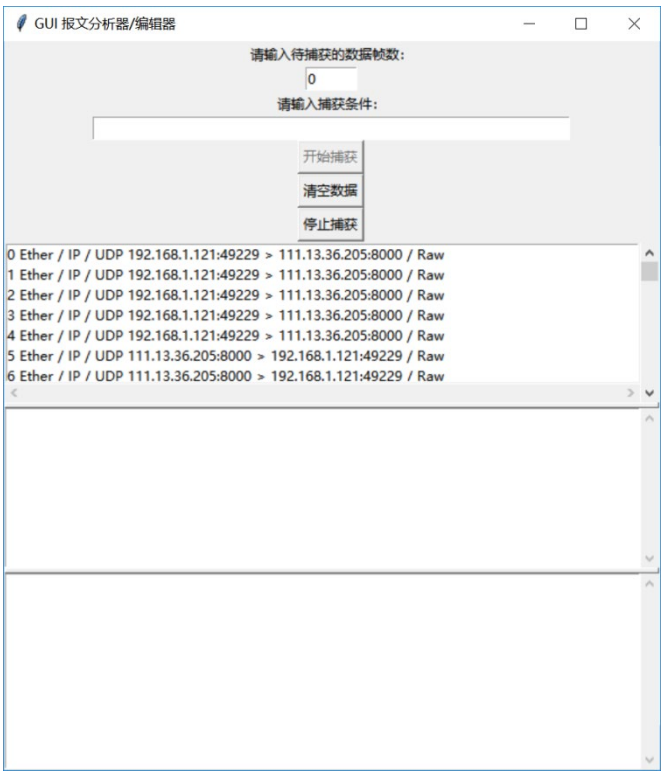
1.1 主要函数说明

函数名	功能
createWidgets	创建分析器界面
createControlWidgets	创建控制面板
createPDUSumPanedWindow	创建显示捕获报文的摘要的窗口
createPDUAnalysisPanedWindow	创建显示捕获报文分层解析的窗口
createPDUCodePanedWindow	创建显示捕获报文原始编码信息的窗口
start_sniff	启动捕获线程
PDU_sniff	捕获线程，捕获数据报，并调用回调函数
stop_sniff	停止捕获线程
clearData	清空捕获数据
split_condition、split_dulequal	分割条件的函数，分别是按空格和==分割
ip_monitor_callback	回调函数，根据筛选条件将符合的报文插入到 listbox 中
IP_headchecksum	检验和计算和验证
proto_IPcol	协议号代表的协议名称
intbin	十进制转二进制
choosedPDUAnalysis	对选择的报文，判断其协议调用不同的分析函数
choosedEtherPDUAnalysis	MAC 数据报分析
choosedIPPDUAnalysis	IP 数据报分析
choosedARPPDUAnalysis	ARP 数据报分析
tcpflag	获取 TCP 的 Flag 每一位的值
choosedTCPPDUAnalysis	TCP 数据报分析
choosedUDPPDUAnalysis	UDP 数据报分析
sendEtherFrame	MAC 数据报发送
sendIp	IP 数据报发送
sendArp	ARP 数据报发送
sendTcp	TCP 数据报发送
sendUdp	UDP 数据报发送

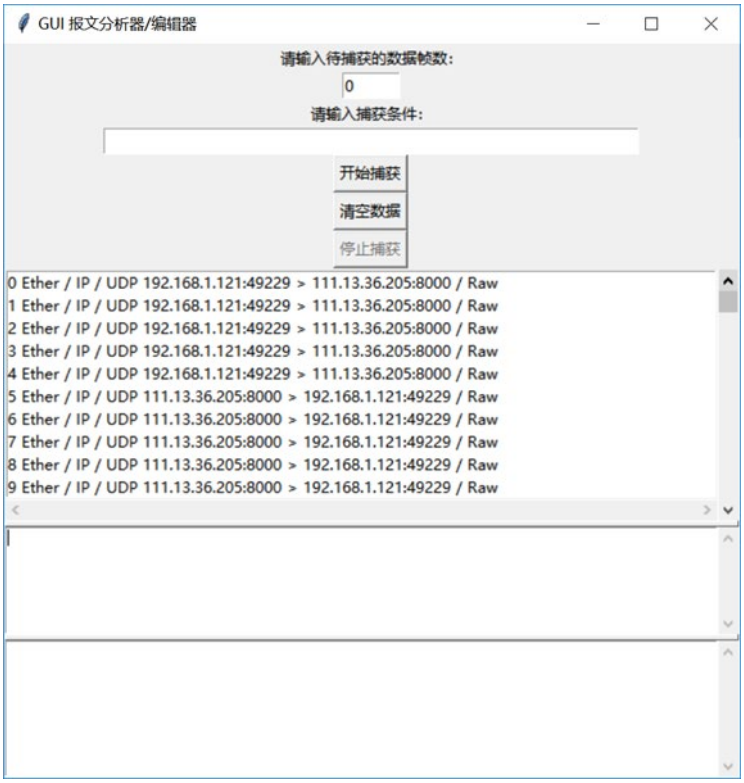
1.2 系统使用说明

1.2.1 在不限制捕获条件和捕获数目时

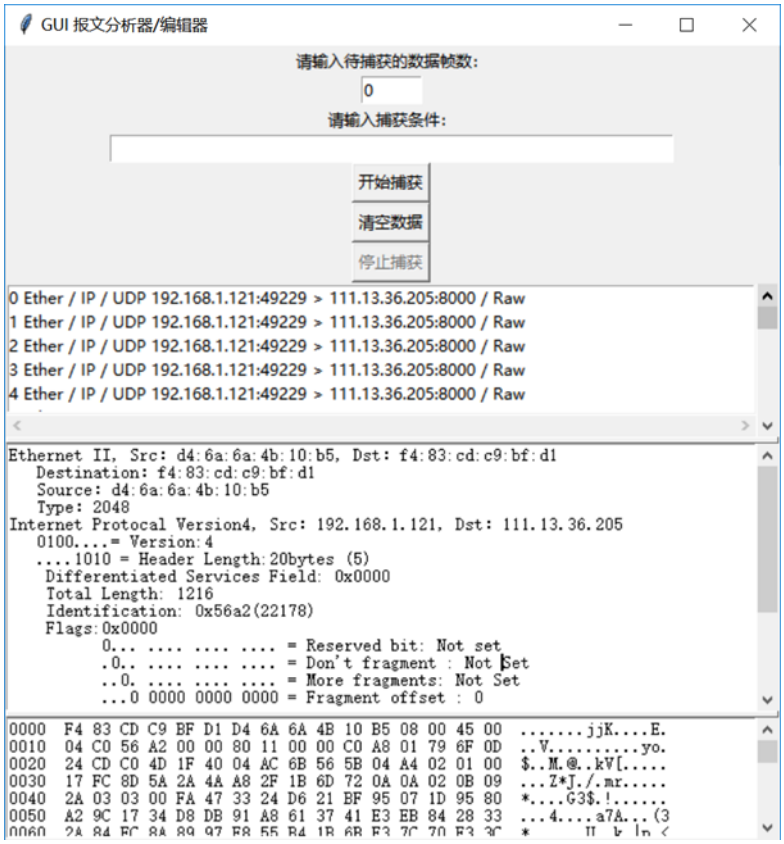
①点击“开始捕获”按钮，即可开始捕获数据报



②点击“停止捕获”按钮，即可停止捕获

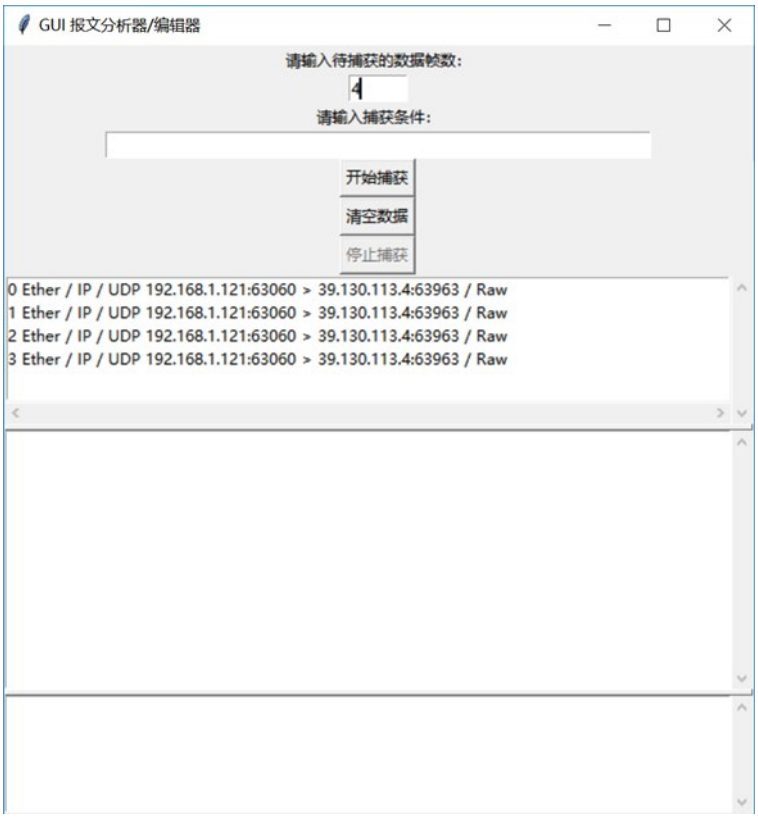


③点击 listbox 框，双击选中一条数据报，显示分析和出现数据



④点击“清空数据”按钮后，所有的数据均被清空

1.2.2 限制捕获数目时，捕获到设定捕获数量后，即不再插入 listbox

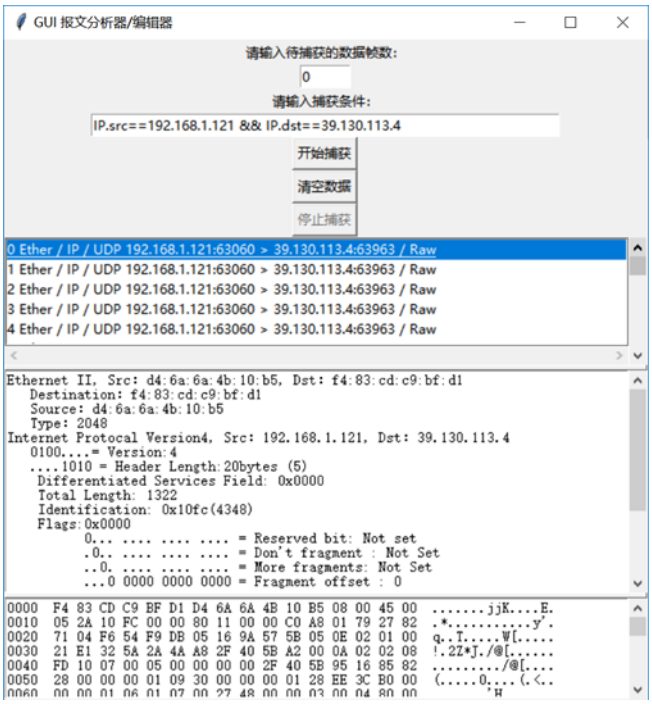


1.2.3 设置捕获条件时（有包含校验和的进行手动验证）

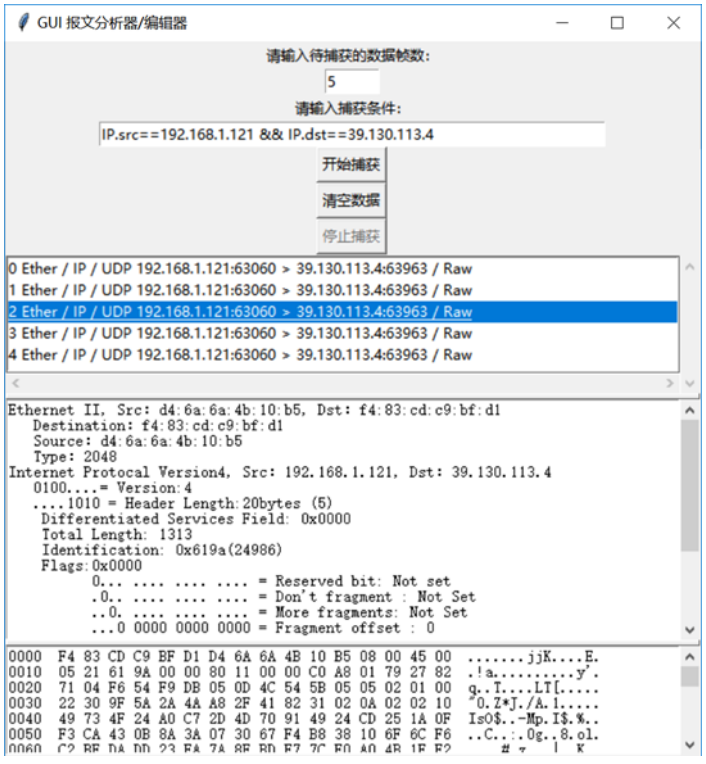
数据报类型	实现的筛选功能
Ether	源 MAC 地址
	目的 MAC 地址
IP	源 IP 地址
	目的 IP 地址
	协议类型
ARP	源 MAC 地址
	目的 MAC 地址
	源 IP 地址
	目的 IP 地址
	操作码
TCP	源端口
	目的端口
UDP	源端口
	目的端口

（1）IP 数据报（接收）

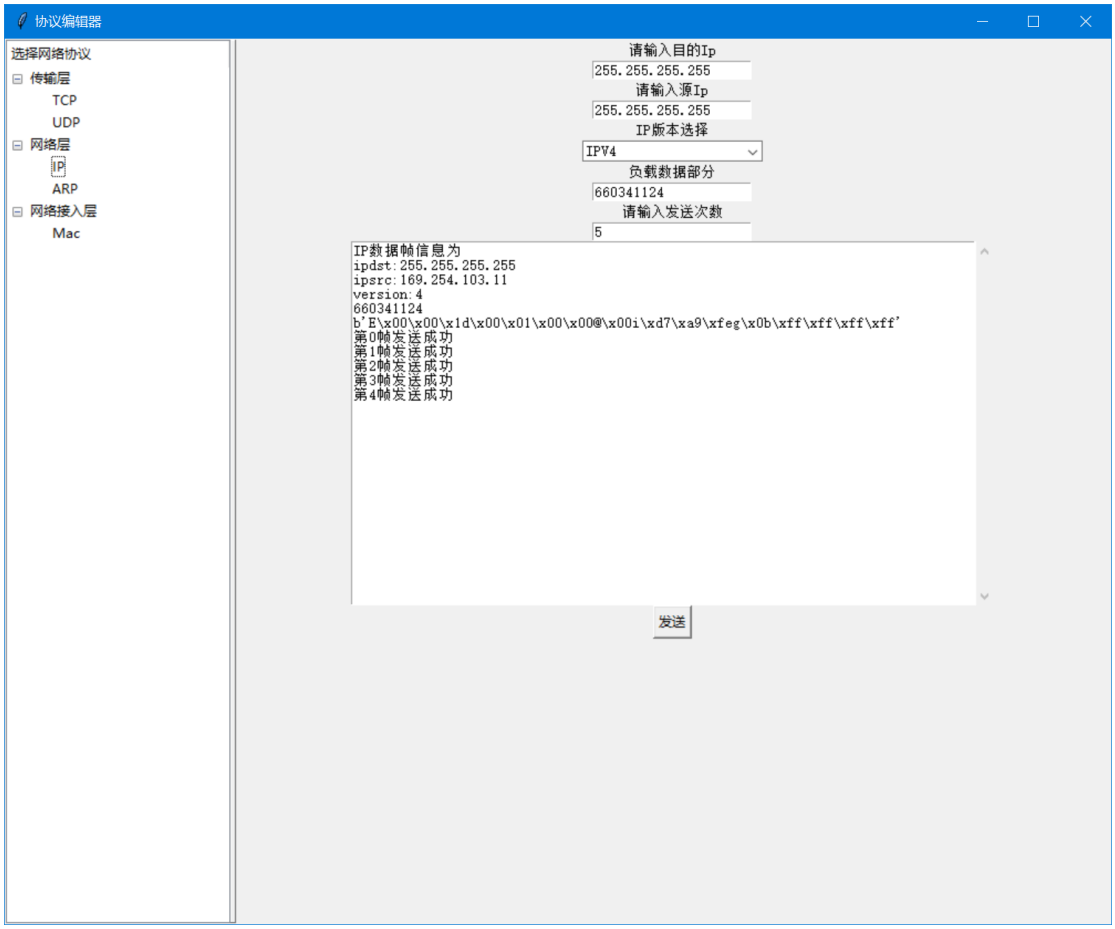
①设置捕获条件（注意：捕获条件分割依靠空格“ ”和双等号“==”）



②设置捕获条件的同时设置捕获数量（以下报文都可以实现）

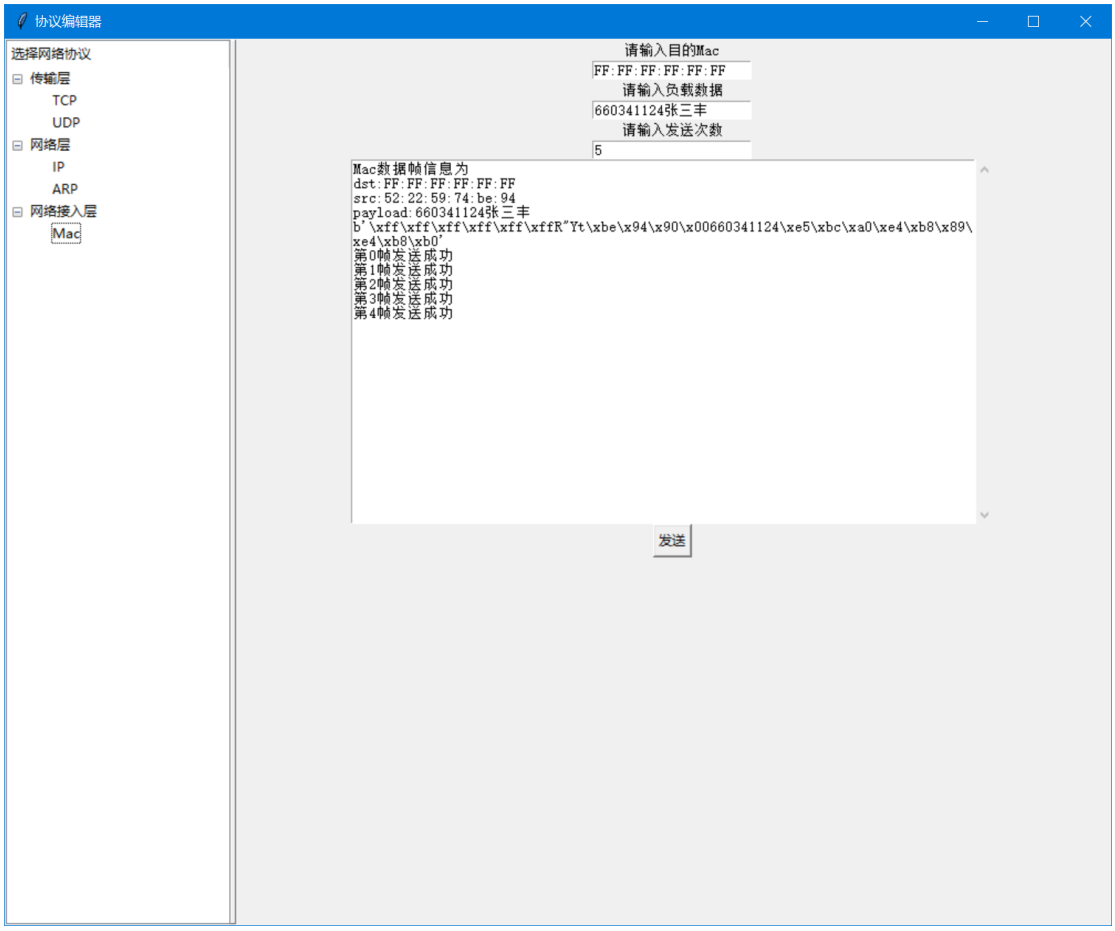


IP 数据报（发送）

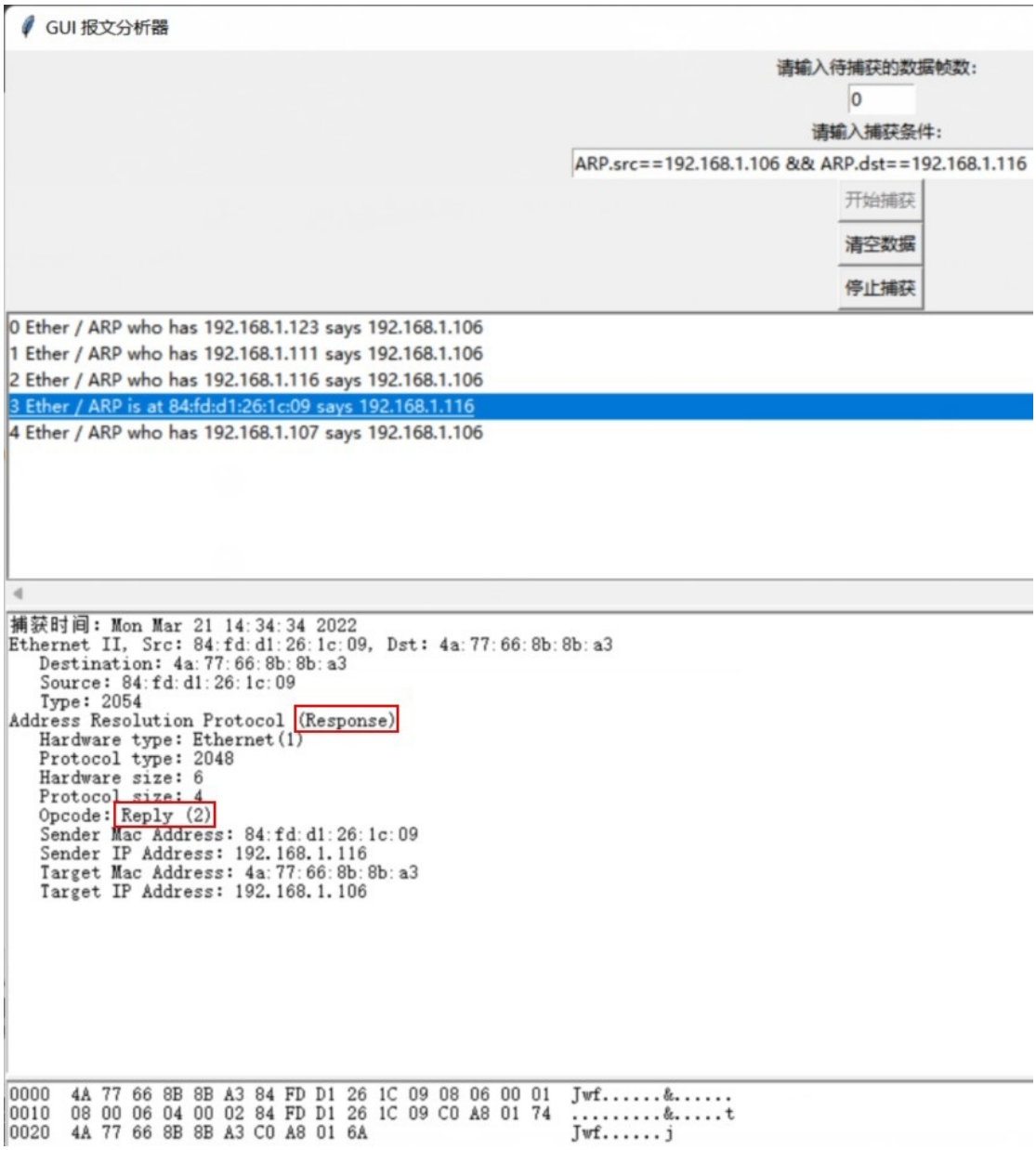


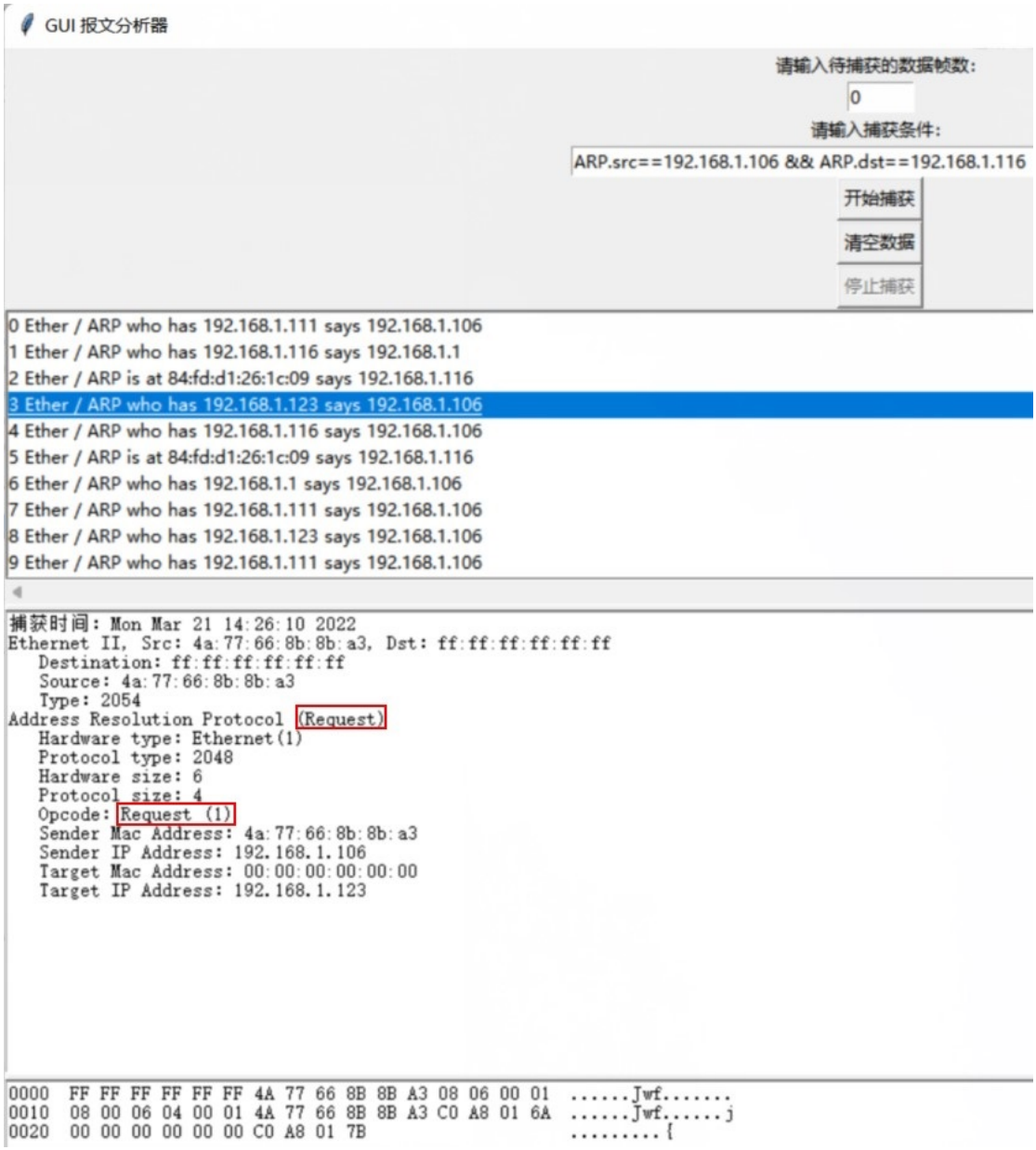
（2）Ether 数据报（MAC 数据报）（接收）

Ether 数据报（MAC 数据报）（发送）

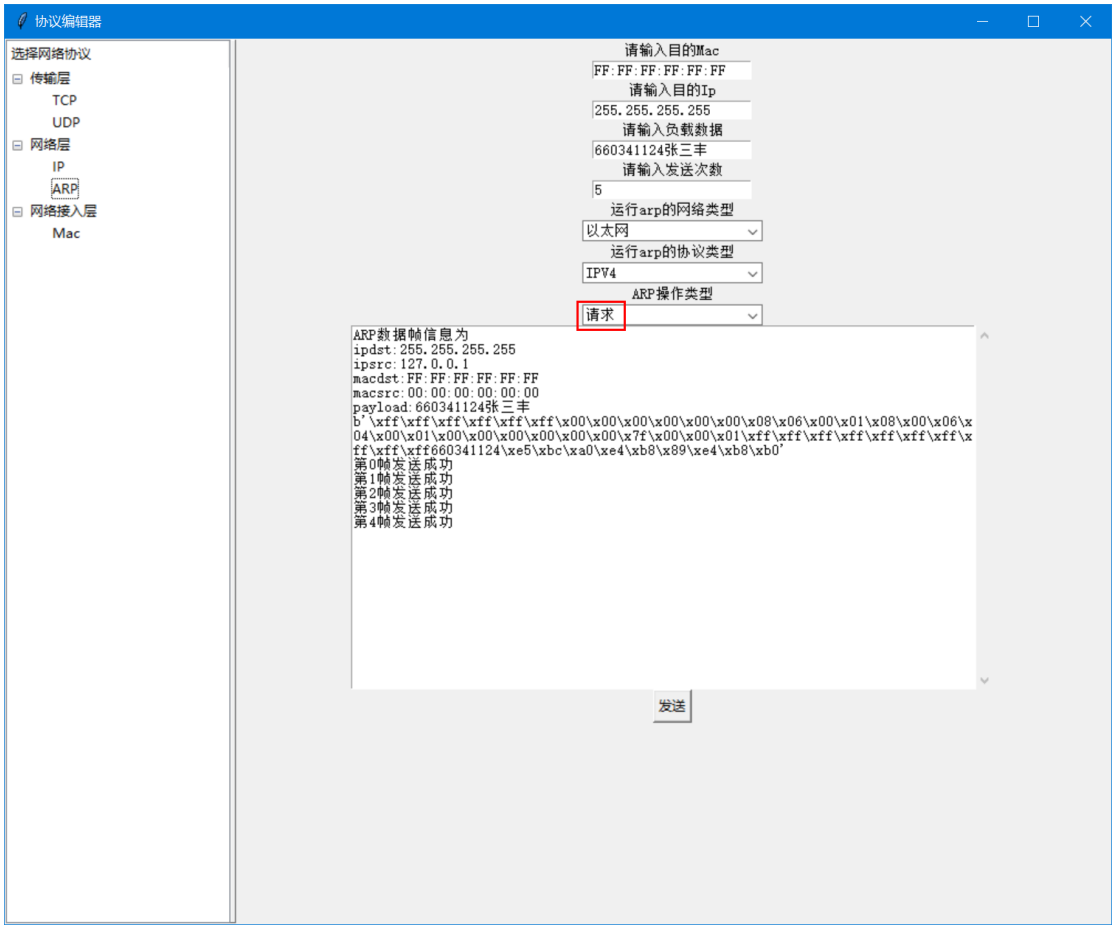


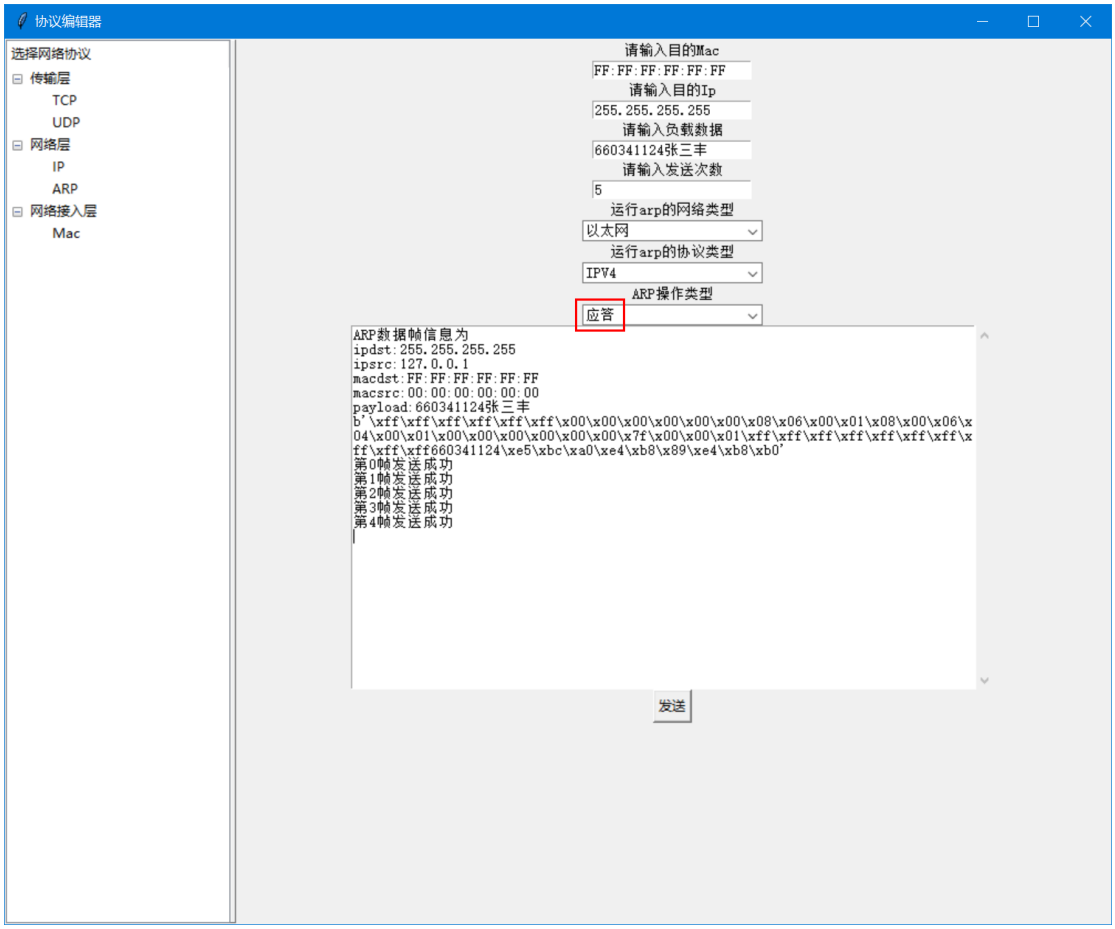
（3）ARP 数据报（接收）


GUI 报文分析器
请输入待捕获的数据帧数:
0
请输入捕获条件:
ARP.src==192.168.1.106 && ARP.dst==192.168.1.116
开始捕获
清空数据
停止捕获
0 Ether / ARP who has 192.168.1.123 says 192.168.1.106
1 Ether / ARP who has 192.168.1.111 says 192.168.1.106
2 Ether / ARP who has 192.168.1.116 says 192.168.1.106
3 Ether / ARP is at 84:fd:d1:26:1c:09 says 192.168.1.116
4 Ether / ARP who has 192.168.1.107 says 192.168.1.106
捕获时间: Mon Mar 21 14:34:34 2022
Ethernet II, Src: 84:fd:d1:26:1c:09, Dst: 4a:77:66:8b:8b:a3
Destination: 4a:77:66:8b:8b:a3
Source: 84:fd:d1:26:1c:09
Type: 2054
Address Resolution Protocol (Response)
Hardware type: Ethernet(1)
Protocol type: 2048
Hardware size: 6
Protocol size: 4
Opcode: Reply (2)
Sender Mac Address: 84:fd:d1:26:1c:09
Sender IP Address: 192.168.1.116
Target Mac Address: 4a:77:66:8b:8b:a3
Target IP Address: 192.168.1.106
0000 4A 77 66 8B 8B A3 84 FD D1 26 1C 09 08 06 00 01 Jwf.....&.....
0010 08 00 06 04 00 02 84 FD D1 26 1C 09 C0 A8 01 74 .....&.....t
0020 4A 77 66 8B 8B A3 C0 A8 01 6A Jwf.....j

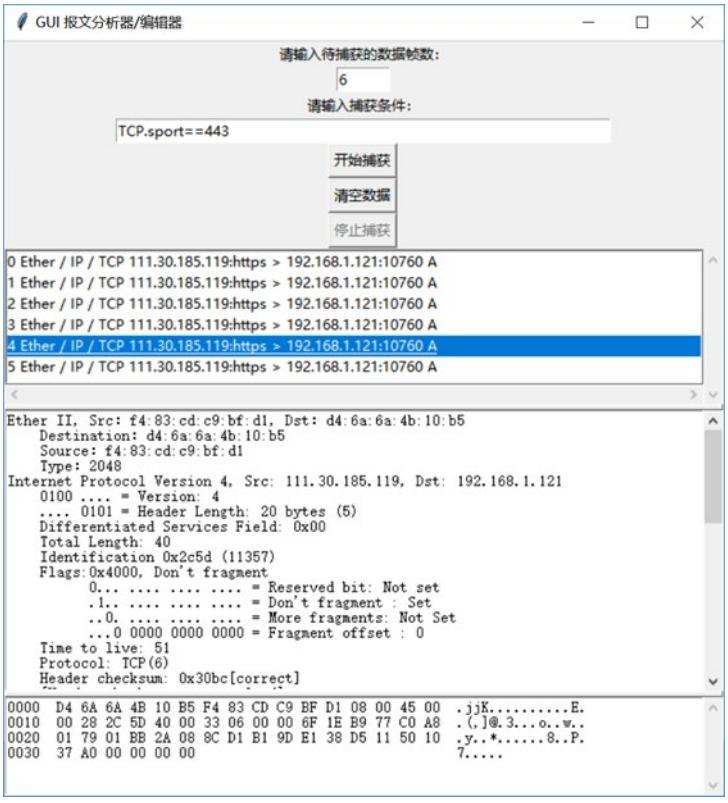

GUI 报文分析器
请输入待捕获的数据帧数：
0
请输入捕获条件：
ARP.src==192.168.1.106 && ARP.dst==192.168.1.116
开始捕获
清空数据
停止捕获
0 Ether / ARP who has 192.168.1.111 says 192.168.1.106
1 Ether / ARP who has 192.168.1.116 says 192.168.1.1
2 Ether / ARP is at 84:fd:d1:26:1c:09 says 192.168.1.116
3 Ether / ARP who has 192.168.1.123 says 192.168.1.106
4 Ether / ARP who has 192.168.1.116 says 192.168.1.106
5 Ether / ARP is at 84:fd:d1:26:1c:09 says 192.168.1.116
6 Ether / ARP who has 192.168.1.1 says 192.168.1.106
7 Ether / ARP who has 192.168.1.111 says 192.168.1.106
8 Ether / ARP who has 192.168.1.123 says 192.168.1.106
9 Ether / ARP who has 192.168.1.111 says 192.168.1.106
捕获时间: Mon Mar 21 14:26:10 2022
Ethernet II, Src: 4a:77:66:8b:8b:a3, Dst: ff:ff:ff:ff:ff:ff
Destination: ff:ff:ff:ff:ff:ff
Source: 4a:77:66:8b:8b:a3
Type: 2054
Address Resolution Protocol (Request)
Hardware type: Ethernet(1)
Protocol type: 2048
Hardware size: 6
Protocol size: 4
Opcode: Request (1)
Sender Mac Address: 4a:77:66:8b:8b:a3
Sender IP Address: 192.168.1.106
Target Mac Address: 00:00:00:00:00:00
Target IP Address: 192.168.1.123
0000 FF FF FF FF FF FF 4A 77 66 8B 8B A3 08 06 00 01 ......Jwf.....
0010 08 00 06 04 00 01 4A 77 66 8B 8B A3 C0 A8 01 6A ......Jwf.....j
0020 00 00 00 00 00 00 C0 A8 01 7B .....{

ARP 数据报（发送）

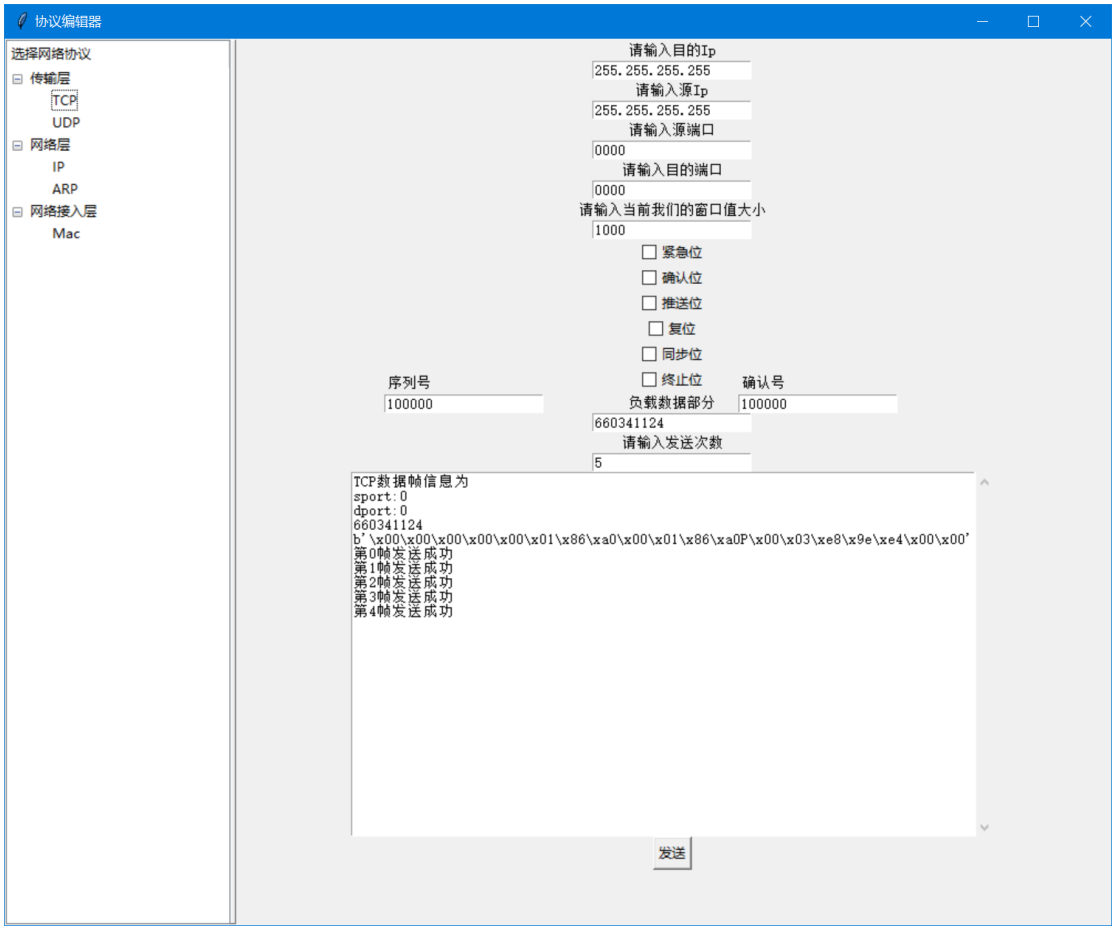

协议编辑器
选择网络协议
传输层
TCP
UDP
网络层
IP
ARP
网络接入层
Mac
请输入目的Mac
FF:FF:FF:FF:FF:FF
请输入目的Ip
255.255.255.255
请输入负载数据
660341124张三丰
请输入发送次数
5
运行arp的网络类型
以太网
运行arp的协议类型
IPV4
ARP操作类型
请求
ARP数据帧信息为
ipdst:255.255.255.255
ipsrc:127.0.0.1
macdst:FF:FF:FF:FF:FF:FF
macsrc:00:00:00:00:00:00
payload:660341124张三丰
b'\xff\xff\xff\xff\xff\xff\x00\x00\x00\x00\x00\x00\x08\x06\x00\x01\x08\x00\x06\x04\x00\x01\x00\x00\x00\x00\x00\x00\x7f\x00\x00\x01\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff660341124\xe5\xbc\xa0\xe4\xb8\x89\xe4\xb8\xb0'
第0帧发送成功
第1帧发送成功
第2帧发送成功
第3帧发送成功
第4帧发送成功
发送


协议编辑器
选择网络协议
传输层
TCP
UDP
网络层
IP
ARP
网络接入层
Mac
请输入目的Mac
FF:FF:FF:FF:FF:FF
请输入目的Ip
255.255.255.255
请输入负载数据
660341124张三丰
请输入发送次数
5
运行arp的网络类型
以太网
运行arp的协议类型
IPV4
ARP操作类型
应答
ARP数据帧信息为
ipdst:255.255.255.255
ipsrc:127.0.0.1
macdst:FF:FF:FF:FF:FF:FF
macsrc:00:00:00:00:00:00
payload:660341124张三丰
第0帧发送成功
第1帧发送成功
第2帧发送成功
第3帧发送成功
第4帧发送成功
发送

（4）TCP 数据报（接收）

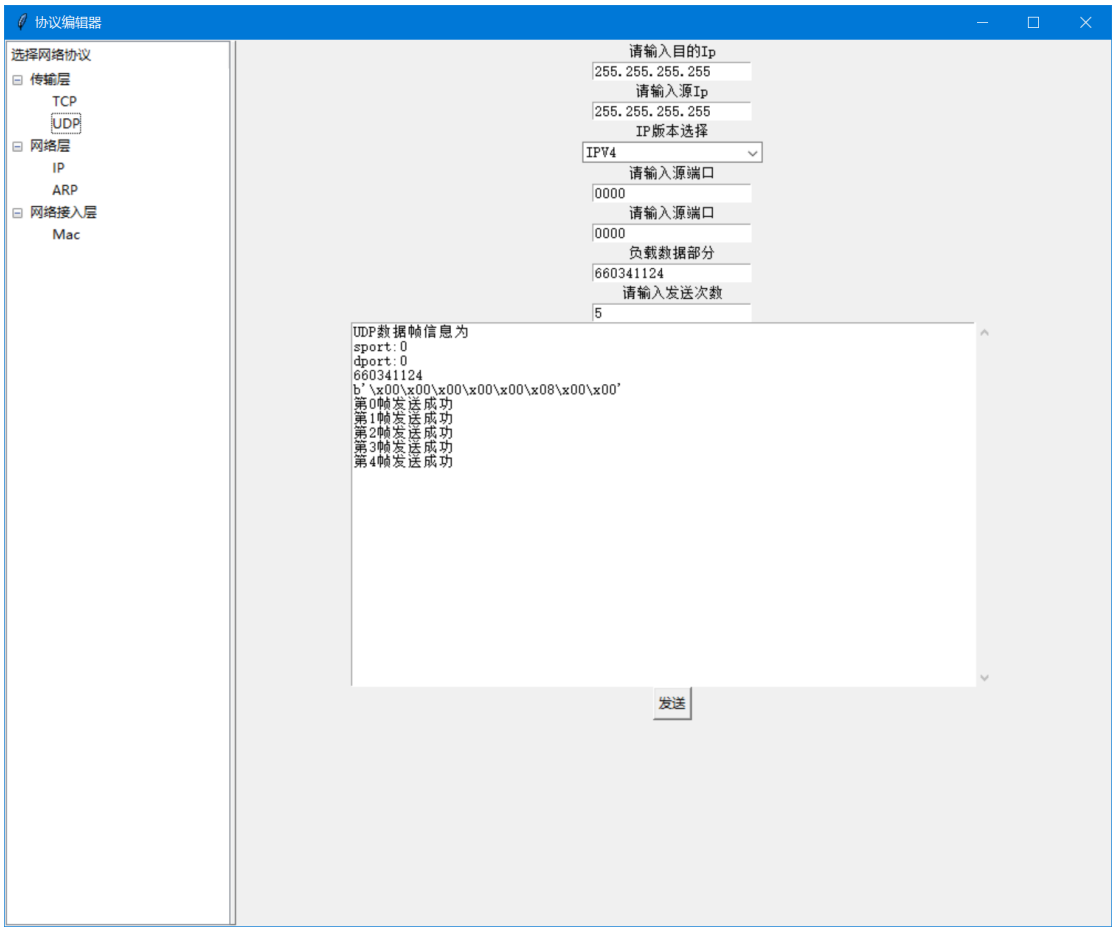

GUI 报文分析器/编辑器
请输入待捕获的数据帧数:
6
请输入捕获条件:
TCP.sport==443
开始捕获
清空数据
停止捕获
0 Ether / IP / TCP 111.30.185.119:https > 192.168.1.121:10760 A
1 Ether / IP / TCP 111.30.185.119:https > 192.168.1.121:10760 A
2 Ether / IP / TCP 111.30.185.119:https > 192.168.1.121:10760 A
3 Ether / IP / TCP 111.30.185.119:https > 192.168.1.121:10760 A
4 Ether / IP / TCP 111.30.185.119:https > 192.168.1.121:10760 A
5 Ether / IP / TCP 111.30.185.119:https > 192.168.1.121:10760 A
Ether II, Src: f4:83:cd:c9:bf:d1, Dst: d4:6a:6a:4b:10:b5
Destination: d4:6a:6a:4b:10:b5
Source: f4:83:cd:c9:bf:d1
Type: 2048
Internet Protocol Version 4, Src: 111.30.185.119, Dst: 192.168.1.121
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00
Total Length: 40
Identification 0x2c5d (11357)
Flags:0x4000, Don't fragment
0... .... = Reserved bit: Not set
.1.. .... = Don't fragment : Set
..0. .... = More fragments: Not Set
...0 0000 0000 0000 = Fragment offset : 0
Time to live: 51
Protocol: TCP(6)
Header checksum: 0x30bc[correct]
0000 D4 6A 6A 4B 10 B5 F4 83 CD C9 BF D1 08 00 45 00 .jjK..........E.
0010 00 28 2C 5D 40 00 33 06 00 00 6F 1E B9 77 C0 A8 .(,]@.3...o..w..
0020 01 79 01 BB 2A 08 8C D1 B1 9D E1 38 D5 11 50 10 .y..*.....8..P.
0030 37 A0 00 00 00 00 7.....

TCP 数据报（发送）



（5）UDP 数据报（接收）

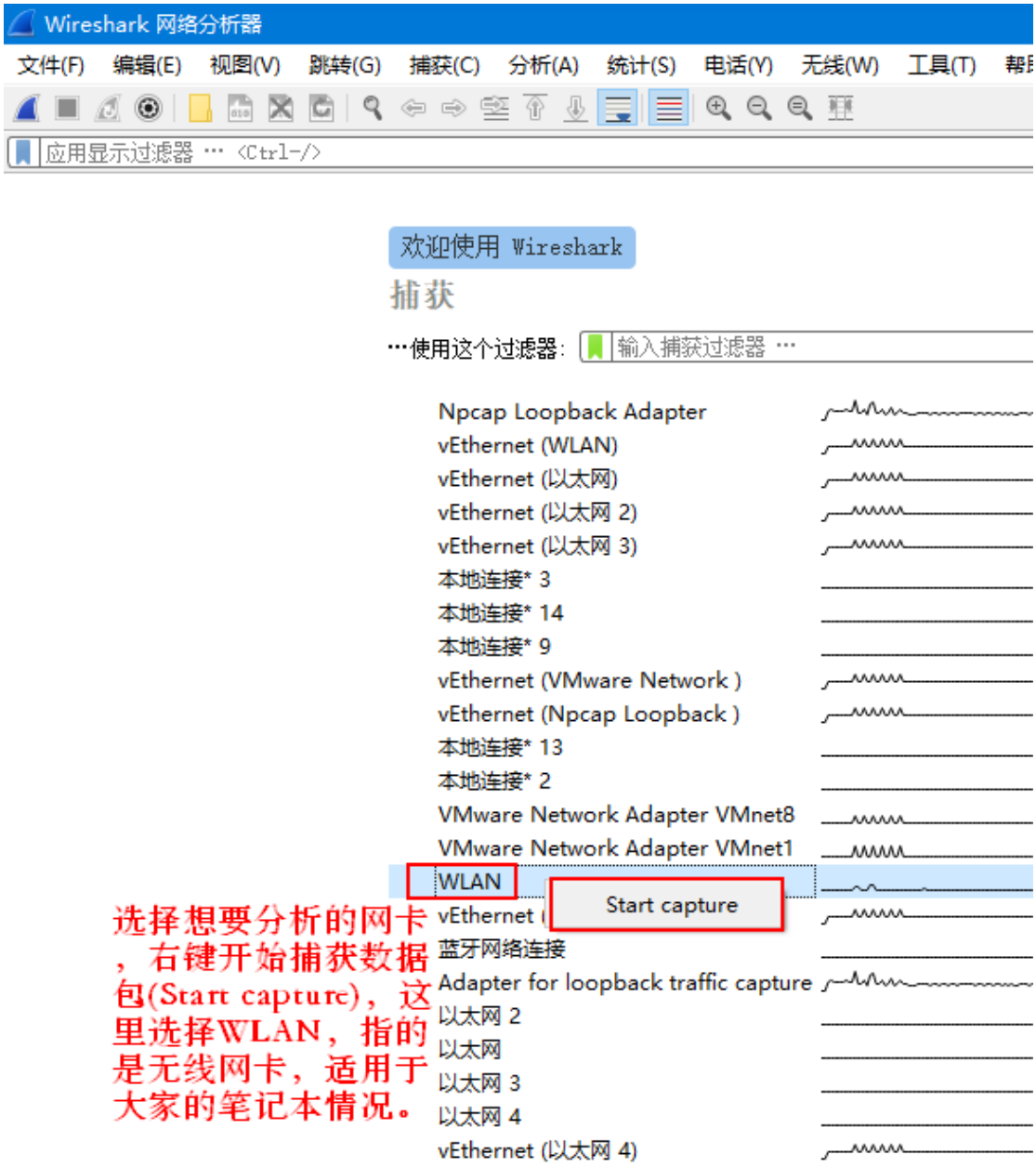
UDP 数据报（发送）



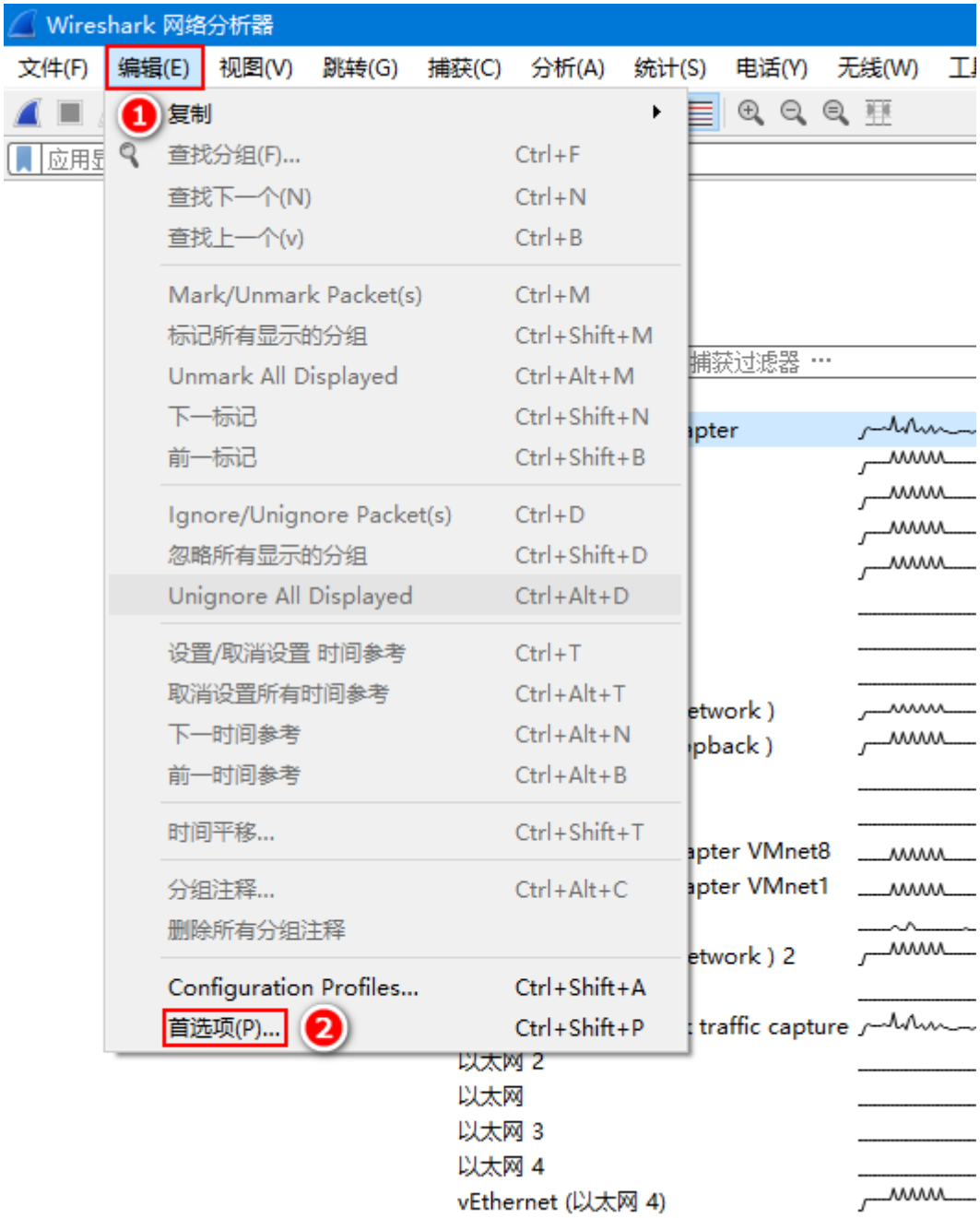
1.2.4 与 Wireshark 对比结果正确（注意五个协议接收/发送的数据包

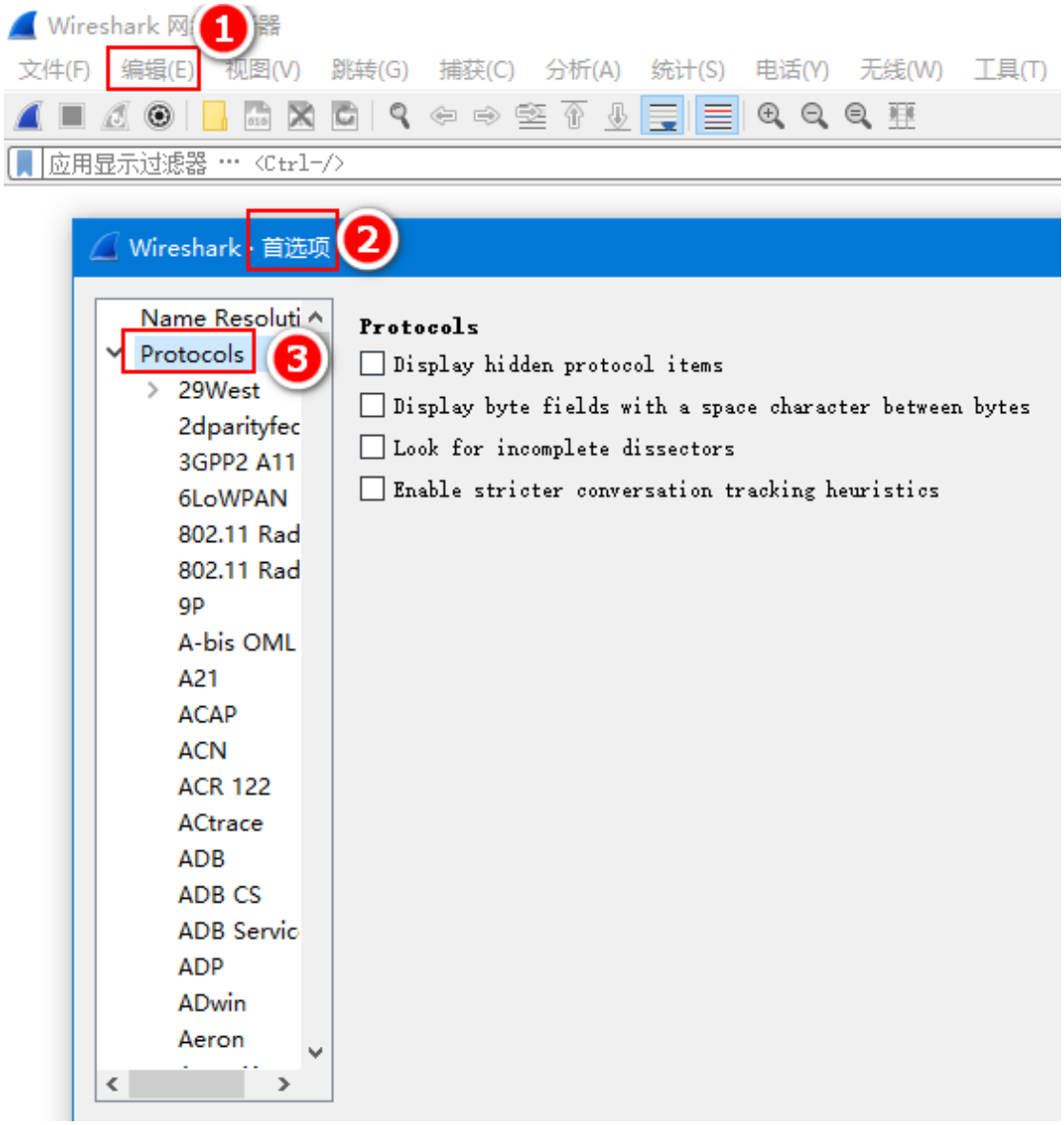
在验收时以及在课程设计报告中都需要截图和 Wireshark 对比验证，否则将影响分数）

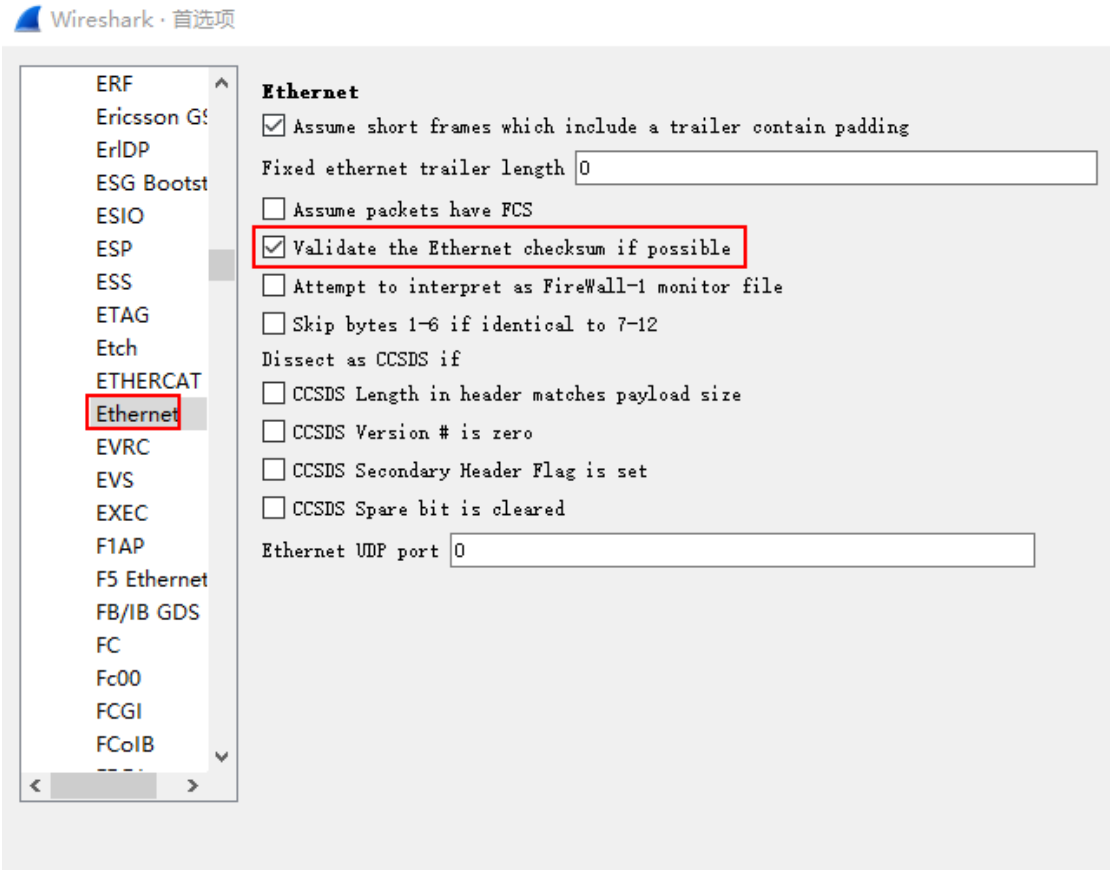
所有类型的报文的分析与检验和的计算验证都与 Wireshark 相同，以 TCP 为例结果对比，从 id (Identification)看是否属于同一个报文，其余字段分析正确，检验和计算正确。值得注意的是有时候 TCP 和 UDP 校验和会由网卡计算(https://blog.csdn.net/weixin_34308389/article/details/93114074)，因此 Wireshark 抓到的本机发送的 TCP/UDP 数据包的校验和都是错误的，这样检验校验和根本没有意义。所以 Wireshark 不自动做 TCP 和 UDP 校验和的校验。如果要校验校验和：可以在 edit(编辑)->preference(首选项)->protocols 中选择相应的 TCP 或者 UDP 协议，在相应的地方打钩。

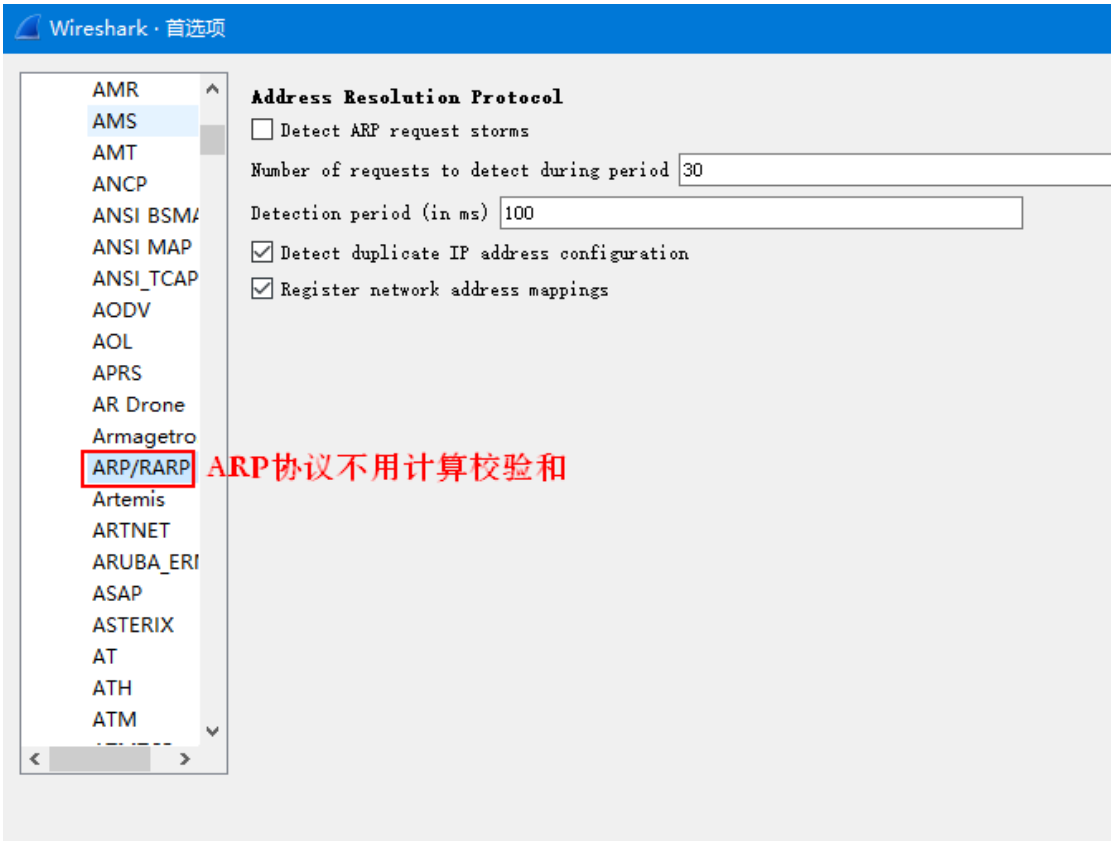


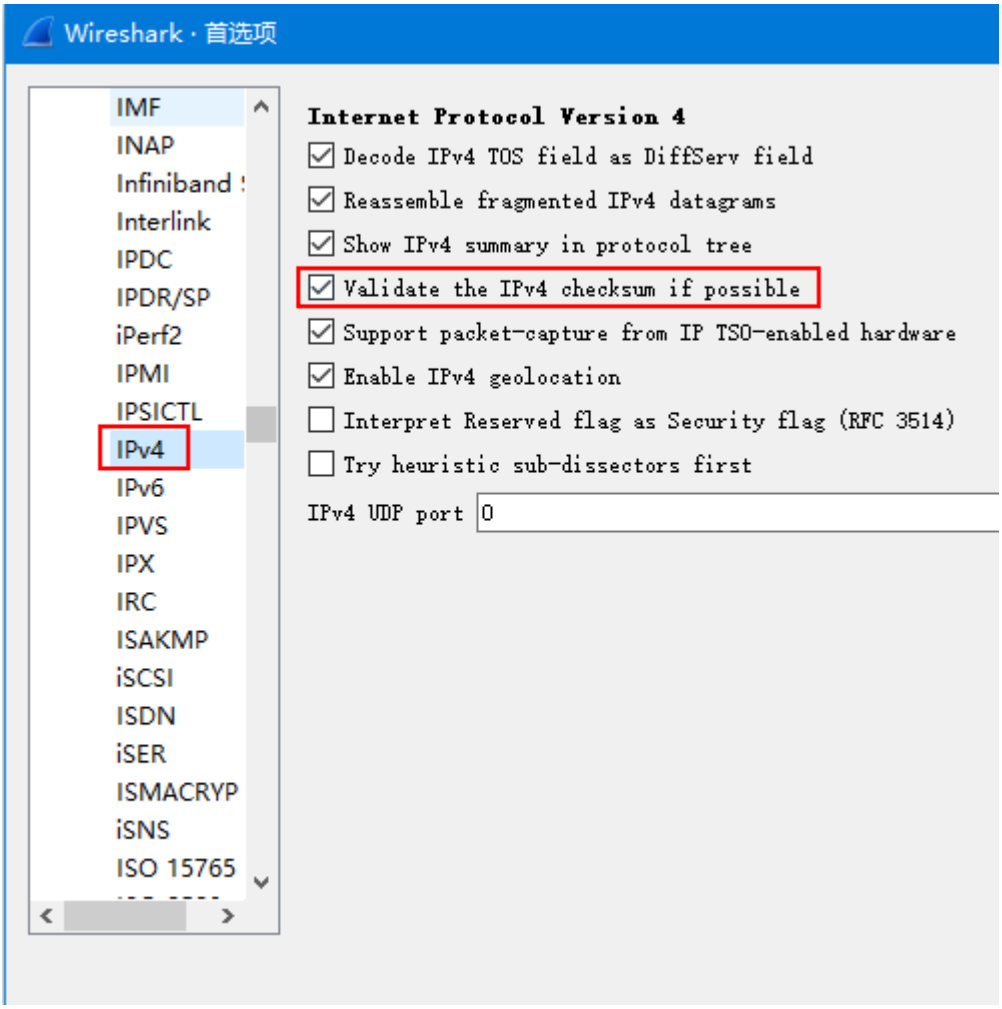
选择想要分析的网卡，右键开始捕获数据包(Start capture)，这里选择WLAN，指的是无线网卡，适用于大家的笔记本情况。

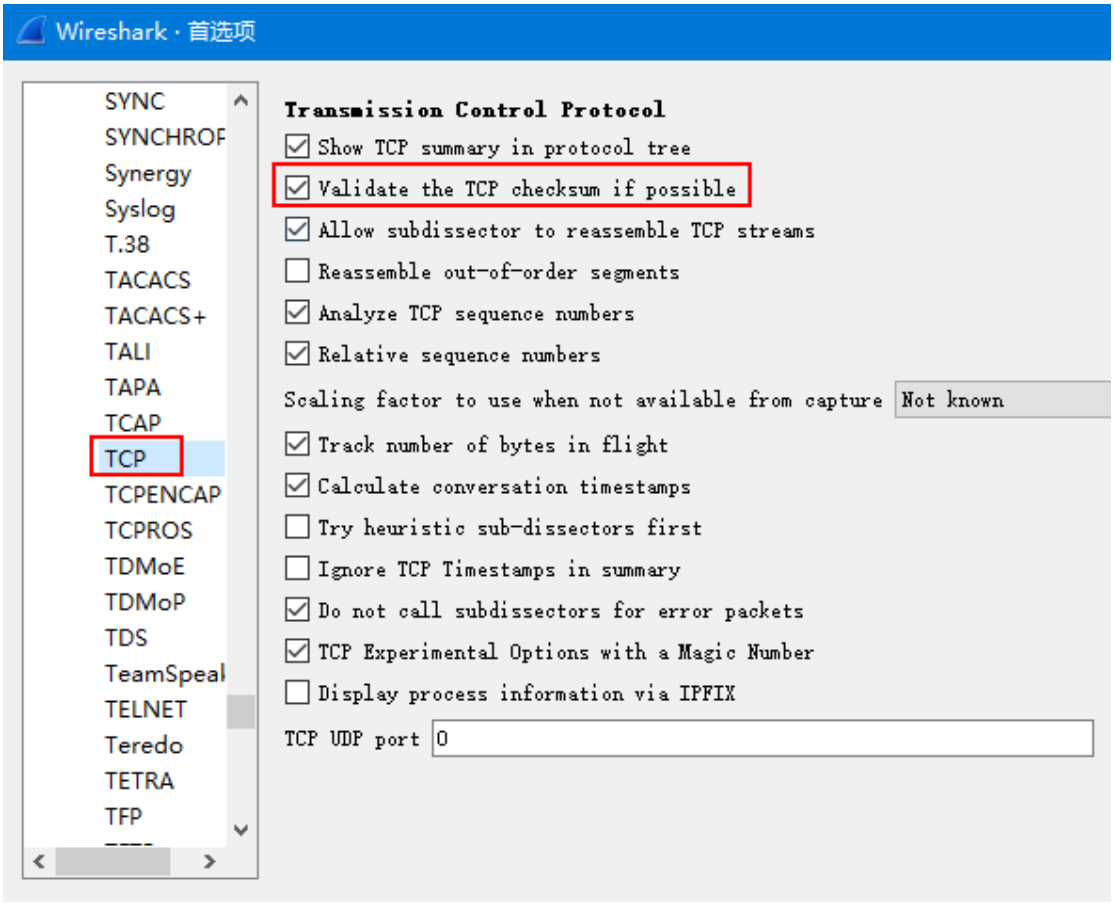

Wireshark 网络分析器
文件(F)
编辑(E)
视图(V)
跳转(G)
捕获(C)
分析(A)
统计(S)
电话(Y)
无线(W)
1
复制
查找分组(F)... Ctrl+F
查找下一个(N) Ctrl+N
查找上一个(v) Ctrl+B
Mark/Unmark Packet(s) Ctrl+M
标记所有显示的分组 Ctrl+Shift+M
Unmark All Displayed Ctrl+Alt+M
下一标记 Ctrl+Shift+N
前一标记 Ctrl+Shift+B
Ignore/Unignore Packet(s) Ctrl+D
忽略所有显示的分组 Ctrl+Shift+D
Unignore All Displayed Ctrl+Alt+D
设置/取消设置 时间参考 Ctrl+T
取消设置所有时间参考 Ctrl+Alt+T
下一时间参考 Ctrl+Alt+N
前一时间参考 Ctrl+Alt+B
时间平移... Ctrl+Shift+T
分组注释... Ctrl+Alt+C
删除所有分组注释
Configuration Profiles... Ctrl+Shift+A
首选项(P)... Ctrl+Shift+P
2
捕获过滤器 ···
以太网 2
以太网
以太网 3
以太网 4
vEthernet (以太网 4)

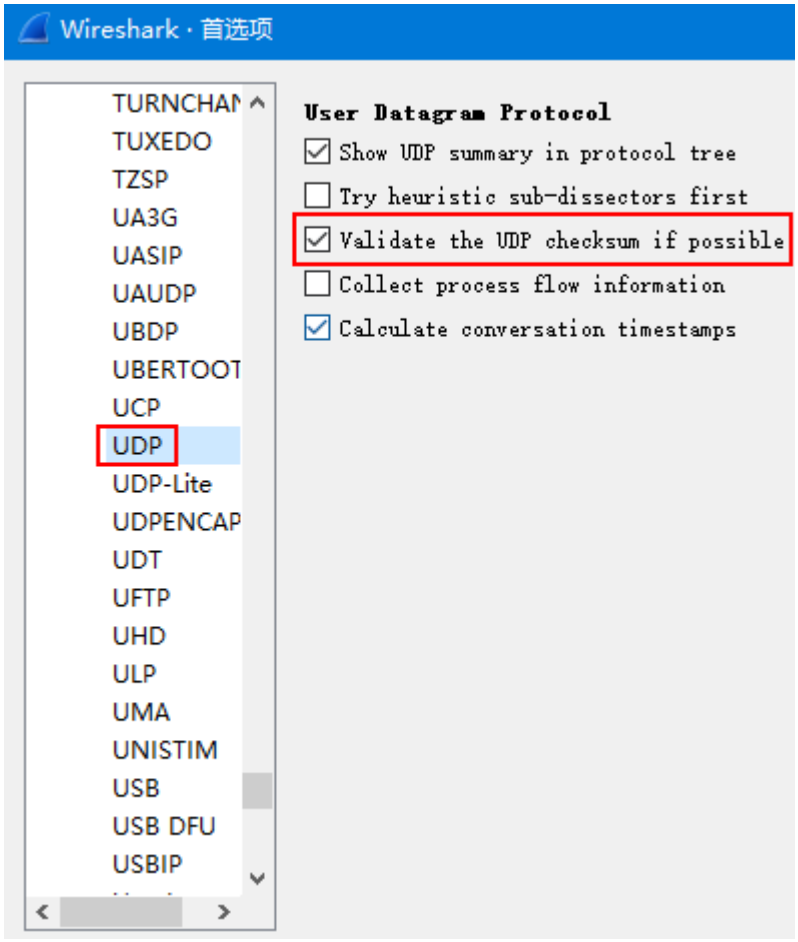

Wireshark 网
1
文件(F)
编辑(E)
视图(V)
跳转(G)
捕获(C)
分析(A)
统计(S)
电话(Y)
无线(W)
工具(T)
应用显示过滤器 … <Ctrl-/>
Wireshark · 首选项
2
Name Resoluti
Protocols
3
29West
2dparityfec
3GPP2 A11
6LoWPAN
802.11 Rad
802.11 Rad
9P
A-bis OML
A21
ACAP
ACN
ACR 122
ACtrace
ADB
ADB CS
ADB Servic
ADP
ADwin
Aeron
Protocols
Display hidden protocol items
Display byte fields with a space character between bytes
Look for incomplete dissectors
Enable stricter conversation tracking heuristics

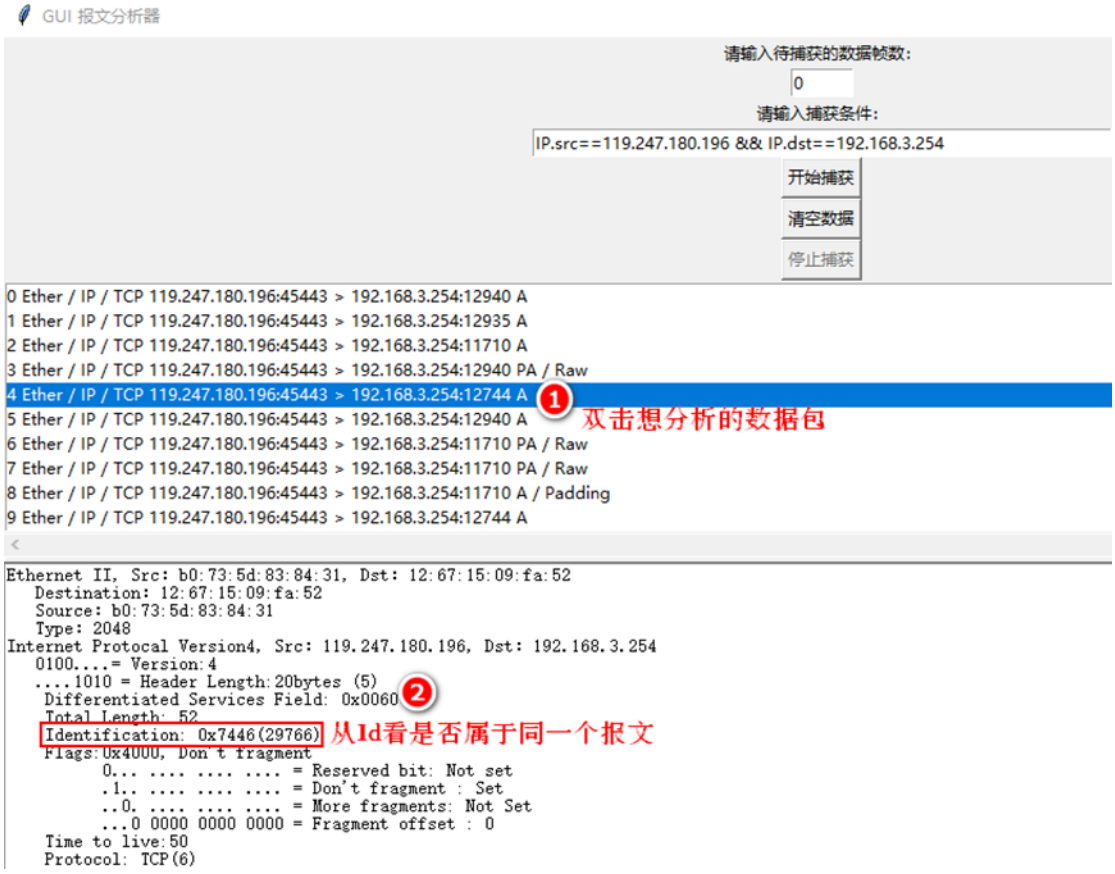

Wireshark · 首选项
ERF
Ericsson G
ErlDP
ESG Bootst
ESIO
ESP
ESS
ETAG
Etch
ETHERCAT
Ethernet
EVRC
EVS
EXEC
F1AP
F5 Ethernet
FB/IB GDS
FC
Fc00
FCGI
FCoIB
Ethernet
Assume short frames which include a trailer contain padding
Fixed ethernet trailer length 0
Assume packets have FCS
Validate the Ethernet checksum if possible
Attempt to interpret as FireWall-1 monitor file
Skip bytes 1-6 if identical to 7-12
Dissect as CCSDS if
CCSDS Length in header matches payload size
CCSDS Version # is zero
CCSDS Secondary Header Flag is set
CCSDS Spare bit is cleared
Ethernet UDP port 0

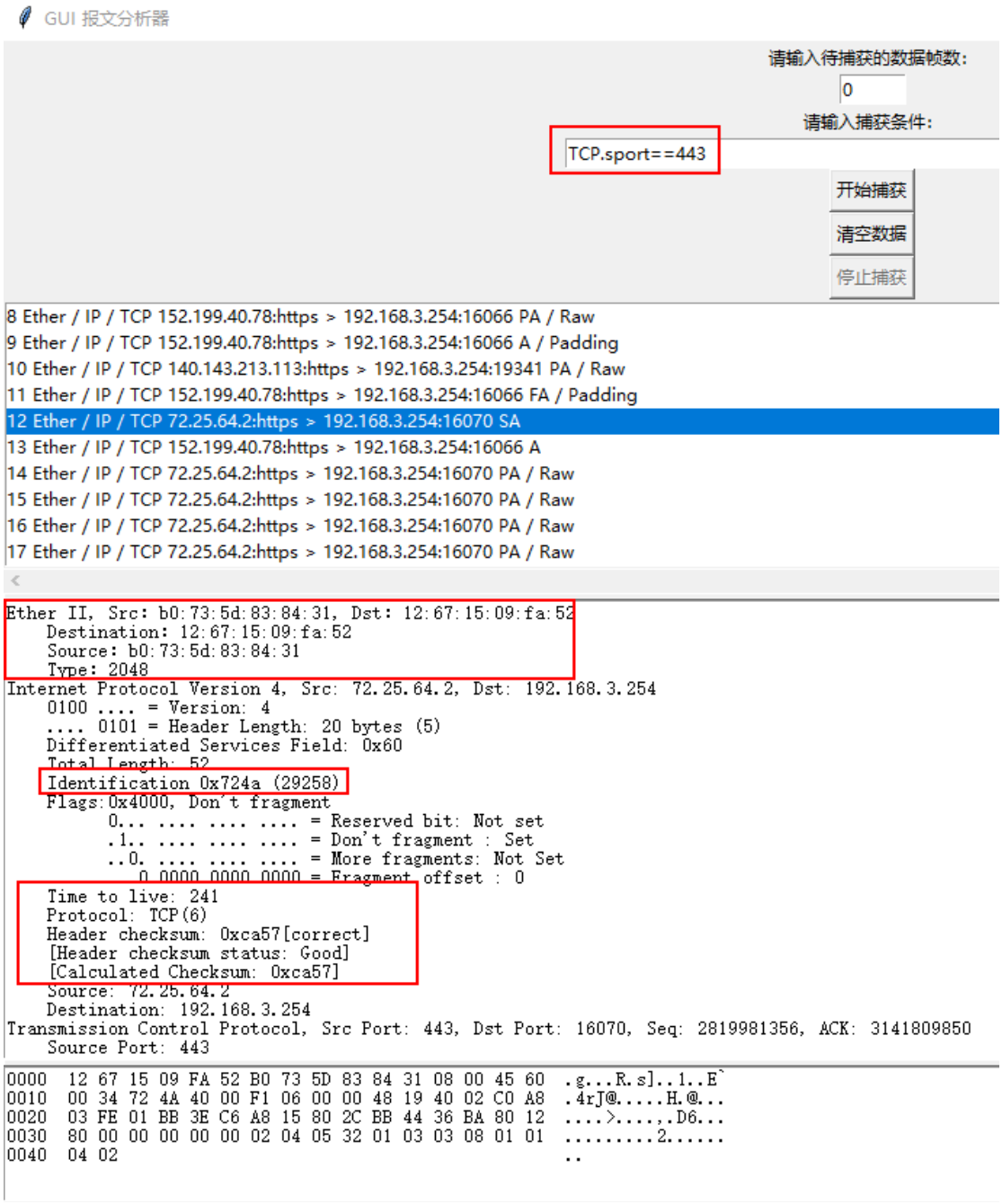

Wireshark · 首选项
AMR
AMS
AMT
ANCP
ANSI BSM
ANSI MAP
ANSI_TCAP
AODV
AOL
APRS
AR Drone
Armagetro
ARP/RARP
Artemis
ARTNET
ARUBA_ERI
ASAP
ASTERIX
AT
ATH
ATM
Address Resolution Protocol
Detect ARP request storms
Number of requests to detect during period 30
Detection period (in ms) 100
Detect duplicate IP address configuration
Register network address mappings
ARP协议不用计算校验和

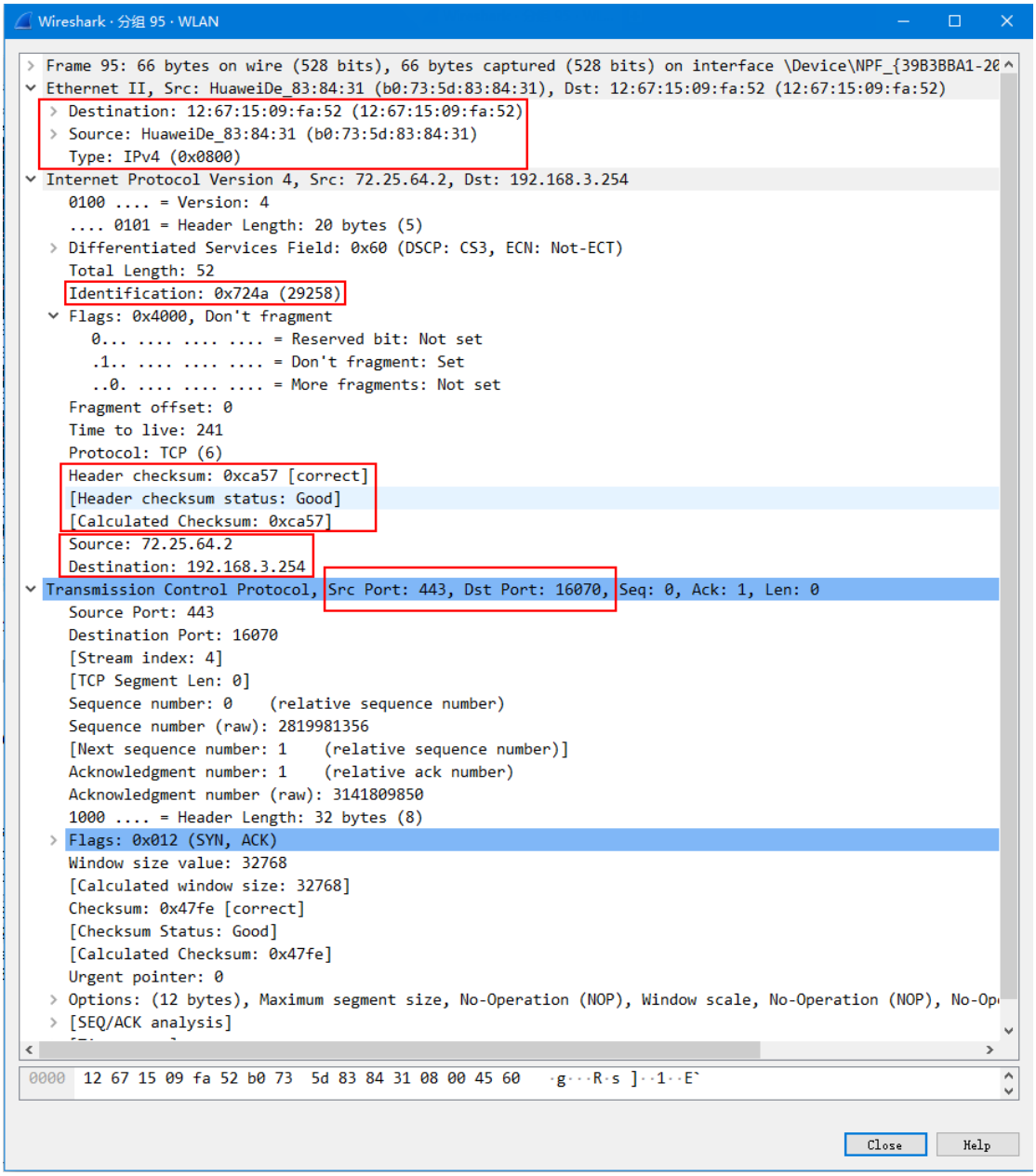

Wireshark · 首选项
IMF
INAP
Infiniband
Interlink
IPDC
IPDR/SP
iPerf2
IPMI
IPSICTL
IPv4
IPv6
IPVS
IPX
IRC
ISAKMP
iSCSI
ISDN
iSER
ISMACRYP
iSNS
ISO 15765
Internet Protocol Version 4
Decode IPv4 TOS field as DiffServ field
Reassemble fragmented IPv4 datagrams
Show IPv4 summary in protocol tree
Validate the IPv4 checksum if possible
Support packet-capture from IP TSO-enabled hardware
Enable IPv4 geolocation
Interpret Reserved flag as Security flag (RFC 3514)
Try heuristic sub-dissectors first
IPv4 UDP port
0


Wireshark · 首选项
SYNC
SYNCHROF
Synergy
Syslog
T.38
TACACS
TACACS+
TALI
TAPA
TCAP
TCP
TCPENCAP
TCPROS
TDMoE
TDMoP
TDS
TeamSpeal
TELNET
Teredo
TETRA
TFP
Transmission Control Protocol
Show TCP summary in protocol tree
Validate the TCP checksum if possible
Allow subdissector to reassemble TCP streams
Reassemble out-of-order segments
Analyze TCP sequence numbers
Relative sequence numbers
Scaling factor to use when not available from capture
Not known
Track number of bytes in flight
Calculate conversation timestamps
Try heuristic sub-dissectors first
Ignore TCP Timestamps in summary
Do not call subdissectors for error packets
TCP Experimental Options with a Magic Number
Display process information via IPFIX
TCP UDP port
0


Wireshark · 首选项
TURNCHAN
TUXEDO
TZSP
UA3G
UASIP
UAUDP
UBDP
UBERTOOT
UCP
UDP
UDP-Lite
UDPENCAP
UDT
UFTP
UHD
ULP
UMA
UNISTIM
USB
USB DFU
USBIP
User Datagram Protocol
Show UDP summary in protocol tree
Try heuristic sub-dissectors first
Validate the UDP checksum if possible
Collect process flow information
Calculate conversation timestamps


GUI 报文分析器
请输入待捕获的数据帧数:
0
请输入捕获条件:
IP.src==119.247.180.196 && IP.dst==192.168.3.254
开始捕获
清空数据
停止捕获
0 Ether / IP / TCP 119.247.180.196:45443 > 192.168.3.254:12940 A
1 Ether / IP / TCP 119.247.180.196:45443 > 192.168.3.254:12935 A
2 Ether / IP / TCP 119.247.180.196:45443 > 192.168.3.254:11710 A
3 Ether / IP / TCP 119.247.180.196:45443 > 192.168.3.254:12940 PA / Raw
4 Ether / IP / TCP 119.247.180.196:45443 > 192.168.3.254:12744 A
1
双击想分析的数据包
5 Ether / IP / TCP 119.247.180.196:45443 > 192.168.3.254:12940 A
6 Ether / IP / TCP 119.247.180.196:45443 > 192.168.3.254:11710 PA / Raw
7 Ether / IP / TCP 119.247.180.196:45443 > 192.168.3.254:11710 PA / Raw
8 Ether / IP / TCP 119.247.180.196:45443 > 192.168.3.254:11710 A / Padding
9 Ether / IP / TCP 119.247.180.196:45443 > 192.168.3.254:12744 A
Ethernet II, Src: b0:73:5d:83:84:31, Dst: 12:67:15:09:fa:52
Destination: 12:67:15:09:fa:52
Source: b0:73:5d:83:84:31
Type: 2048
Internet Protocal Version4, Src: 119.247.180.196, Dst: 192.168.3.254
0100.... = Version: 4
....1010 = Header Length: 20bytes (5)
2
Differentiated Services Field: 0x0060
Total Length: 52
Identification: 0x7446(29766)
从Id看是否属于同一个报文
Flags:0x4000, Don't fragment
0... .... = Reserved bit: Not set
.1.. .... = Don't fragment : Set
..0. .... = More fragments: Not Set
...0 0000 0000 0000 = Fragment offset : 0
Time to live: 50
Protocol: TCP(6)


GUI 报文分析器
请输入待捕获的数据帧数:
0
请输入捕获条件:
TCP.sport==443
开始捕获
清空数据
停止捕获
8 Ether / IP / TCP 152.199.40.78:https > 192.168.3.254:16066 PA / Raw
9 Ether / IP / TCP 152.199.40.78:https > 192.168.3.254:16066 A / Padding
10 Ether / IP / TCP 140.143.213.113:https > 192.168.3.254:19341 PA / Raw
11 Ether / IP / TCP 152.199.40.78:https > 192.168.3.254:16066 FA / Padding
12 Ether / IP / TCP 72.25.64.2:https > 192.168.3.254:16070 SA
13 Ether / IP / TCP 152.199.40.78:https > 192.168.3.254:16066 A
14 Ether / IP / TCP 72.25.64.2:https > 192.168.3.254:16070 PA / Raw
15 Ether / IP / TCP 72.25.64.2:https > 192.168.3.254:16070 PA / Raw
16 Ether / IP / TCP 72.25.64.2:https > 192.168.3.254:16070 PA / Raw
17 Ether / IP / TCP 72.25.64.2:https > 192.168.3.254:16070 PA / Raw
Ether II, Src: b0:73:5d:83:84:31, Dst: 12:67:15:09:fa:52
Destination: 12:67:15:09:fa:52
Source: b0:73:5d:83:84:31
Type: 2048
Internet Protocol Version 4, Src: 72.25.64.2, Dst: 192.168.3.254
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x60
Total Length: 52
Identification 0x724a (29258)
Flags:0x4000, Don't fragment
0... .... = Reserved bit: Not set
.1.. .... = Don't fragment : Set
..0. .... = More fragments: Not Set
...0 0000 0000 0000 = Fragment offset : 0
Time to live: 241
Protocol: TCP(6)
Header checksum: 0xca57[correct]
[Header checksum status: Good]
[Calculated Checksum: 0xca57]
Source: 72.25.64.2
Destination: 192.168.3.254
Transmission Control Protocol, Src Port: 443, Dst Port: 16070, Seq: 2819981356, ACK: 3141809850
Source Port: 443
0000 12 67 15 09 FA 52 B0 73 5D 83 84 31 08 00 45 60 .g...R.s]..1..E`
0010 00 34 72 4A 40 00 F1 06 00 00 48 19 40 02 C0 A8 .4rJ@.....H.@...
0020 03 FE 01 BB 3E C6 A8 15 80 2C BB 44 36 BA 80 12 ....>....,.D6...
0030 80 00 00 00 00 00 02 04 05 32 01 03 03 08 01 01 ....2......
0040 04 02 ..


Wireshark · 分组 95 · WLAN
Frame 95: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{39B3BBA1-20
Ethernet II, Src: HuaweiDe_83:84:31 (b0:73:5d:83:84:31), Dst: 12:67:15:09:fa:52 (12:67:15:09:fa:52)
Destination: 12:67:15:09:fa:52 (12:67:15:09:fa:52)
Source: HuaweiDe_83:84:31 (b0:73:5d:83:84:31)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 72.25.64.2, Dst: 192.168.3.254
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x60 (DSCP: CS3, ECN: Not-ECT)
Total Length: 52
Identification: 0x724a (29258)
Flags: 0x4000, Don't fragment
0... .... = Reserved bit: Not set
.1.. .... = Don't fragment: Set
..0. .... = More fragments: Not set
Fragment offset: 0
Time to live: 241
Protocol: TCP (6)
Header checksum: 0xca57 [correct]
[Header checksum status: Good]
[Calculated Checksum: 0xca57]
Source: 72.25.64.2
Destination: 192.168.3.254
Transmission Control Protocol, Src Port: 443, Dst Port: 16070, Seq: 0, Ack: 1, Len: 0
Source Port: 443
Destination Port: 16070
[Stream index: 4]
[TCP Segment Len: 0]
Sequence number: 0 (relative sequence number)
Sequence number (raw): 2819981356
[Next sequence number: 1 (relative sequence number)]
Acknowledgment number: 1 (relative ack number)
Acknowledgment number (raw): 3141809850
1000 .... = Header Length: 32 bytes (8)
Flags: 0x012 (SYN, ACK)
Window size value: 32768
[Calculated window size: 32768]
Checksum: 0x47fe [correct]
[Checksum Status: Good]
[Calculated Checksum: 0x47fe]
Urgent pointer: 0
Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Op
[SEQ/ACK analysis]
0000 12 67 15 09 fa 52 b0 73 5d 83 84 31 08 00 45 60
Close
Help