

link: null  
title: 珠峰架构师成长计划  
description: null  
keywords: null  
author: null  
date: null  
publisher: 珠峰架构师成长计划  
stats: paragraph=44 sentences=89, words=270

1. wireshark有什么用 #

- 1. 分析网络底层协议
- 2. 解决网络故障问题
- 3. 找寻网络安全问题

2. 安装 #

- [download \(https://www.wireshark.org/download.html\)](https://www.wireshark.org/download.html)
- [wiresharkbook \(http://wiresharkbook.com/\)](http://wiresharkbook.com/)

3. 快速抓包 #

- 初始化界面
- 选择网卡
- 停止抓包
- 保存数据包

4. 界面 #

- 标题栏
- 主菜单栏
- 工具栏
- 数据包过滤栏
- 数据包列表区
- 数据包详细区
- 数据包字节区
- 数据包统计区

5. 过滤器设置 #

5.1 抓包过滤器 #

5.1.1 语法 #

协议+方向+类型+值

- HOST net port host
- 方向 src、dst、src and dst、src or dst
- 协议 ether ip tcp udp http ftp
- 逻辑运算符 && || !

5.1.2 例子 #

- src host 192.168.1.1 && dst port 80 抓取来源地址为192.168.1.1，并且目的为80端口的流量
- host 192.168.1.1|| host host 192.168.1.2 抓取192.168.1.1或192.168.1.2的流量
- !broadcast 不抓取广播包

5.1.2.1 过滤MAC #

- ether host 00:00:00:00:00:00 网卡主机
- ether src host 00:00:00:00:00:00 来源MAC
- ether dst host 00:00:00:00:00:00 目标MAC

5.1.2.2 过滤IP #

- host 192.168.1.1
- src host 192.168.1.1
- det host 192.168.1.1

5.1.2.3 过滤端口 #

- port 80
- lport 80
- dst port 80
- src port 80

5.1.2.4 过滤协议 #

- arp
- tcp

5.1.2.5 综合过滤 #

- host 192.168.1.100 && port 8080

5.2 显示过滤器 #

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

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

5.2.1 语法 #

- 比较操作符 == != > < >=

5.2.2 过滤端口 #

- tcp.port == 80

- tcp.srcport == 80
- tcp.dstport == 80
- tcp.flags.sync == 1

#### 5.2.2.3 过滤协议 #

- arp
- tcp
- udp
- not http
- not arp

#### 5.2.2.4 案例 #

- ip.src == 192.168.0.1 and tcp.dstport == 80
- ip.addr == 192.168.0.1 and udp.port == 60000

## 6. 三次握手 #

No.	Time	来源地址	目的地址	Protocol	Length	来源端口	目标端口	Info
90	2018-03-29 16:21:54.623	192.168.1.103	23.105.205.147	TCP	66	65178	8888	65178 → 8888 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 S...
92	2018-03-29 16:21:54.781	23.105.205.147	192.168.1.103	TCP	66	8888	65178	8888 → 65178 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS...
93	2018-03-29 16:21:54.782	192.168.1.103	23.105.205.147	TCP	54	65178	8888	65178 → 8888 [ACK] Seq=1 Ack=1 Win=66792 Len=0
94	2018-03-29 16:21:54.782	192.168.1.103	23.105.205.147	HTTP	137	65178	8888	GET / HTTP/1.1
95	2018-03-29 16:21:54.946	23.105.205.147	192.168.1.103	TCP	54	8888	65178	8888 → 65178 [ACK] Seq=1 Ack=84 Win=29312 Len=0
96	2018-03-29 16:21:54.946	23.105.205.147	192.168.1.103	HTTP	155	8888	65178	HTTP/1.1 200 OK
97	2018-03-29 16:21:54.946	192.168.1.103	23.105.205.147	TCP	54	65178	8888	65178 → 8888 [FIN, ACK] Seq=84 Ack=102 Win=66688 Len=0
99	2018-03-29 16:21:55.105	23.105.205.147	192.168.1.103	TCP	54	8888	65178	8888 → 65178 [FIN, ACK] Seq=102 Ack=85 Win=29312 Len=0
100	2018-03-29 16:21:55.105	192.168.1.103	23.105.205.147	TCP	54	65178	8888	65178 → 8888 [ACK] Seq=85 Ack=103 Win=66688 Len=0

## 7. Wireshark与对应的OSI七层模型 #

## TCP包具体内容 #

## 8. 参考 #

- [wireshark \(https://www.cnblogs.com/TankXiao/archive/2012/10/10/2711777.html\)](https://www.cnblogs.com/TankXiao/archive/2012/10/10/2711777.html)