般项目都会使用后者作为读取数据包的函数；虽然在本次实验中，使用前者就够了。

这两个函数都有一个回调函数；这个回调函数会在这两个函数捕获到数据包的时候被调用，用来处理捕获到的数据包；这个回调函数需要遵顼特定的格式。但是需要注意的是我们无法发现 CRC 冗余校验码——因为帧到达适配器之后，会经过校验确认的过程；这个过程成功，则适配器会删除 CRC；否则，大多数适配器会删除整个包，因此无法被 NpCap 确认到。

4. pcap_datalink

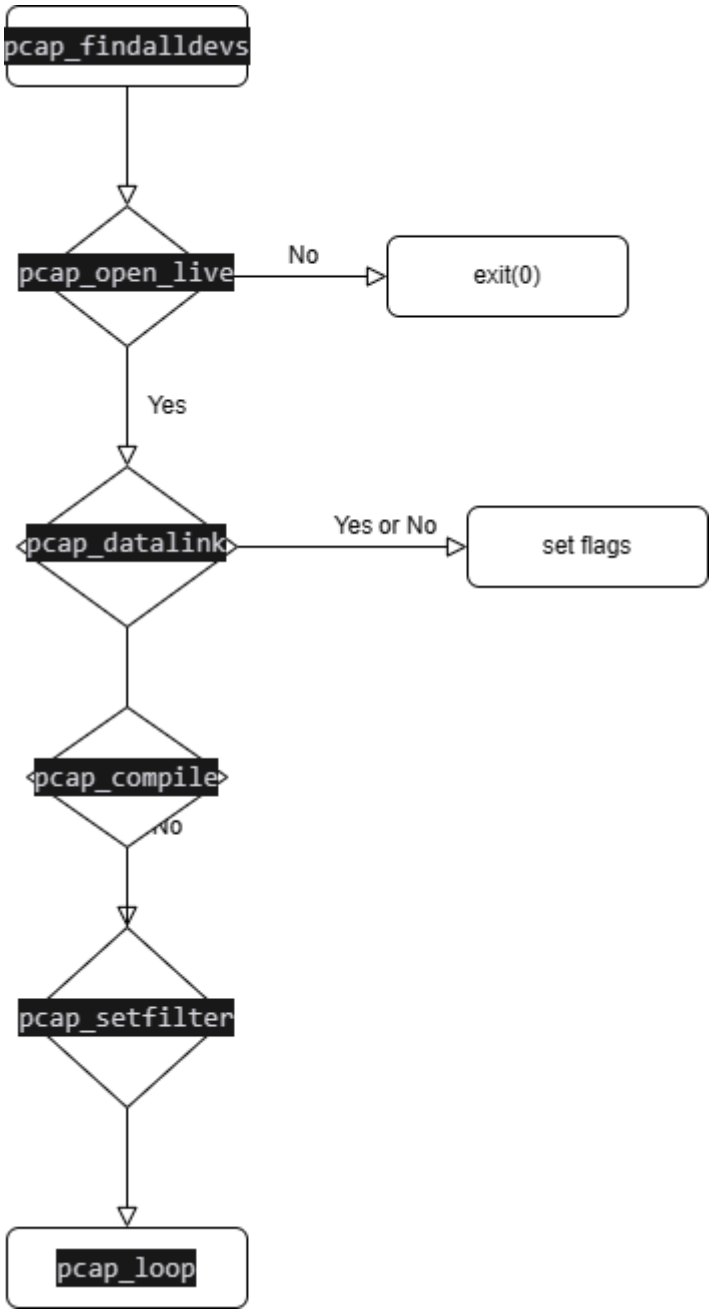
用于对 MAC 层进行了检测，以确保在处理一个以太网络，确保 MAC 首

部是 14 位的。IP 数据包的首部就位于 MAC 首部的后面，将从 IP 数据包的首部解析到源 IP 地址和目的 IP 地址。

本次实验主要抓取了以太网的包和本地回环网络的数据包，由于这两种包包头格式不一样，所以必须用该函数区分，设置标志

5. pcap_compile & pcap_setfilter

用来设置过滤器，以避免处理一些无用的包，提高包处理的效率。在本次实验中我们需要可以按照过滤语句格式输入过滤规则



3.定义网络层，传输层，应用层协议头部信息

报文定义如下所示：

```
typedef struct ip_header
{
    u_char  ver_ihl;        // Version (4 bits) + Internet header length (4 bits)
    u_char  tos;            // Type of service
    u_short tlen;           // Total length
    u_short identification; // Identification
    u_short flags_fo;       // Flags (3 bits) + Fragment offset (13 bits)
    u_char  ttl;            // Time to live
    u_char  proto;          // Protocol
    u_short crc;            // Header checksum
    ip_address  saddr;      // Source address
    ip_address  daddr;      // Destination address
    u_int   op_pad;         // Option + Padding
}ip_header;

/* UDP header*/
typedef struct udp_header
{
    u_short sport;          // Source port
    u_short dport;          // Destination port
    u_short len;            // Datagram length
    u_short crc;            // Checksum
}udp_header;


//TCP首部
typedef struct tcp_header
{
    unsigned short src_port;    //源端口号
    unsigned short dst_port;    //目的端口号
    unsigned int seq_no;        //序列号
    unsigned int ack_no;        //确认号
    unsigned char reserved_1; //保留6位中的4位首部长度
    // unsigned char thl : 4;        //tcp头部长度
    // unsigned char flag : 6;       //6位标志
    unsigned char reseverd_2; //保留6位中的2位
    unsigned short wnd_size;    //16位窗口大小
    unsigned short chk_sum;     //16位TCP检验和
    unsigned short urgt_p;      //16为紧急指针

}tcp_header;
```

这里需要注意的要点：

1.超过一个字节字段定义要注意大小端，大小端是指字节排序方式，大端高位数据在低地址，小端低位数据在低地址，要注意使用 windows 特有函数（不是 mingw 编译器中带有）,ntohs(短型)，ntohl(长型整数)（win_32），在一个字节内只要按位取需要的值即可

2．关于数据报帧头，以太网是 14 字节，本地回环包是 4 字节。

3．以太网最小帧大小是 60，小于这个会用 0x00padding

4.以太网最大数据包长度是 1514 = 54 + 1460（数据大小）

4.解析 http 数据包内容

图 15-4　HTTP 请求报文

需要注意的头部信息：

content_type:关心是一个 html/txt 文本页面或者是 png,jpeg 图片格式，或者是.css 网页渲染器

Content_encoding:现在 http 网页为了加快响应速度，基本都会采用 gzip 压缩网页内容传输，所以需要判断是否为 gzip 压缩，并进行解压

Charset:主要为了后面解压后用什么编码二进制串

Content_length:为了判断是否获取数据包全部内容

示例：

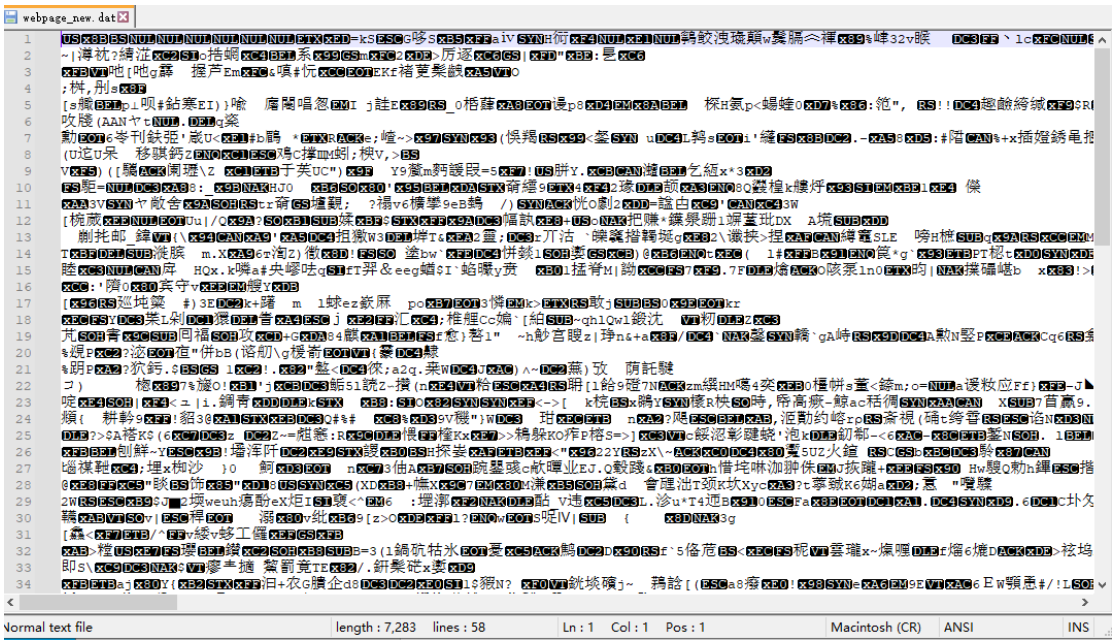
这里我抓取的是使用 ipv4 协议的 http 协议的一个网站，网址为 www.weislank.com

抓取 http 相应报文内容如下图所示：

```
> HTTP/1.1 200 OK\r\n
  Date: Sun, 18 Jun 2023 04:19:08 GMT\r\n
  Server: Apache/2\r\n
  Upgrade: h2,h2c\r\n
  Connection: Upgrade, Keep-Alive\r\n
  Vary: Accept-Encoding,User-Agent\r\n
  Content-Encoding: gzip\r\n
> Content-Length: 7283\r\n
  Keep-Alive: timeout=2, max=100\r\n
  Content-Type: text/html; charset=utf-8\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.104551000 seconds]
  [Request in frame: 1146]
  [Request URI: http://weislank.com/]
  Content-encoded entity body (gzip): 7283 bytes -> 30740 bytes
  File Data: 30740 bytes
```

可以看到这里使用 gzip 压缩，编码格式是 utf-8，数据包内容一共有 7283 个字节

这是我重组帧之后得到的数据内容,用 nodepad++打开：



此时 length 大小确实为 7283

使用 gzip 解压后，并使用 utf-8 编码，得到 html 文件如下所示：

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>威斯兰卡官方网站</title>
<meta name="keywords" content="威斯兰卡官方网站">
<meta name="description" content="威斯兰卡（weislank）是个产品远销国内外200多个国家的国际化品牌，专注于乒乓球行业产品，采用优质的原材料，严格贯彻

公司将秉承以探索求进取、诚信求发展、质量求生存的理念携手广大客商和合作伙伴相互依托、共同发展，开创体育事业的新世纪新曙光。">
<link rel="shortcut icon" href="www.weislank.com/uploads/20220708/4b9cff1528817b82a9d07aff2cebeec0.png"><link rel="stylesheet"
<link rel="stylesheet" href="www.weislank.com/static/index/zm001/css/index.css?v=1.1">
<link rel="stylesheet" href="www.weislank.com/static/index/zm001/css/inner.css?v=1.2.1">
<link rel="stylesheet" href="www.weislank.com/static/index/zm001/js/swiper7/swiper-bundle.min.css">
<link rel="stylesheet" href="www.weislank.com/static/index/zm001/css/skrles.css" />
<link rel="stylesheet" href="www.weislank.com/static/index/zm001/css/iconfont.css">
<link rel="stylesheet" href="www.weislank.com/static/index/zm001/css/animate.min.css">
<script src="www.weislank.com/static/index/zm001/js/jquery.min.js"></script>
<script src="www.weislank.com/static/index/zm001/js/layer/layer.js"></script>
</head>

<body class="bg-color ">
  <header id="header" class="web-header w1920">
    <nav class="header-nav main flex">
        <div class="nav-logo">
            <h1>
                <a href="/">
                    <img src="www.weislank.com/uploads/20220713/f18be01be3014541fb4354bdbe447355.PNG" title="威斯兰卡官方网站" al
```

4.重组数据报

虽然 http 数据报内容显然无法用一个窗口发送完毕，但我发现服务器并没有对 ip 数据报分片，而是采用了不同 tcp 包+fin 片标识来发送整个数据包

所以，程序实现逻辑如下：

1. 找到第一个相应数据包

   以 HTTP 开头的数据包，去掉 http 协议头部信息，将数据部分内容写入二进制文件中

```
00 ed 06 3e 00 00 48 54  54 50 2f 31 2e 31 20 32   ···>··HT TP/1.1 2
30 30 20 4f 4b 0d 0a 44  61 74 65 3a 20 53 75 6e   00 OK··D ate: Sun
2c 20 31 38 20 4a 75 6e  20 32 30 32 33 20 30 34   , 18 Jun  2023 04
3a 31 39 3a 30 34 20 47  4d 54 0d 0a 53 65 72 76   :19:04 G MT··Serv
65 72 3a 20 41 70 61 63  68 65 2f 32 0d 0a 55 70   er: Apac he/2··Up
67 72 61 64 65 3a 20 68  32 2c 68 32 63 0d 0a 43   grade: h 2,h2c··C
6f 6e 6e 65 63 74 69 6f  6e 3a 20 55 70 67 72 61   onnectio n: Upgra
64 65 2c 20 4b 65 65 70  2d 41 6c 69 76 65 0d 0a   de, Keep -Alive··
56 61 72 79 3a 20 41 63  63 65 70 74 2d 45 6e 63   Vary: Ac cept-Enc
6f 64 69 6e 67 2c 55 73  65 72 2d 41 67 65 6e 74   oding,Us er-Agent
0d 0a 43 6f 6e 74 65 6e  74 2d 45 6e 63 6f 64 69   ··Conten t-Encodi
6e 67 3a 20 67 7a 69 70  0d 0a 43 6f 6e 74 65 6e   ng: gzip ··Conten
74 2d 4c 65 6e 67 74 68  3a 20 37 32 38 33 0d 0a   t-Length : 7283··
4b 65 65 70 2d 41 6c 69  76 65 3a 20 74 69 6d 65   Keep-Ali ve: time
6f 75 74 3d 32 2c 20 6d  61 78 3d 31 30 30 0d 0a   out=2, m ax=100··
43 6f 6e 74 65 6e 74 2d  54 79 70 65 3a 20 74 65   Content- Type: te
78 74 2f 68 74 6d 6c 3b  20 63 68 61 72 73 65 74   xt/html;  charset
3d 75 74 66 2d 38 0d 0a  0d 0a 1f 8b 08 00 00 00   =utf-8·· ........
00 00 00 03 ed 3d 6b 53  1b 47 b6 df 53 b5 ff 61   ·····=kS ·G··S··a
a2 a4 16 48 d0 63 f4 00  e1 00 fb 48 f6 de 9b aa   ···H·c·· ···H····
ad 4d ee 8d 77 f7 de eb  f5 a6 e6 d1 82 89 25 8d   ·M··w··· ......%·
56 33 32 76 b2 54 09 13  0c d8 bc 6c 63 fc 00 02   V32v·T·· ···lc···
d8 c6 b0 4e 0c 7e 9b a7  f9 31 ab 1e 49 9f f2 17   ···N·~·· ·1··I···
ee e9 e9 91 34 4f 49 80  13 ef dd 8d ca 65 66 7a   ····4OI· ·····efz
4e 9f 3e 7d 5e 7d fa 4c  eb a8 fb ed 8f 3e f9 f0   N·>}^}·L ·····>··
e4 ff 7c fa 1b a6 5f 4d  25 7b 7f f6 56 b7 f1 97   ··|···_M %{··V···
5c 21 4e 84 2b 86 e9 4e  21 95 63 84 7e 2e ab 20   \!N·+··N !·c·~.·
b5 c7 f7 fb 93 ff e6 8f  fb 08 ac de de af aa 19   ........ ........
3f fa 4b 4e 3a db e3 fb  6f ff ef 7f e5 ff 50 4e   ?·KN:··· o·····PN
65 38 55 e2 93 c8 c7 08  72 5a 45 69 e8 24 a1 1e   e8U····· rZEi·$··
24 f6 21 bd 9b 2a a9 49  d4 8b d7 87 b4 b9 4d 3c   $·!··*·I ......M<
f2 08 4f ae e0 8d 9b da  dc 76 71 ff 4a f1 db 5b   ··O····· ·vq·J··[
dd 41 fa bc 82 3f cd a5  50 8f ef 0c 3a 3f 20 67   ·A···?·· P···:? g
45 c5 84 d2 0b 81 cf d6  55 44 8a 90 95 32 aa 24   E····· UD···2·$
a7 3d 7a 7f bf 37 36 80  24 25 c9 a5 cf 7c bf 37   ·=z··76· $%···|·7
ae dd dc 2c 6c 3d 28 ec  ac e1 6b 43 a5 83 85 f2   ···,l=(· ··kC····
6c 1e cf ef e3 8b 23 f8  de 5c 38 14 c2 f7 6e c3   l·····#· ·\8···n·
```

2. 判断接下来的序列号是否 == 上一个包序列号+数据大小（1460）

```
if(seq_pre!=0&& seq_no!= seq_pre+len_pre){
    if(remove("webpage_new.dat")==0) cout<<"remove success!"<<endl;
    printf("wrong sequence!\n");
    exit(0);
    // exit(0);
}
```

3. 判断 fin 标识是否为真，若为真，且数据大小==报文大小，退出程序

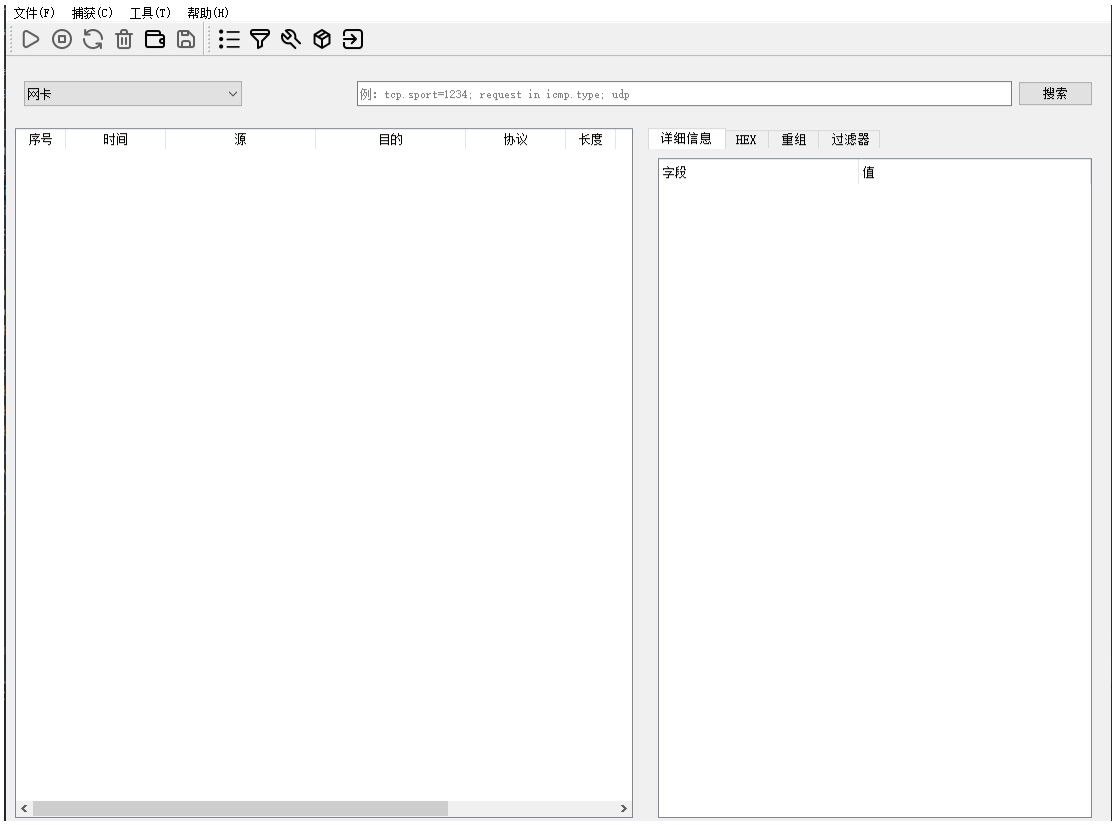
```
    cout<<endl;
    if(ack&&fin&&flag_pre){
        // deal_file();
        exit(0);
    }
    if(header->len <= 60) return;
```

最终重组包信息为：

```
data_len: 1460
HTTP/1.1 200 OK
Date: Sun, 18 Jun 2023 04:38:40 GMT
Server: Apache/2
Upgrade: h2,h2c
Connection: Upgrade, Keep-Alive
Vary: Accept-Encoding,User-Agent
Content-Encoding: gzip
Content-Length: 7283
Keep-Alive: timeout=2, max=100
Content-Type: text/html; charset=utf-8120.48.171.216:80 -> 10.249.43.35:50395
12:38:39.645125 len:1514
head_len: 54
data_len: 1460
120.48.171.216:80 -> 10.249.43.35:50395
12:38:39.645317 len:1514
head_len: 54
data_len: 1460
120.48.171.216:80 -> 10.249.43.35:50395
12:38:39.645317 len:1514
head_len: 54
data_len: 1460
120.48.171.216:80 -> 10.249.43.35:50395
12:38:39.645317 len:1514
head_len: 54
data_len: 1460
120.48.171.216:80 -> 10.249.43.35:50395
12:38:39.645317 len:313
head_len: 54
data_len: 259
120.48.171.216:80 -> 10.249.43.35:50395
12:38:41.645864 len:60
```

5.可视化界面

可视化使用 python 编写，实现起来较为简单快捷，主要使用了 pyqt5
窗口如下所示：



实验结果

1.实验条件

1.1 运行条件：

访问公有网站：www.weislank.com
网卡：以太网卡
过滤语句： src host 120.48.171.216
访问自己搭建网页：

```
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:8083
 * Running on http://10.249.43.35:8083
```

网卡：本地回环抓包网卡
过滤语句：src port 8083

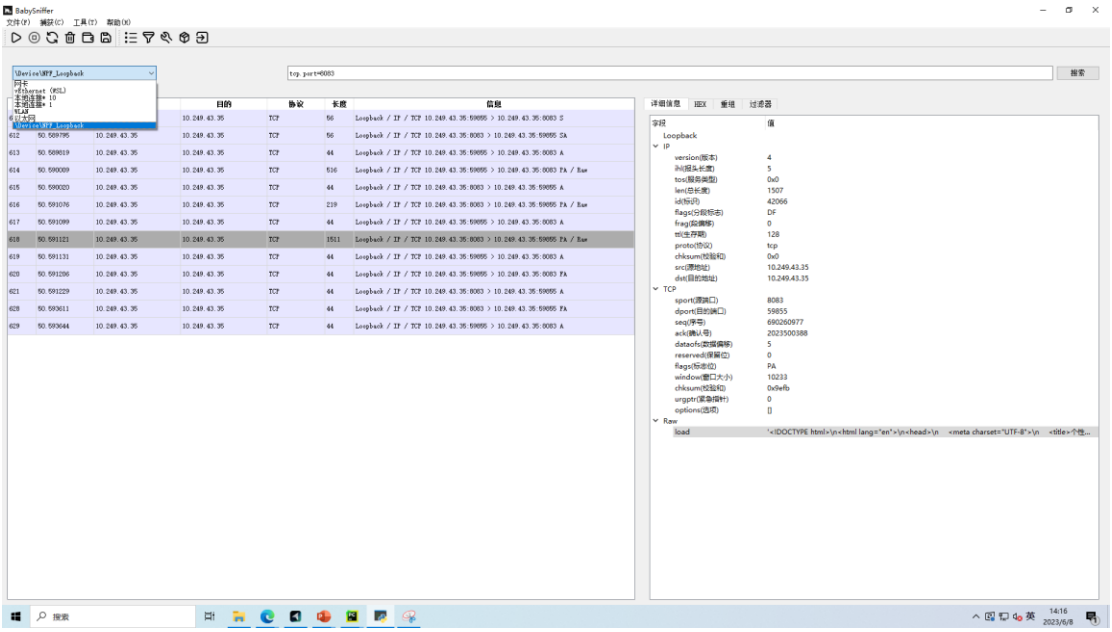
1.2 项目结构：

分为 c++文件和 python 文件
C++文件功能为捕包，重组包，分析网页内容

具体运行方法：
运行：C:\Users\18399\Desktop\project1\npcap\build 路径下的 NPCAP.exe 文件即可，注意网卡选择和过滤语句编写
Python 文件主要为可视化，页面搭载，以及数据包解压
具体运行方法：
运行 C:\Users\18399\Desktop\project1\Sniffer-main\Sniffer-main\source 路径下的 main.py 即可实现可视化
运行 C:\Users\18399\Desktop\project1\Sniffer-main\Sniffer-main\utils 路径下的 webpage.py 即可搭建简单页面
运行 C:\Users\18399\Desktop\project1\Sniffer-main\Sniffer-main\utils 路径下的 decode.py 对二进制文件解压

2.可视化界面实现功能

1. 网卡识别



2. 包过滤

\Device\NPF_Loopback

tcp.port=8083

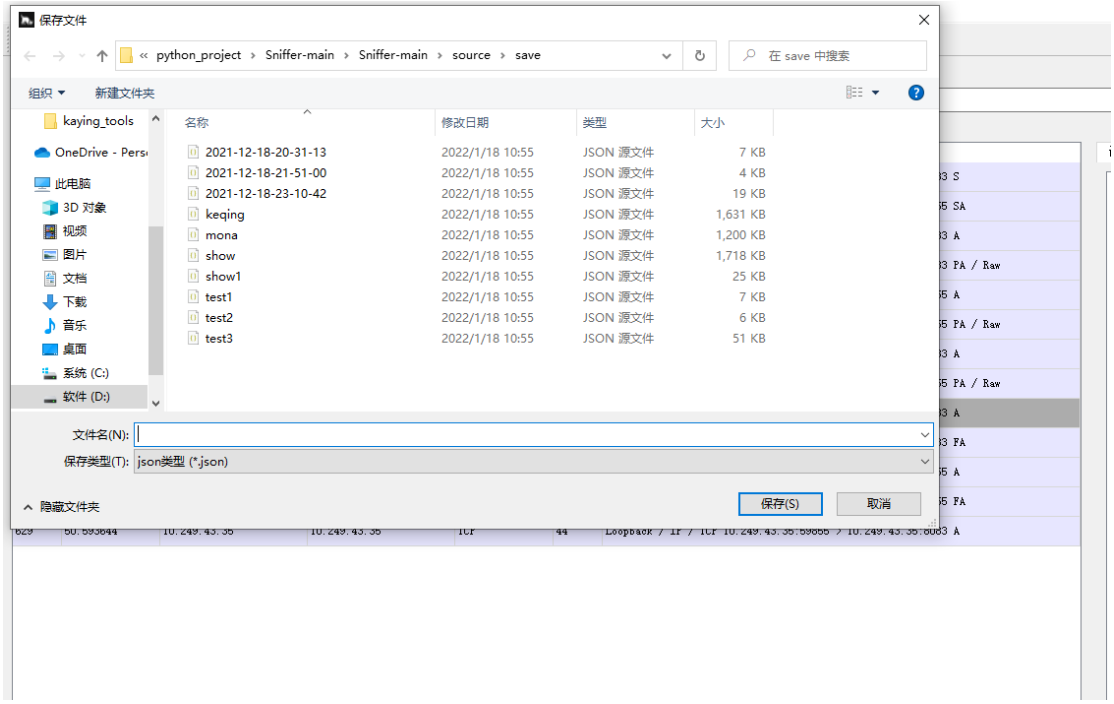
序号	时间	源	目的	协议	长度	信息
611	50.589734	10.249.43.35	10.249.43.35	TCP	56	Loopback / IP / TCP 10.249.43.35:59855 > 10.249.43.35:8083 S
612	50.589795	10.249.43.35	10.249.43.35	TCP	56	Loopback / IP / TCP 10.249.43.35:8083 > 10.249.43.35:59855 SA
613	50.589819	10.249.43.35	10.249.43.35	TCP	44	Loopback / IP / TCP 10.249.43.35:59855 > 10.249.43.35:8083 A
614	50.590009	10.249.43.35	10.249.43.35	TCP	516	Loopback / IP / TCP 10.249.43.35:59855 > 10.249.43.35:8083 PA / Raw
615	50.590020	10.249.43.35	10.249.43.35	TCP	44	Loopback / IP / TCP 10.249.43.35:8083 > 10.249.43.35:59855 A
616	50.591076	10.249.43.35	10.249.43.35	TCP	219	Loopback / IP / TCP 10.249.43.35:8083 > 10.249.43.35:59855 PA / Raw
617	50.591099	10.249.43.35	10.249.43.35	TCP	44	Loopback / IP / TCP 10.249.43.35:59855 > 10.249.43.35:8083 A
618	50.591121	10.249.43.35	10.249.43.35	TCP	1511	Loopback / IP / TCP 10.249.43.35:8083 > 10.249.43.35:59855 PA / Raw
619	50.591131	10.249.43.35	10.249.43.35	TCP	44	Loopback / IP / TCP 10.249.43.35:59855 > 10.249.43.35:8083 A
620	50.591206	10.249.43.35	10.249.43.35	TCP	44	Loopback / IP / TCP 10.249.43.35:59855 > 10.249.43.35:8083 FA
621	50.591229	10.249.43.35	10.249.43.35	TCP	44	Loopback / IP / TCP 10.249.43.35:8083 > 10.249.43.35:59855 A
628	50.593611	10.249.43.35	10.249.43.35	TCP	44	Loopback / IP / TCP 10.249.43.35:8083 > 10.249.43.35:59855 FA
629	50.593644	10.249.43.35	10.249.43.35	TCP	44	Loopback / IP / TCP 10.249.43.35:59855 > 10.249.43.35:8083 A

3. 协议头部解析

详细信息 HEX 重组 过滤器

字段	值
Loopback	
IP	
version(版本)	4
ihl(报头长度)	5
tos(服务类型)	0x0
len(总长度)	1507
id(标识)	42066
flags(分段标志)	DF
frag(段偏移)	0
ttl(生存期)	128
proto(协议)	tcp
chksum(校验和)	0x0
src(源地址)	10.249.43.35
dst(目的地址)	10.249.43.35
TCP	
sport(源端口)	8083
dport(目的端口)	59855
seq(序号)	690260977
ack(确认号)	2023500388
dataofs(数据偏移)	5
reserved(保留位)	0
flags(标志位)	PA
window(窗口大小)	10233
chksum(校验和)	0x9efb
urgptr(紧急指针)	0
options(选项)	[]
Raw	
load	'<!DOCTYPE html>\n<html lang="en">\n<head>\n <meta charset="UTF-8">\n <title>个性...

4.保存包内容



3.还原网页

这一部分使用 c++编写

具体过程：

1. 运行结果

   若抓取 www.weislank.com

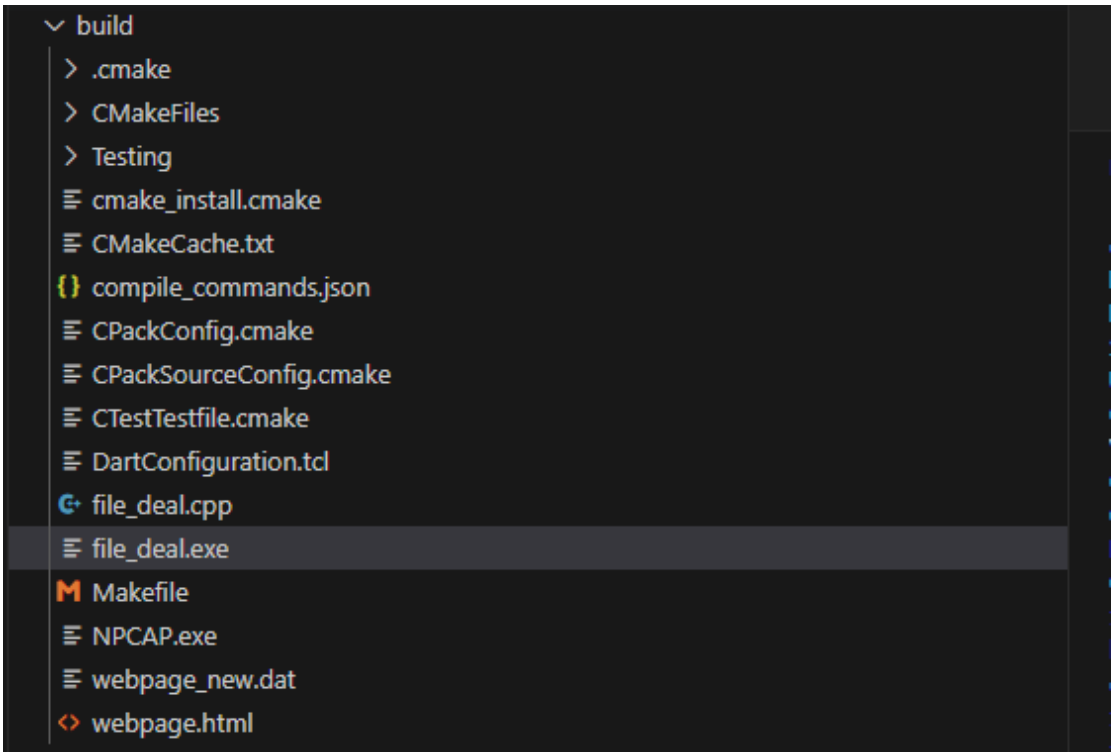
   NPCAP.exe 会生成一个二进制文件————》 webpage_new.dat

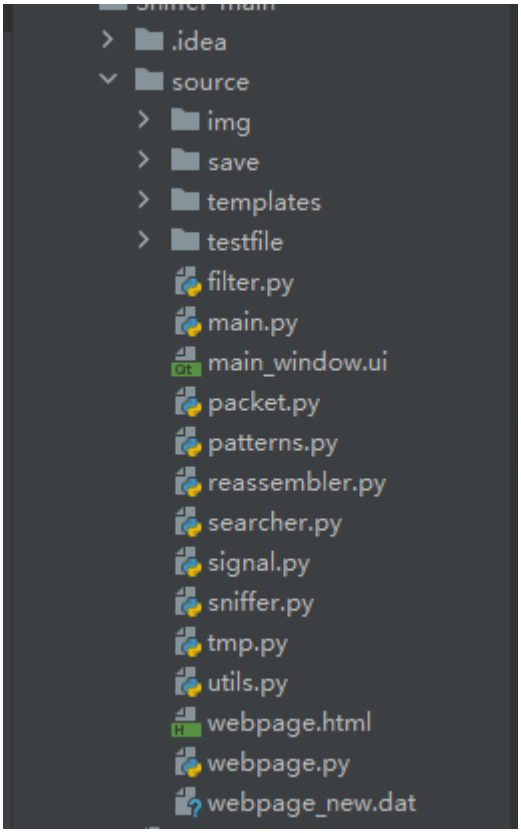
   将其复制到 decode.py 文件夹下，使用 decode.py 进行解压，编码后得到 html 网页->Webpage.html

   若抓取本地网页

   直接在 NPCAP 路径下生成 webpage.html，打开即可

```
data_len: 1460
HTTP/1.1 200 OK
Date: Sun, 18 Jun 2023 04:38:40 GMT
Server: Apache/2
Upgrade: h2,h2c
Connection: Upgrade, Keep-Alive
Vary: Accept-Encoding,User-Agent
Content-Encoding: gzip
Content-Length: 7283
Keep-Alive: timeout=2, max=100
Content-Type: text/html; charset=utf-8120.48.171.216:80 -> 10.249.43.35:50395
12:38:39.645125 len:1514
head_len: 54
data_len: 1460
120.48.171.216:80 -> 10.249.43.35:50395
12:38:39.645317 len:1514
head_len: 54
data_len: 1460
120.48.171.216:80 -> 10.249.43.35:50395
12:38:39.645317 len:1514
head_len: 54
data_len: 1460
120.48.171.216:80 -> 10.249.43.35:50395
12:38:39.645317 len:1514
head_len: 54
data_len: 1460
120.48.171.216:80 -> 10.249.43.35:50395
12:38:39.645317 len:313
head_len: 54
data_len: 259
120.48.171.216:80 -> 10.249.43.35:50395
12:38:41.645864 len:60
PS D:\cpp_project\npcap\build>
```




.idea
source
img
save
templates
testfile
filter.py
main.py
main_window.ui
packet.py
patterns.py
reassembler.py
searcher.py
signal.py
sniffer.py
tmp.py
utils.py
webpage.html
webpage.py
webpage_new.dat

```
Address   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  Dump
00000000  1f 8b 08 00 00 00 00 00 00 03 ed 3d 6b 53 1b 47  .?.......?kS.G□□
00000010  b6 df 53 b5 ff 61 a2 a4 16 48 d0 63 f4 00 e1 00  哆S?aiv.H衍??□□□□
00000020  fb 48 f6 de 9b aa ad 4d ee 8d 77 f7 de eb f5 a6  鶤鲛洩璇?w鬓膈?□
00000030  e6 d1 82 89 25 8d 56 33 32 76 b2 54 09 13 0c d8  嫜軌%.V32v瞍...?
00000040  bc 6c 63 fc 00 02 d8 c6 b0 4e 0c 7e 9b a7 f9 31  糽c?.仄瘖.~泠?□□
00000050  ab 1e 49 9f f2 17 ee e9 e9 91 34 4f 49 80 13 ef  ?I燆.鉍閼4OI€.?□
00000060  dd 8d ca 65 66 7a 4e 9f 3e 7d 5e 7d fa 4c eb a8  ?蔱fzN?}^}鶯毹□□
00000070  fb ed 8f 3e f9 f0 e4 ff 7c fa 1b a6 5f 4d 25 7b    .>  ?|?  M%{□□
00000080  7f f6 56 b7 f1 97 5c 21 4e 84 2b 86 e9 4e 21 95  .鮌否梊!N?嗛N!?□
00000090  63 84 7e 2e ab 20 b5 c7 f7 fb 93 ff e6 8f fb 08  c剆.?登鼷???□□□□
000000a0  ac de de af aa 19 3f fa 4b 4e 3a db e3 fb 6f ff    薤??鷊N:坫鷁  
000000b0  ef 7f e5 ff 50 4e 65 38 55 e2 93 c8 c7 08 72 5a  ??PNe8U鈸惹.rZ□□
000000c0  45 69 e8 24 a1 1e 24 f6 21 bd 9b 2a a9 49 d4 8b  Ei??$?經*ⓒ誏□□□
000000d0  d7 87 b4 b9 4d 3c f2 08 4f ae e0 8d 9b da dc 76  讎垂M<?O  .涄躺□
000000e0  71 ff 4a f1 db 5b dd 41 fa bc 82 3f cd a5 50 8f  q  J褊[軦  ?庭P.
000000f0  ef 0c 3a 3f 20 67 45 c5 84 d2 0b 81 cf d6 55 44  ?:? gE艅?.现UD□□
00000100  8a 90 95 32 aa 24 a7 3d 7a 7f bf 37 36 80 24 25  ????z.?6€$%□□□□□
00000110  c9 a5 cf 7c bf 37 ae dd dc 2c 6c 3d 28 ec ac e1  丧蟶?  ?l=(飕?□□
00000120  6b 43 a5 83 85 f2 6c 1e cf ef e3 8b 23 f8 de 5c  kC  咈l.巷鈃#  \
00000130  38 14 c2 f7 6e c3 53 d2 b2 f1 a2 78 fb 6b b8 28  8.瞒n胚也瘼x鹂?□
00000140  df 1a c1 13 73 00 5c 1c 9f f8 7e 6f a2 b0 75 4d  ??s.\.熿~o  uM□□
00000150  7b ba 5e d8 99 2a 6c 5f 2d 6c 5f 2b ce 0c 97 56  {篰貥*l_-l_+?栜□
00000160  a0 f1 36 45 08 00 e5 d1 d1 e2 ec 7a 61 ef 66 e9  .?E..逖砚靭a飂?□
00000170  d9 3a c1 30 b5 a4 2d ce 68 73 b7 f4 be ab da f2  ??丹-蝖s肤精隍□□
00000180  5e e9 d9 26 de df d5 c6 d7 a0 23 c0 94 47 a7 b5  ^橘&捱掌?#罃GY□□
00000190  e5 51 3c 7a 91 f4 cd df 2e 1d 8c 6a 13 43 c5 dd  錛<z戶瓦..寀.C泡
000001a0  87 d0 b7 b8 3c 8b a7 af 68 97 ae e3 91 55 6d 71  囆犯<嫥痟棶銘Umq
000001b0  b5 b0 35 a9 cd 91 76 6d 7a 06 4f 8e e2 b1 17 e5  蛋5丨憊mz.O庳??□□
000001c0  fc 32 7e b9 5a 1a df 81 be da ec 76 69 7c 93 92  ?~築.?沮靨i|搾□□
000001d0  c1 68 f9 b5 d2 f0 fe df f3 17 f0 e8 c3 e2 62 1e  羑  茵  ?痂免b.□
000001e0  90 16 57 be 83 9e 74 3c 80 c6 1b df 68 c3 cb d0  ..W緝瀟<€?遠盟?□
000001f0  52 d8 7d 09 34 91 fe 2f 1f e3 e9 1b 55 5e d4 a6  R貆.4扆/.汩.U^驭
00000200  77 30 83 2f ad 17 c7 1e e3 d5 fd 52 7e 84 4c 69  w0???阏齊~凤i□□□
```

威斯兰卡官方网站 威斯兰卡官方网站

- 网站首页
- 关于我们
- 产品展示
- 防伪查询
- 公司资讯
- 联系我们

banner
banner
banner

防伪查询

product

产品展示

乒乓球拍 底板 套胶 配附件
乒乓球拍

乒乓球拍

乒乓球拍

乒乓球拍

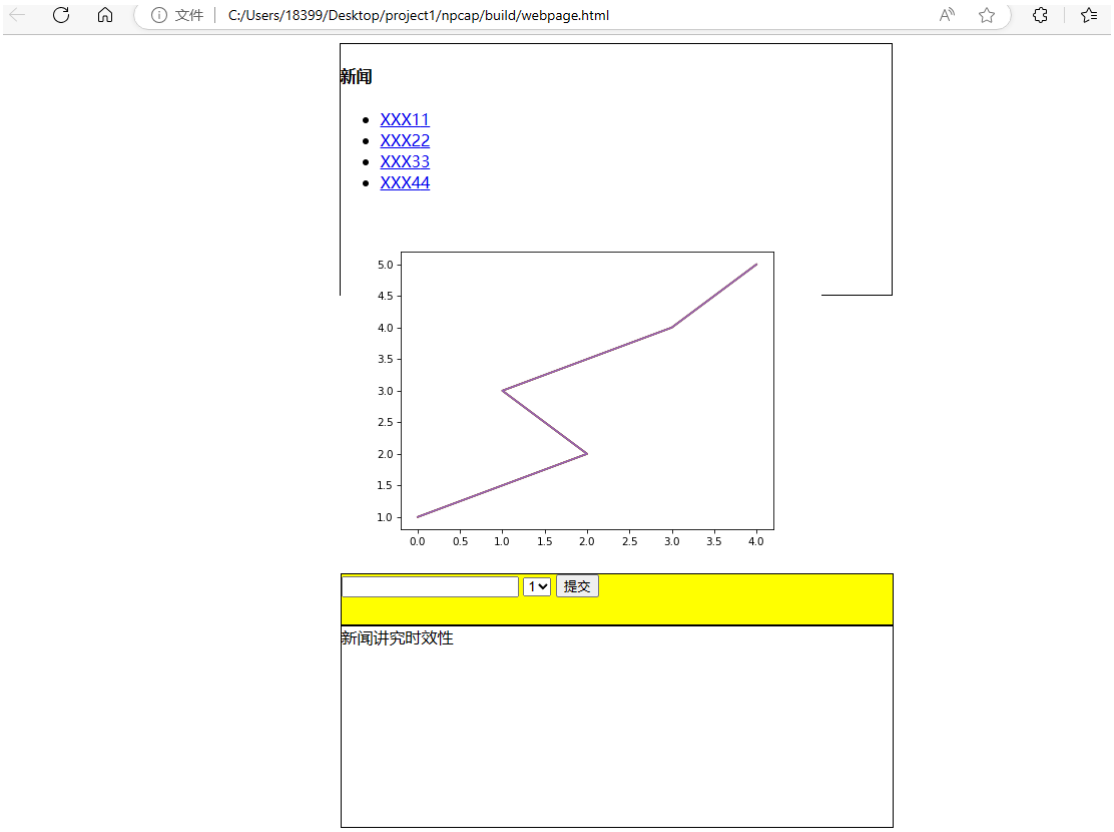
乒乓球拍

乒乓球拍

乒乓球拍

乒乓球拍

乒乓球拍


文件 | C:/Users/18399/Desktop/project1/npcap/build/webpage.html
新闻
XXX11
XXX22
XXX33
XXX44
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
1
提交
新闻讲究时效性