Task 2.3：Wireshark抓包初步

1#Wireshark的安装使用

1## linux(以fedora为例)：

dnf install wireshark -y

```
root@localhost ~# dnf install wireshark -y
```

```
Installed:
  wireshark.x86_64 2.2.8-1.fc26        compat-lua-libs.x86_64 5.1.5-7.fc26    libsmi.x86_64 0.4.8-18.fc
  wireshark-cli.x86_64 2.2.8-1.fc26

Complete!
```

安装wireshark图形界面

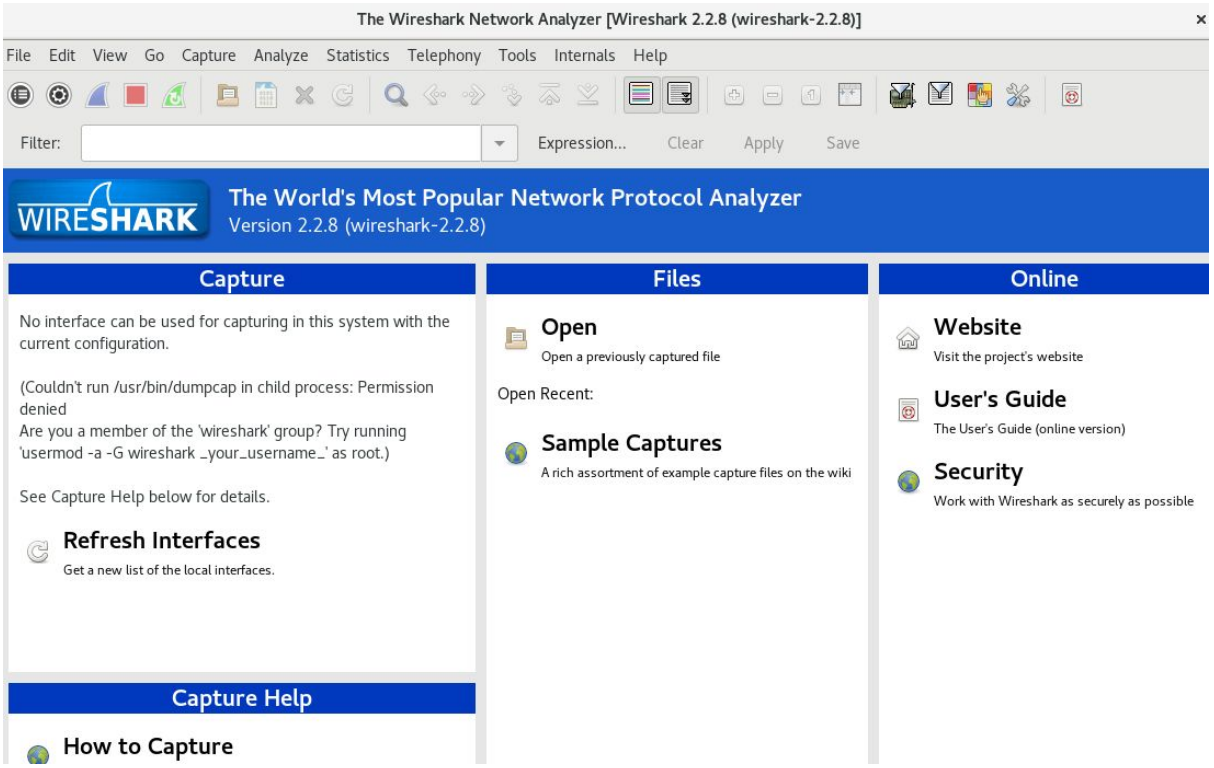
```
root@localhost ~# dnf install wireshark-gnome -y
```

```
Installed:
  wireshark-gtk.x86_64 2.2.8-1.fc26    jack-audio-connection-kit.x86_64 1.9.10-8.fc26
  libxml++.x86_64 2.40.1-3.fc26        portaudio.x86_64 19-24.fc26

Complete!
```

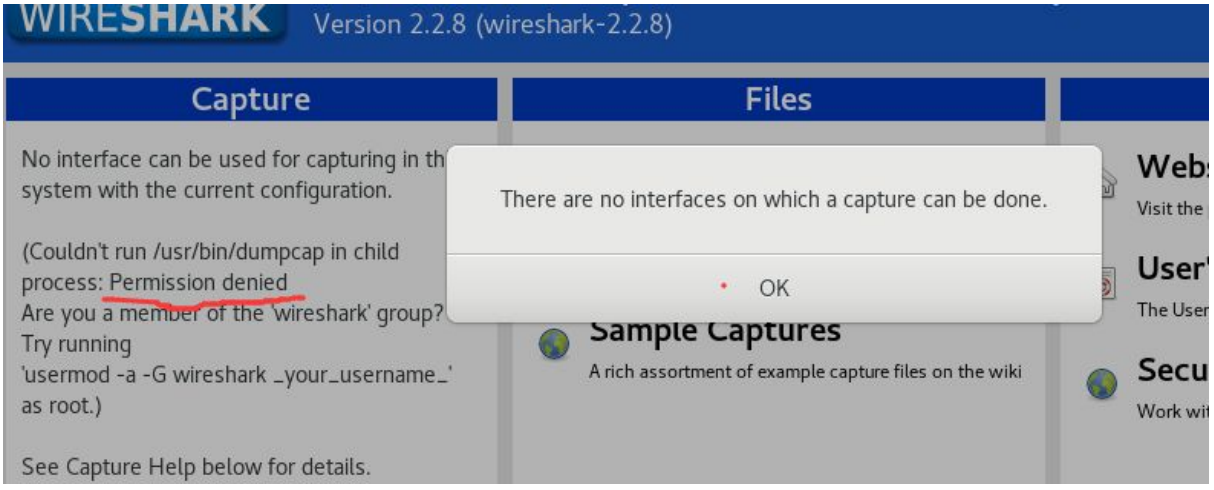
在activities里打开wireshark如下：

出现页面



安装即完成

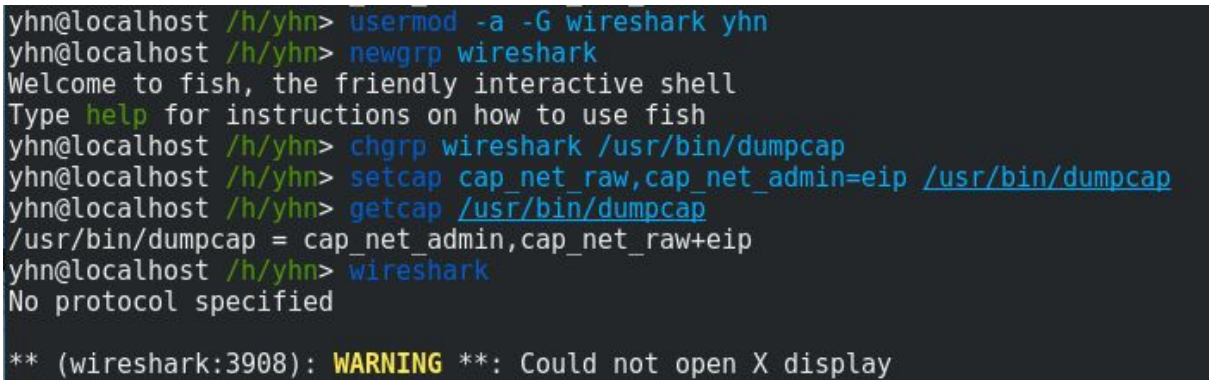
此时root用户可以直接打开wireshark进行抓包，但是普通用户没有执行权限不能进行操作



这时需要我们修改一下权限

首先创建wireshark用户组
groupadd wireshark

整个提权的操作截图如下：



添加自己进入组内
usermod -a -G wireshark xxx

此时可以重新登陆或者执行命令来使得新建的组起作用
newgrp wireshark

wireshark会使用到 /usr/bin/dumpcap 这个执行程序，修改它的组权限
chgrp wireshark /usr/bin/dumpcap

使wireshark组在运行dumpcap程序具有执行权限
chmod 750 /usr/bin/dumpcap

使用setcap授予能力
setcap cap_net_raw,cap_net_admin=eip /usr/bin/dumpcap

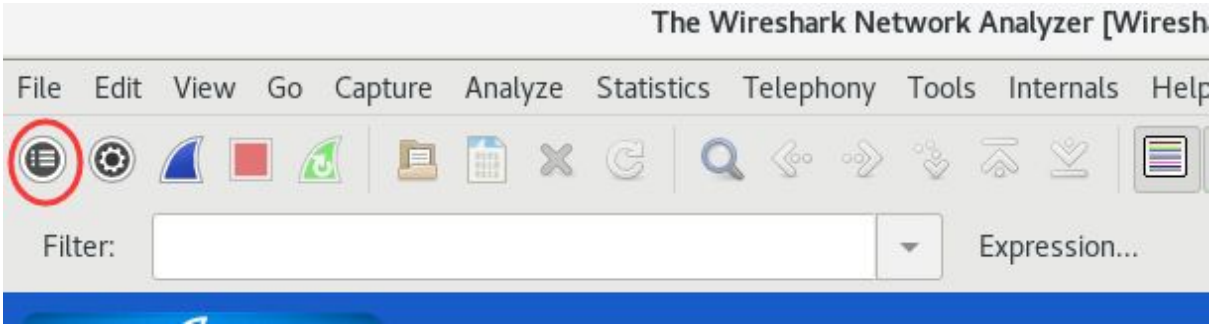
使用getcap验证
getcap /usr/bin/dumpcap

最后，普通用户就可以打开wireshark进行抓包操作了

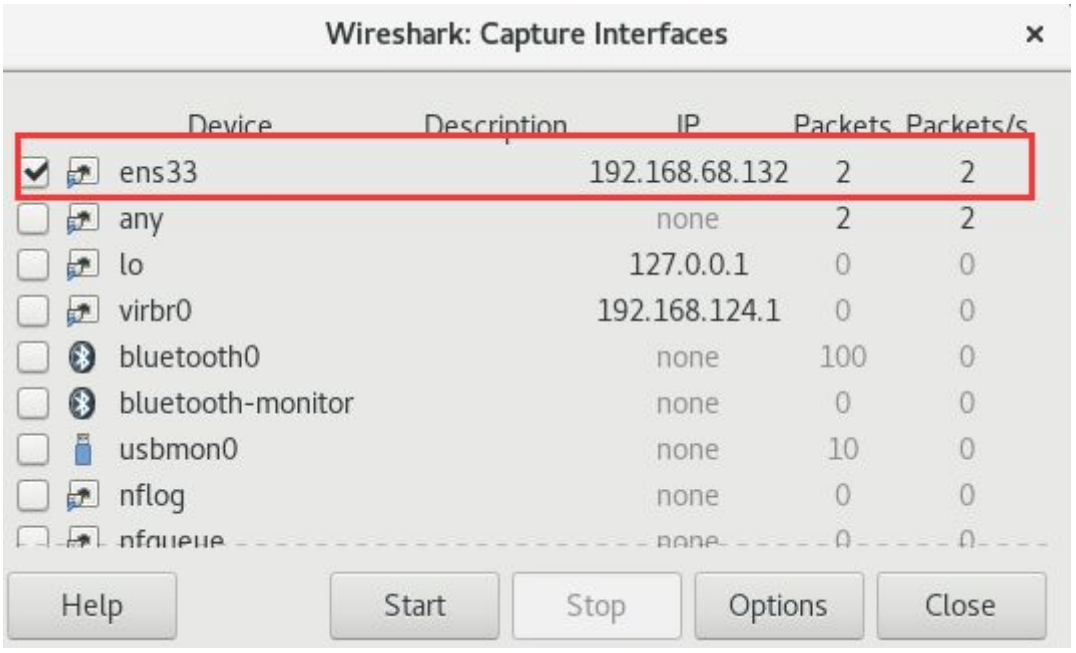
2##windows/mac OS ：
官网下载安装打开，与常用软件一致

2#结果的查看、导出、复制

抓取数据包



选择网卡ens33

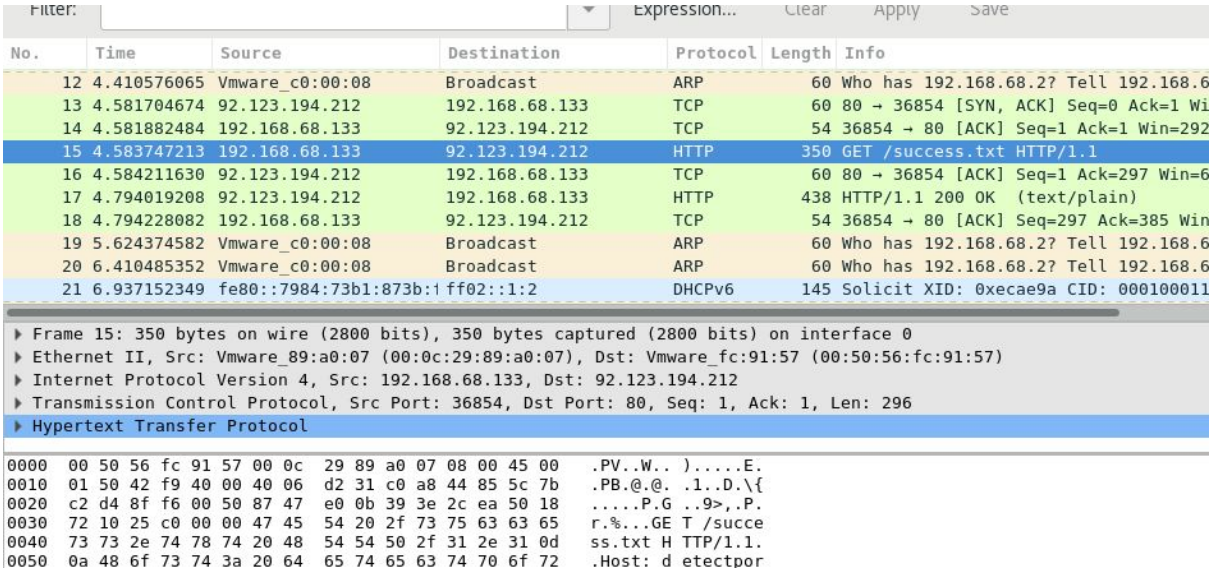


start，抓取到的数据包如下：

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.68.1	192.168.68.132	SSH	158	Client: Encrypted packet (len=104)
2	0.000662748	192.168.68.132	192.168.68.1	SSH	90	Server: Encrypted packet (len=36)
3	0.000817471	192.168.68.132	192.168.68.1	SSH	106	Server: Encrypted packet (len=52)
4	0.000938635	192.168.68.1	192.168.68.132	TCP	60	17727 → 22 [ACK] Seq=105 Ack=89 Win=6
5	0.000943207	192.168.68.132	192.168.68.1	SSH	114	Server: Encrypted packet (len=60)
6	0.051027249	192.168.68.1	192.168.68.132	TCP	60	17727 → 22 [ACK] Seq=105 Ack=149 Win=
7	0.086696538	192.168.68.1	192.168.68.132	SSH	90	Client: Encrypted packet (len=36)
8	0.087166995	192.168.68.132	192.168.68.1	SSH	106	Server: Encrypted packet (len=52)
9	0.087260399	192.168.68.132	192.168.68.1	SSH	114	Server: Encrypted packet (len=60)
10	0.087588241	192.168.68.132	192.168.68.1	SSH	114	Server: Encrypted packet (len=60)
11	0.087593708	192.168.68.1	192.168.68.132	TCP	60	17727 → 22 [ACK] Seq=141 Ack=261 Win=
12	0.138153241	192.168.68.1	192.168.68.132	TCP	60	17727 → 22 [ACK] Seq=141 Ack=321 Win=
13	0.254635735	192.168.68.1	192.168.68.132	SSH	90	Client: Encrypted packet (len=36)
14	0.255980916	192.168.68.132	192.168.68.1	SSH	98	Server: Encrypted packet (len=44)
15	0.256102277	192.168.68.132	192.168.68.1	SSH	90	Server: Encrypted packet (len=36)

其中每列代表的意思依次是
序号
抓到该包的时间
包发出的来源ip地址
包要发到的目的ip地址
使用的协议类型整个包的长度
包的具体信息

1##双击即可查看

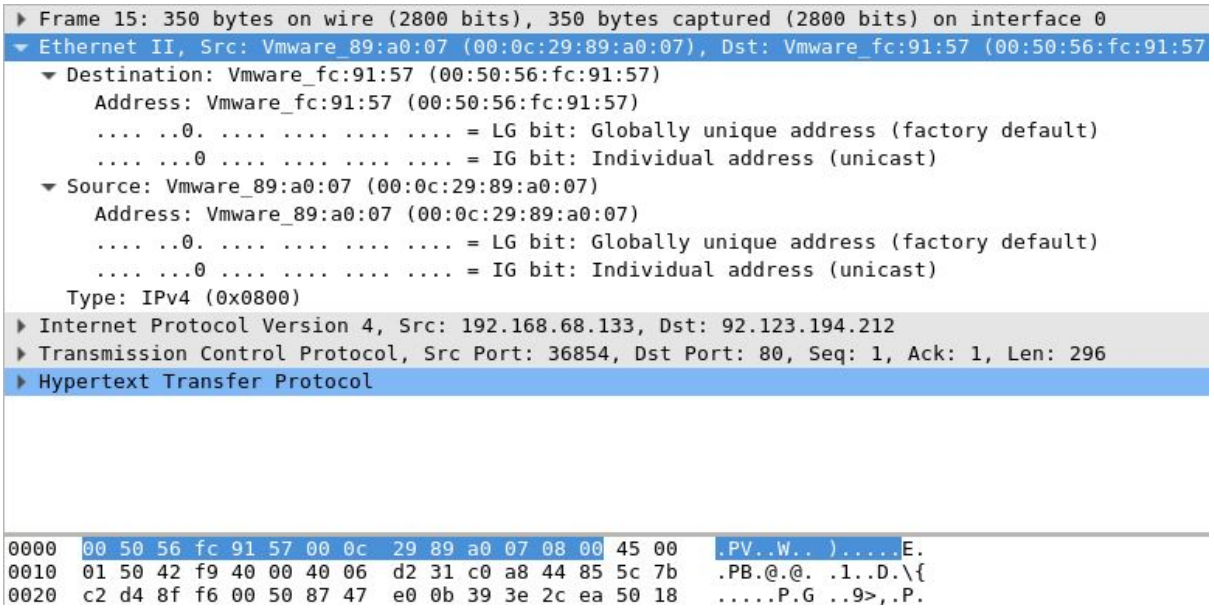


第一层：物理层的数据帧概况

```
Frame 15: 350 bytes on wire (2800 bits), 350 bytes captured (2800 bits) on interface 0
    Interface id: 0 (ens33)
    Encapsulation type: Ethernet (1)
    Arrival Time: Nov 19, 2017 15:15:22.032478928 CST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1511075722.032478928 seconds
    [Time delta from previous captured frame: 0.001864729 seconds]
    [Time delta from previous displayed frame: 0.001864729 seconds]
    [Time since reference or first frame: 4.583747213 seconds]
    Frame Number: 15
    Frame Length: 350 bytes (2800 bits)
    Capture Length: 350 bytes (2800 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp:http]
    [Coloring Rule Name: HTTP]
    [Coloring Rule String: http || tcp.port == 80 || http2]
```

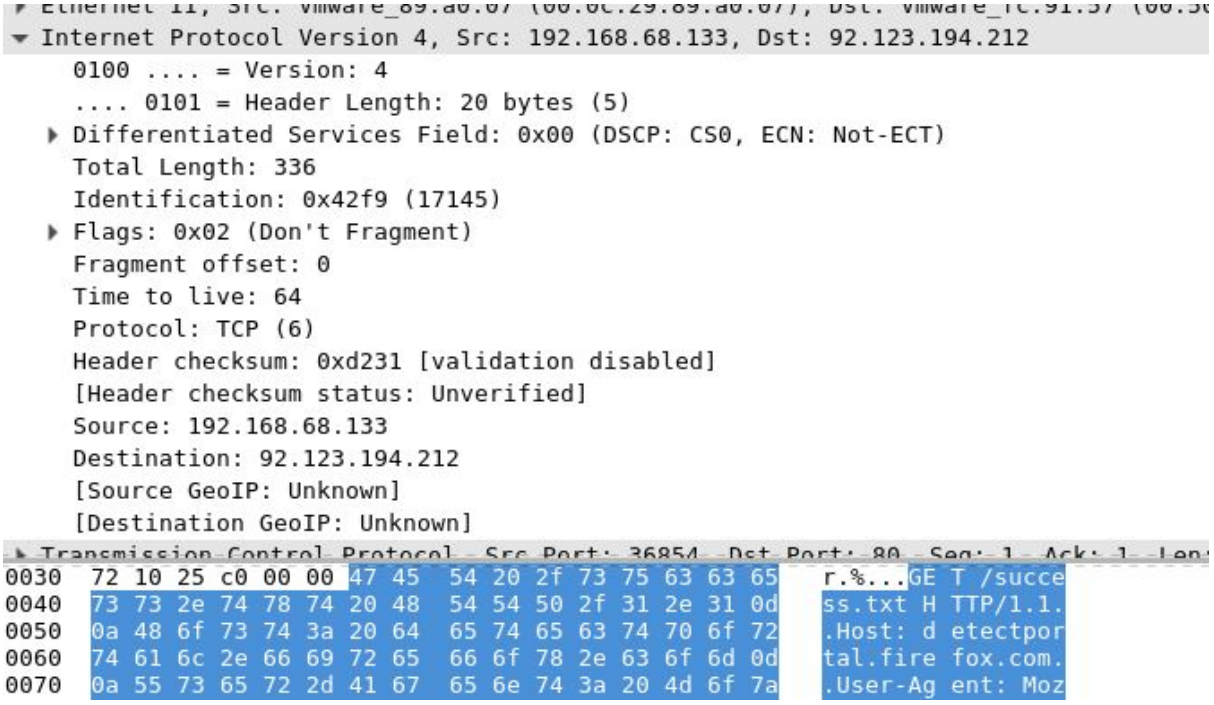
从上到下依次为：
15号帧，线路350字节，实际捕获350字节
接口id为0
封装类型
捕获时间
此包与前一包的时间间隔
此包与第一帧的时间间隔
帧序号：15
帧长度
捕获长度
此帧是否做了标记：否
此帧是否被忽略：否
帧内封装协议层次结构
着色标记的协议名称
着色规则显示的字符串

第二层：数据链路层以太网帧头部信息



从上到下依次为：
目标MAC地址
源MAC地址

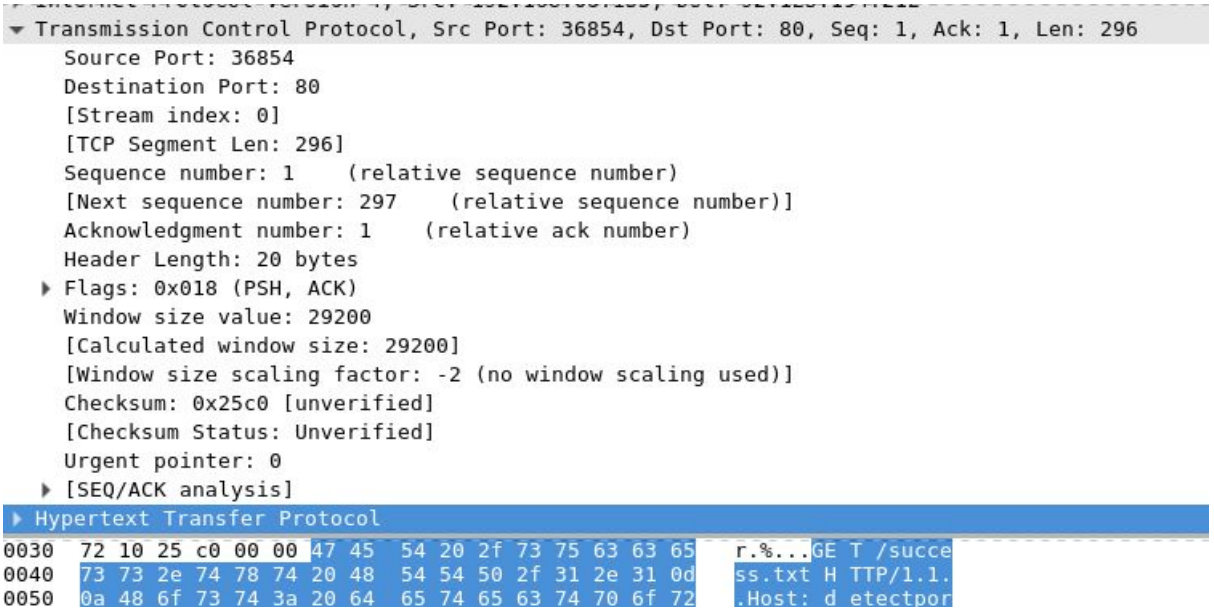
第三层：网络层IP包头部信息



从上到下依次是：
互联网协议：IPV4
IP包头长度：20bytes
差分服务字段
IP包总长度：336
标志字段
标记字段
偏移量

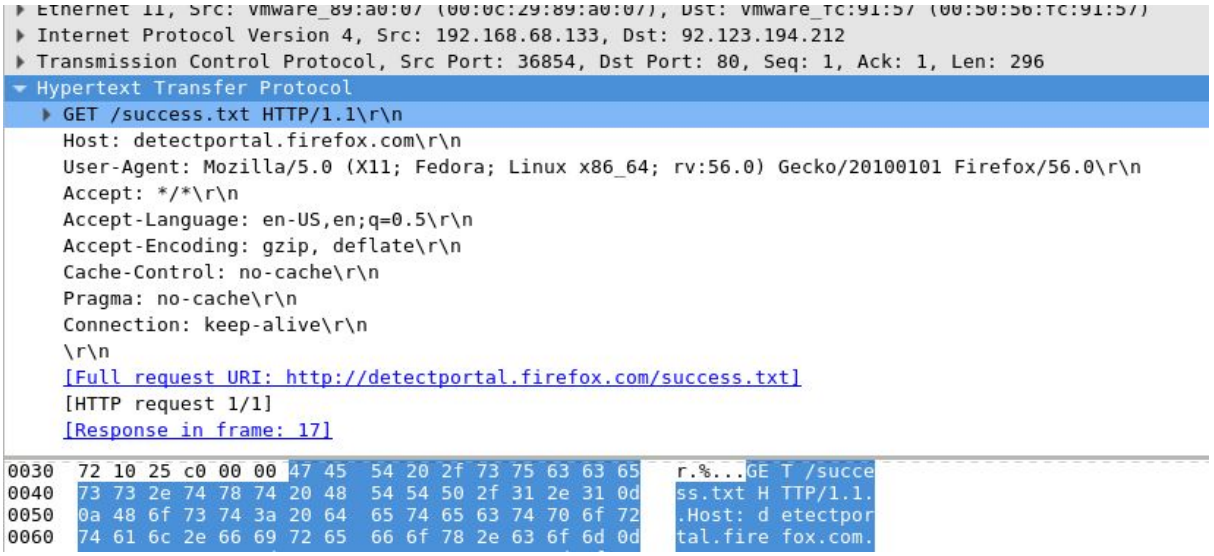
生存周期
包内封装的上层协议：TCP
头部数据校验和
源IP地址
目标IP地址

第四层：传输层的数据段头部信息，此处是TCP协议

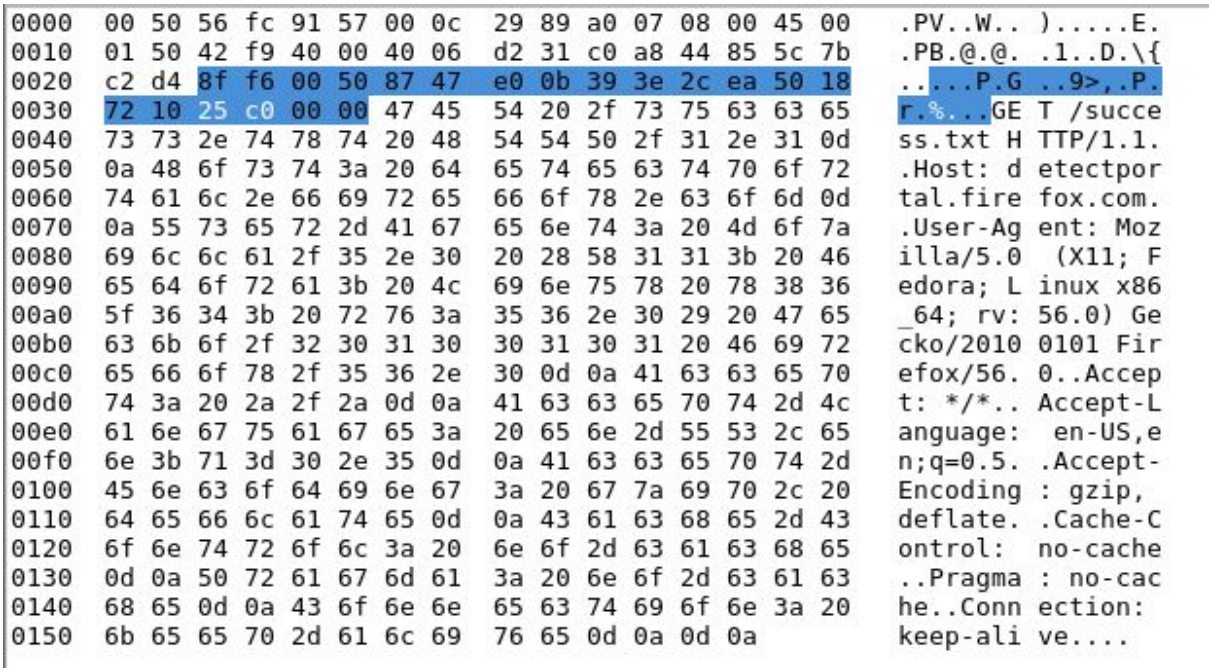


从上到下依次为：
源端口
目标端口
相对序列号
下一个序列号
确认序列号
头部长度
TCP标记字段
流量控制窗口大小
TCP数据段校验和

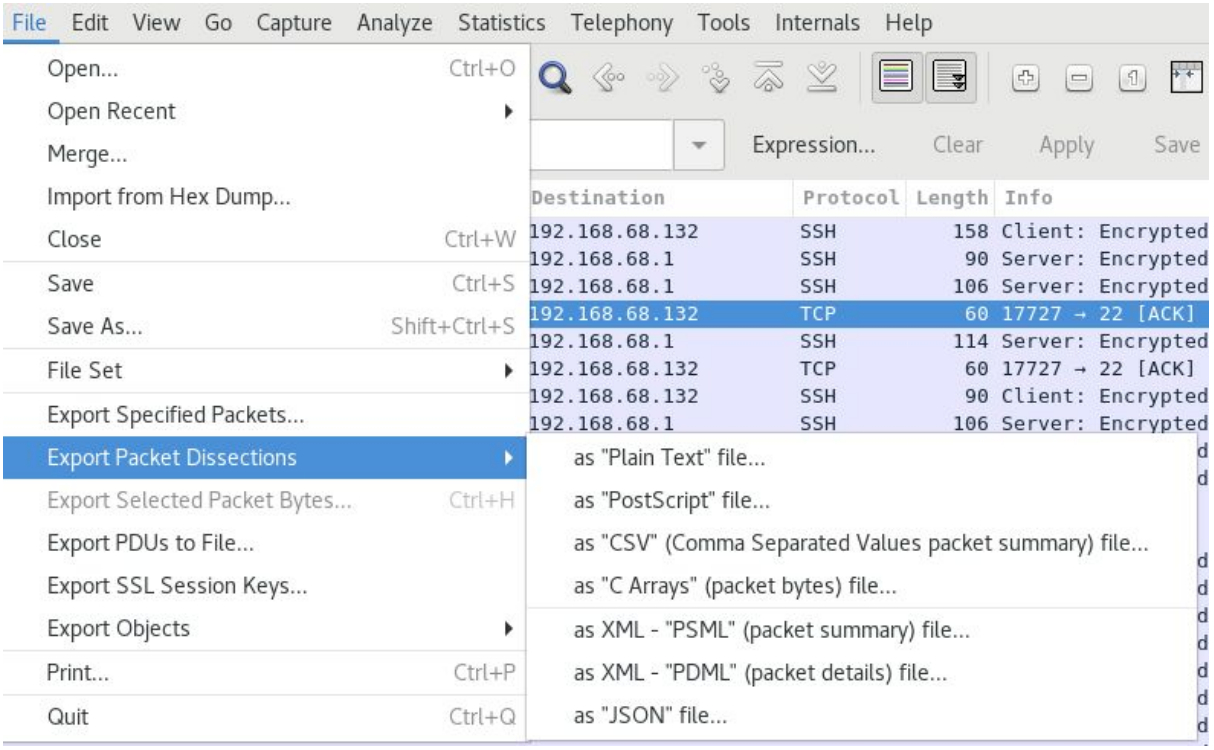
第五层：应用层的信息，此处是HTTP协议



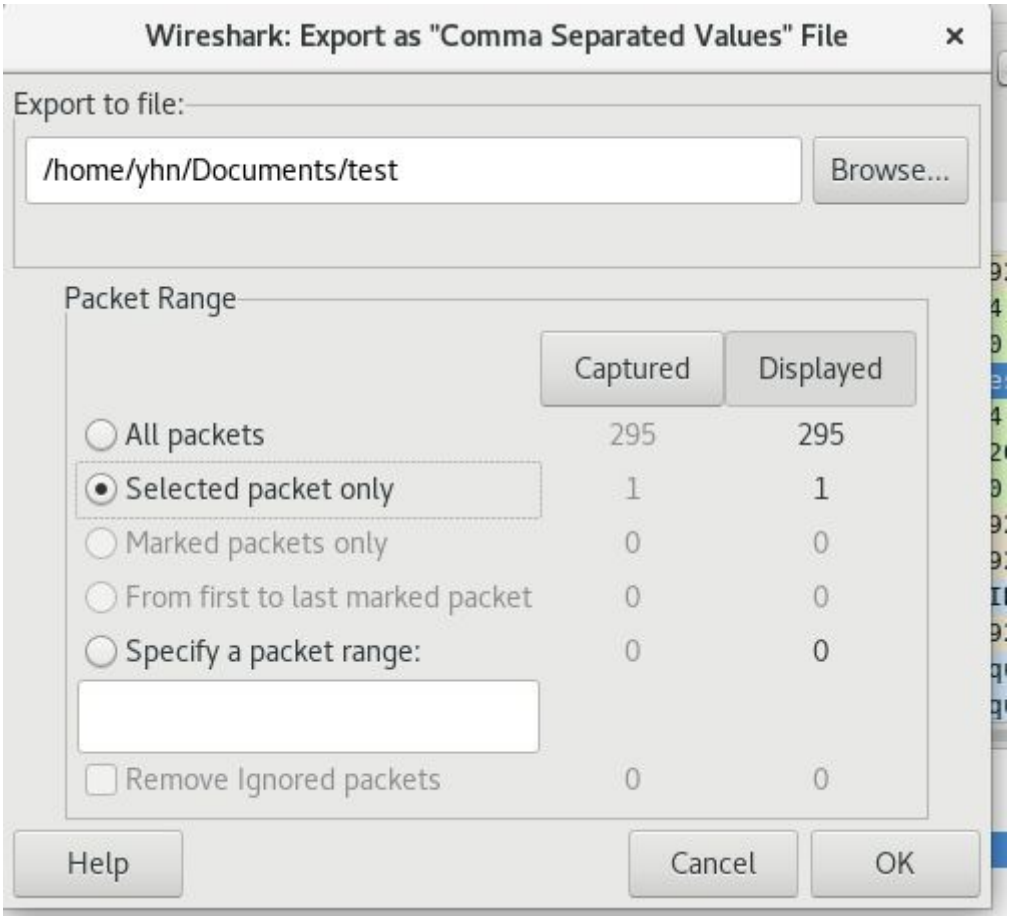
最后一栏是该数据包的字节，16进制，分别对应之前的数据包每层的解析内容。



2##导出可以自己选择：



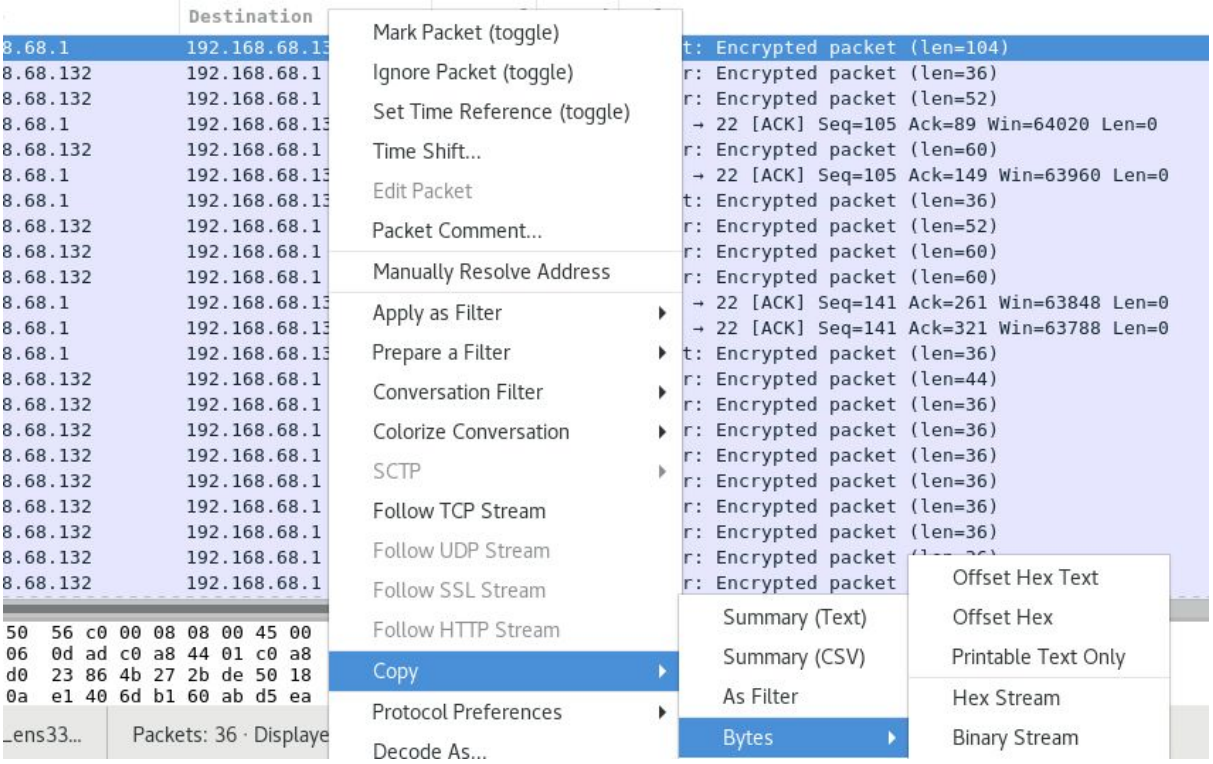
选择CSV格式



打开文件查看如下

```
yhn@localhost /h/y/Documents> cat test
"No.","Time","Source","Destination","Protocol","Length","Info"
"15","4.583747213","192.168.68.133","92.123.194.212","HTTP","350","GET /success.
txt HTTP/1.1 "
```

3##选择数据包右键copy即可复制



选择不同复制的结果都不同

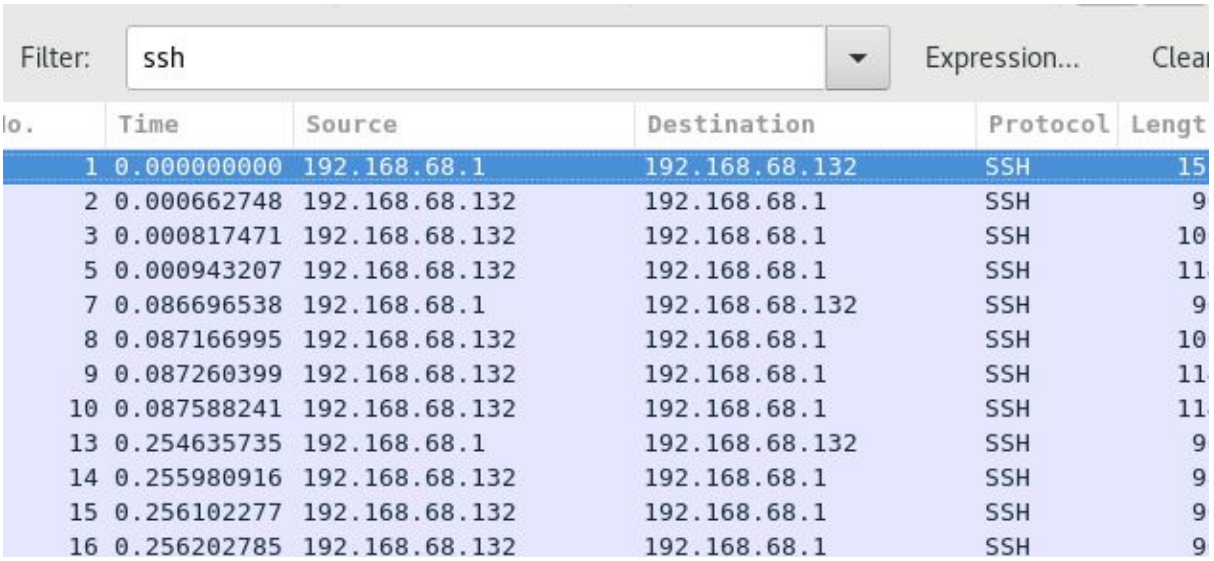
```
Hex Stream
005056fc9157000c2989a00708004500015035fd4000400616b7c0a84485407ca74a8d1e00506
fac934f65ae750187210ee360000474554202f737563636573732e74787420485454502f312e3
0a486f73743a20646574656374706f7274616c2e66697265666f782e636f6d0d0a557365722d4
656e743a204d6f7a696c6c612f352e3020285831313b204665646f72613b204c696e757820783
5f36343b2072763a35362e3029204765636b6f2f32303130303130312046697265666f782f353
300d0a4163636570743a202a2f2a0d0a4163636570742d4c616e67756167653a20656e2d55532
6e3b713d302e350d0a4163636570742d456e636f64696e673a20677a69702c206465666c61746
0a43616368652d436f6e74726f6c3a206e6f2d63616368650d0a507261676d613a206e6f2d636
68650d0a436f6e6e656374696f6e3a206b6565702d616c6976650d0a0d0a

Offset Hex Text
0000   00 50 56 fc 91 57 00 0c 29 89 a0 07 08 00 45 00  .PV..W..).....E.
0010   01 50 35 fd 40 00 40 06 16 b7 c0 a8 44 85 40 7c  .P5.@.@.....D.@|
0020   a7 4a 8d 1e 00 50 69 62 fa c9 34 f6 5a e7 50 18  .J...Pib..4.Z.P.
0030   72 10 ee 36 00 00 47 45 54 20 2f 73 75 63 63 65  r..6..GET /succe
0040   73 73 2e 74 78 74 20 48 54 54 50 2f 31 2e 31 0d  ss.txt HTTP/1.1.
0050   0a 48 6f 73 74 3a 20 64 65 74 65 63 74 70 6f 72  .Host: detectpor
0060   74 61 6c 2e 66 69 72 65 66 6f 78 2e 63 6f 6d 0d  tal.firefox.com.
0070   0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f 7a  .User-Agent: Moz
0080   69 6c 6c 61 2f 35 2e 30 20 28 58 31 31 3b 20 46  illa/5.0 (X11; F
0090   65 64 6f 72 61 3b 20 4c 69 6e 75 78 20 78 38 36  edora; Linux x86
00a0   5f 36 34 3b 20 72 76 3a 35 36 2e 30 29 20 47 65  _64; rv:56.0) Ge
00b0   63 6b 6f 2f 32 30 31 30 30 31 30 31 20 46 69 72  cko/20100101 Fir
00c0   65 66 6f 78 2f 35 36 2e 30 0d 0a 41 63 63 65 70  efox/56.0..Accep
```

3#结果过滤器的使用

常见应用：

ssh //只显示采用ssh协议的包
如图只显示ssh



No.	Time	Source	Destination	Protocol	Length
1	0.000000000	192.168.68.1	192.168.68.132	SSH	15
2	0.000662748	192.168.68.132	192.168.68.1	SSH	9
3	0.000817471	192.168.68.132	192.168.68.1	SSH	10
5	0.000943207	192.168.68.132	192.168.68.1	SSH	11
7	0.086696538	192.168.68.1	192.168.68.132	SSH	9
8	0.087166995	192.168.68.132	192.168.68.1	SSH	10
9	0.087260399	192.168.68.132	192.168.68.1	SSH	11
10	0.087588241	192.168.68.132	192.168.68.1	SSH	11
13	0.254635735	192.168.68.1	192.168.68.132	SSH	9
14	0.255980916	192.168.68.132	192.168.68.1	SSH	9
15	0.256102277	192.168.68.132	192.168.68.1	SSH	9
16	0.256202785	192.168.68.132	192.168.68.1	SSH	9

其他例如：
http or arp //只显示采用http或arp协议的包
snmp || dns || icmp //显示采用SNMP或DNS或ICMP协议的包
not arp //不显示采用arp协议的包
!tcp //不显示采用tcp协议的包
!(ip.addr == 192.168.0.1) //排除含有IP地址192.168.0.1的包
ip.addr == 192.168.0.1 //筛选出含有IP地址192.168.0.1的包
ip.src == 192.168.0.1 //筛选出源地址为192.168.0.1的包
ip.dst == 192.168.0.1 //筛选出目的地址为192.168.0.1的包
frame.len<=128 //只显示长度小于128字节的包
tcp.port == 80 //只显示含有80端口的包
tcp.dstport == 25 //只显示目的TCP端口号为25的包
tcp.port > 1024 //只显示端口号大于1024的包

其余详细语法可参考：https://www.wireshark.org/docs/man-pages/wireshark-filter.html

问题：

1##熟悉了解https的握手流程，使用wireshark进行抓包，并过滤出https流量。
2##将一次完整的https握手包保存到本地，csv格式。
3##分析https握手的过程，截图并分析具体的包内容及对应用途。

以上务必将邮件同时发送至250733748@qq.com和573698408@qq.com