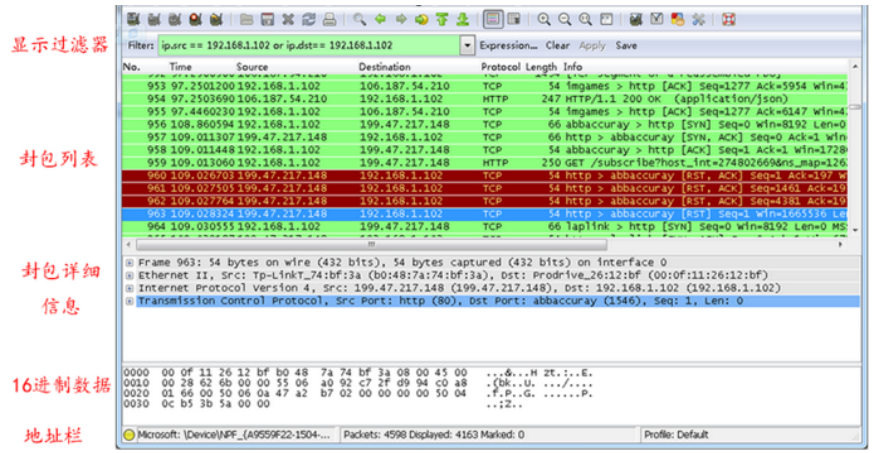


Wireshark主要分为这几个界面

1 Display Filter(显示过滤器), 用于过滤

2 Packet List Pane(封包列表)，显示捕获到的封包，有源地址和目标地址、端口号。

3 Packet Details Pane(封包详细信息)，显示封包中的字段

4 Dissector Pane(16进制数据)

5 Miscellanous(地址栏等其他)

wireshark两种过滤器

过滤器有：捕捉过滤器和显示过滤器

常用的过滤表达式

1协议过滤：

HTTP、TCP、UDP、rtsp、rtp等等

wireshark工具使用详解以及tcp三次握手抓包解析

概述

今天学习了下抓包工作的使用，写个文档记录下笔记总结。

Wireshark介绍

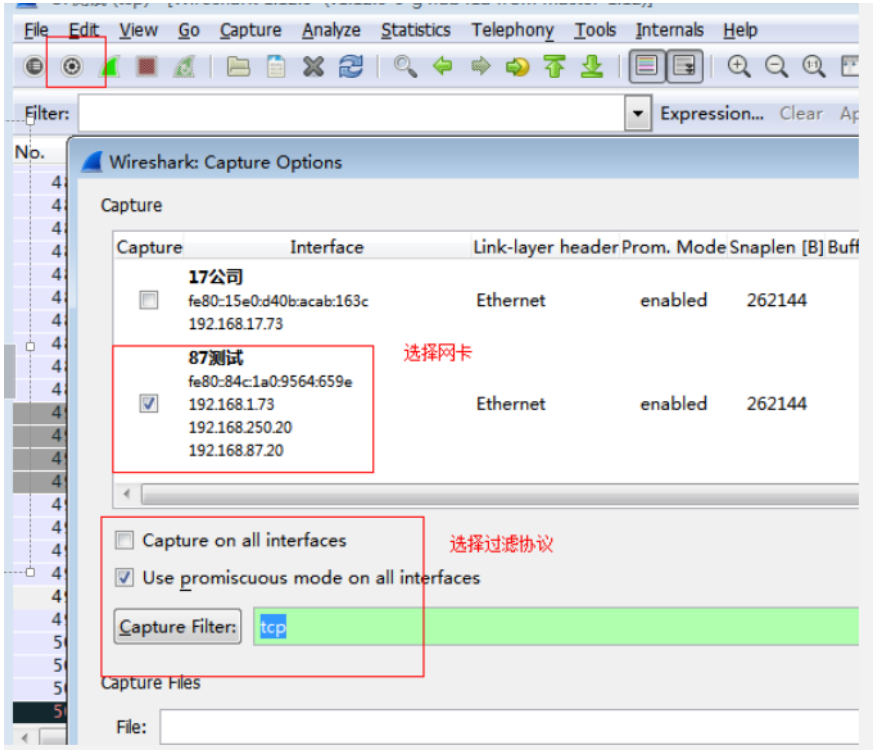
wireshark是非常流行的网络封包分析软件，可以截取各种网络封包，显示网络封包的详细信息。

wireshark用处：

wireshark是捕获机器上的某一块网卡的网络包，当你的机器上有多块网卡的时候，你需要选择一个网卡。

为了安全考虑，wireshark只能查看封包，而不能修改封包的内容，或者发送封包。

Wireshark窗口介绍



Protocol(协议):

语法：	Protocol	Direction	Host(s)	Value	Logical Operations	Other expression

可能值： ip、arp、rarp、tcp、udp、smtp、http、ftp、dns、icmp、...

Direction(方向):

可能值： stc、dst、src and dst、src or dst

如果没有指明方向，则默认使用“src or dst”作为关键字

“host 10.2.2.2”与“src or dst host 10.2.2.2”等价

Host(s):

ip.src == 192.168.1. 100显示源地址为192.168.1.100

ip.dst == 192.168.1.200 显示目标地址为192.168.1.200

3 端口过滤

tcp.port == 80 端口为80的

tcp.srcport== 80 只显示tcp协议的源端口为80的

udp.port eq 15000 udp端口为15000

tcp.port >= 1 and tcp.port <= 80 过滤端口范围

4 HTTP模式过滤

http.request.method == “Get” 只显示端口范围

5逻辑运算符为 AND / OR， && / ||

捕捉过滤器（CaptureFilters）

1.用于决定将什么样的信息记录在捕捉结果中

2.捕捉过滤器在抓抱前进行设置，决定抓取怎样的数据

3.捕捉过滤器仅支持协议过滤

捕捉过滤器： *捕捉前依据协议的相关信息进行过滤设置*

捕捉过滤器语法

语法：	Protocol	String 1	String 2	Comparison operator	Value	Logical Operations	Other expression

例如：http.request.method == "POST" string1和string2是可选的。

依据协议过滤时，可直接通过协议来进行过滤，也能依据协议的属性值进行过滤。

按协议进行过滤：

snmp || dns || icmp 显示SNMP或DNS或ICMP封包。

按协议的属性值进行过滤：

ip.addr == 10.1.1.1 ip.src != 10.1.2.3 or ip.dst != 10.4.5.6

ip.src == 10.230.0.0/16 显示来自10.230网段的封包。

tcp.port == 25 显示来源或目的TCP端口号为25的封包。

tcp.dstport == 25 显示目的TCP端口号为25的封包。

http.request.method== "POST" 显示post请求方式的http封包。

http.host == "tracker.1ting.com" 显示请求的域名为tracker.1ting.com的http封包。

tcp.flags.syn == 0X02 显示包含TCP SYN标志的封包。

内容过滤语法

深度字符串匹配

tcp contains "http"

显示payload中包含"http"字符串的tcp封包。

http.request.uri contains "online"

显示请求的uri包含"online"的http封包。

Logical Operations(逻辑运算)：

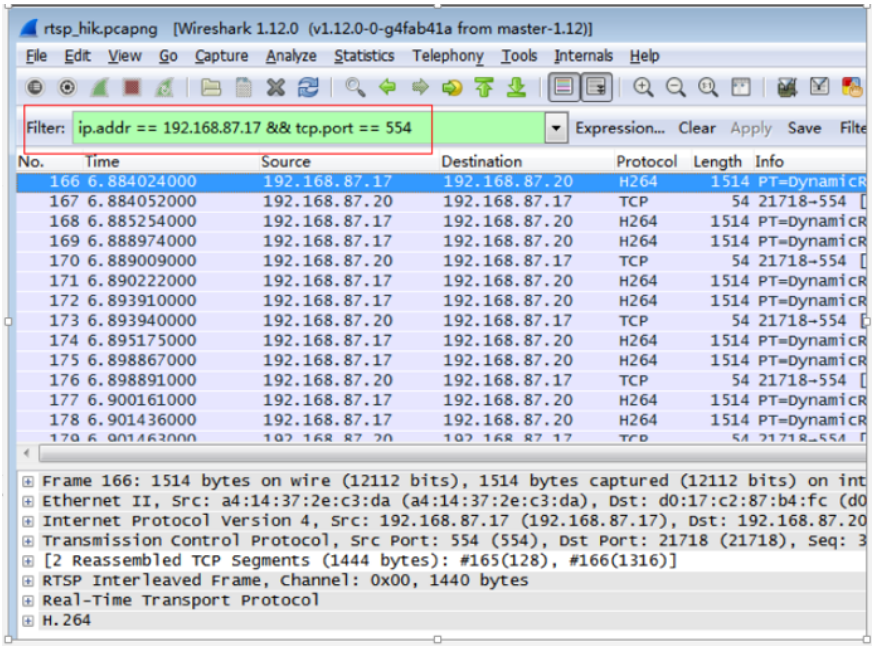
可能只： not、and、 or

显示过滤器（DisplayFilters）

1.用于在捕捉结果中进行详细查找

2.用于过滤抓包数据，方便stream的追踪和排查

3.显示过滤器既支持协议过滤也支持内容过滤



显示过滤器：*对捕捉到的数据包依据协议或包的内容进行过滤*

协议过滤语法

// 常用的研究两者间的通信
ip.addr== 192.168.8.54 || ip.addr== 112.80.248.74
ip.src == 10.43.54.65 or ip.dst == 10.43.54.65

tcp和udp过滤

tcp.port == 80
tcp.port eq 80 or udp.port eq 80
tcp.port eq 25 or icmp
tcp.port >= 1 and tcp.port <= 80
tcp.window_size == 0 && tcp.flags.reset != 1
udp.length == 26
tcp类型和内容：
tcp[13] & 0×00 = 0: No flags set (null scan)
tcp[13] & 0×01 = 1: FIN set and ACK not set
tcp[13] & 0×03 = 3: SYN set and FIN set
tcp[13] & 0×05 = 5: RST set and FIN set
tcp[13] & 0×06 = 6: SYN set and RST set
tcp[13] & 0×08 = 8: PSH set and ACK not set

http过滤示例

http.request.method == “GET”

http.request.method == “POST”

http.request.uri == “/img/logo-edu.gif”

http.host == party.syyx.com
http.response.code == 404
http.content_type contains "javascript"

http contains “GET”

http.request.method == "GET" && http contains "Host: "

tcp[20:3] == 47:45:54

/* 16进制形式，tcp头部一般是20字节，

所以这个是对payload的前三个字节进行过滤 */

http.host[0:4] == "trac"

包长度过滤

udp.length == 26

这个长度是指udp本身固定长度8加上udp下面那块数据包之和

tcp.len >= 7

指的是ip数据包(tcp下面那块数据),不包括tcp本身

ip.len == 94

除了以太网头固定长度14,其它都算是ip.len,即从ip本身到最后

frame.len == 119

整个数据包长度,从eth开始到最后

过滤实例

mac过滤 IP过滤

Mac 过滤

eth.dst == A0:00:00:04:C5:84
eth.src eq A0:00:00:04:C5:84
eth.dst==A0:00:00:04:C5:84
eth.dst==A0-00-00-04-C5-84
IP过滤