

# Wireshark 笔记

作者：享学课堂网安院长--James 老师

## 章节一：Wireshark 抓包介绍

### 1.1, Wireshark 简介

Wireshark 是一个网络封包分析软件。网络封包分析软件的功能是撷取网络封包，并尽可能显示出最为详细的网络封包资料。Wireshark 使用 WinPCAP 作为接口，直接与网卡进行数据报文交换。

### 1.2, Wireshark 的应用

网络管理员使用 Wireshark 来检测网络问题，  
网络安全工程师使用 Wireshark 来检查资讯安全相关问题，  
开发者使用 Wireshark 来为新的通讯协议除错，  
普通使用者使用 Wireshark 来学习网络协议的相关知识。  
当然，有的人也会“居心叵测”的用它来寻找一些敏感信息.....

### 1.3, Wireshark 抓数据包技巧-

(1)确定 Wireshark 的物理位置。如果没有一个正确的位置，启动 Wireshark 后会花费很长的时间捕获一些与自己无关的数据。

(2)选择捕获接口。一般都是选择连接到 Internet 网络的接口，这样才可以捕获到与网络相关的数据。否则，捕获到的其它数据对自己也没有任何帮助。

(3)使用捕获过滤器。通过设置捕获过滤器，可以避免产生过大的捕获数据。这样用户在分析数据时，也不会受其它数据干扰。而且，还可以为用户节约大量的时间。e

(4)使用显示过滤器。通常使用捕获过滤器过滤后的数据，往往还是很复杂。为了使过滤的数据包再更细致，此时使用显示过滤器进行过滤。火

(5)使用着色规则。通常使用显示过滤器过滤后的数据，都是有用的数据包。如果想更加突出的显示某个会话，可以使用着色规则高亮显示。

(6)构建图表。如果用户想要更明显的看出一个网络中数据的变化情况，使用图表的形式可以很方便的展现数据分布情况。

(7)重组数据。当传输较大的图片或文件时，需要将信息分布在多个数据包中。这时候就需要使用重组数据的方法来抓取完整的数据。Wireshark 的重组功能，可以重组一个会话中不同数据包的信息，或者是重组一个完整的图片或文件。

## 章节二：Wireshark 抓包入门操作

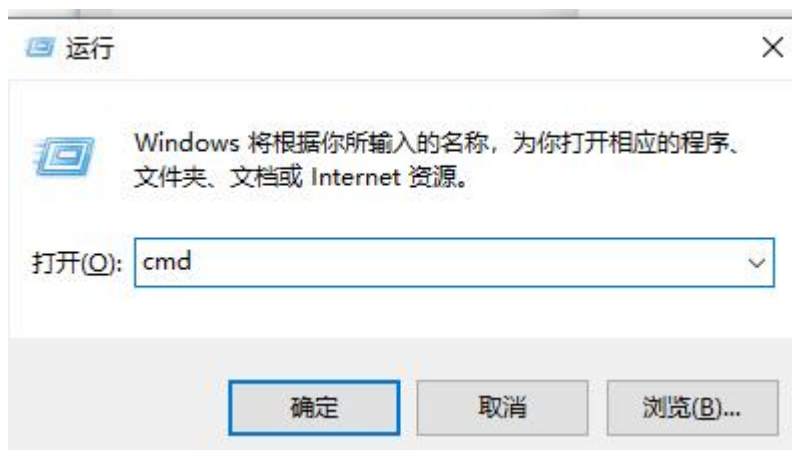
### 2.1 常见协议包

本节课主要分析以下几种协议类型。

ARP 协议

ICMP 协议  
TCP 协议  
UDP 协议  
DNS 协议  
HTTP 协议

## 2.2 查看本机要抓包的网路



输入指令 `ipconfig` 找到对应的网路

```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.19045.2364]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\james>ipconfig

Windows IP 配置

以太网适配器 以太网:

    连接特定的 DNS 后缀 . . . . . : 
    本地链接 IPv6 地址. . . . . : fe80::def5:5405:8843:5732%14
    IPv4 地址 . . . . . : 192.168.2.199
    子网掩码 . . . . . : 255.255.255.0
    默认网关. . . . . : 192.168.2.1

以太网适配器 VMware Network Adapter VMnet1:

    连接特定的 DNS 后缀 . . . . . : 
    本地链接 IPv6 地址. . . . . : fe80::e470:3e11:6058:832%19
    IPv4 地址 . . . . . : 192.168.127.1
    子网掩码 . . . . . : 255.255.255.0
    默认网关. . . . . : 

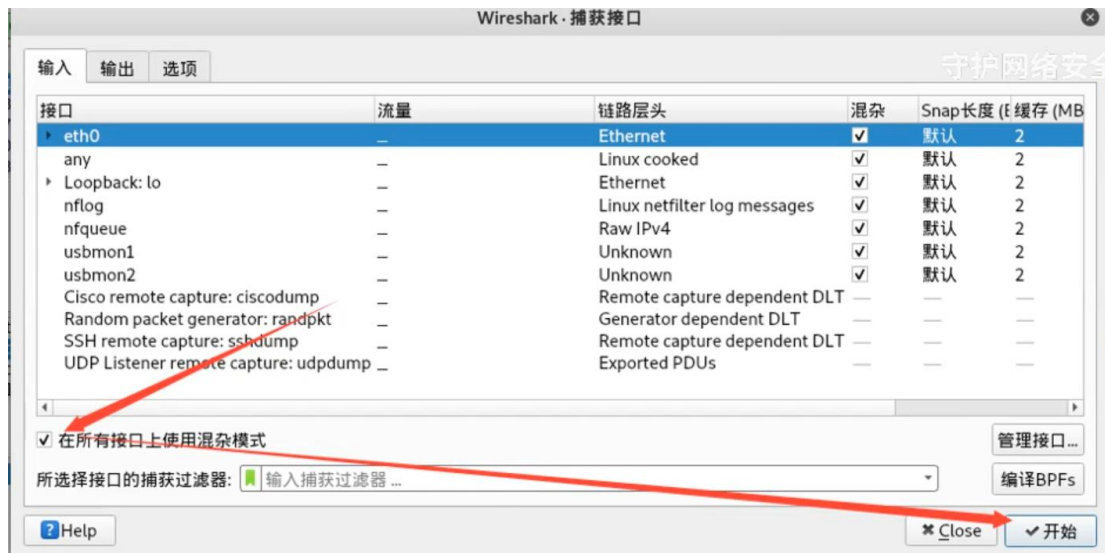
以太网适配器 以太网 2:
```

## 2.3 混杂模式介绍

1、混杂模式概述:混杂模式就是接收所有经过网卡的数据包，包括不是发给本机的包，即不验证 MAC 地址。普通模式下网卡只接收发给本机的包（包括广播包）传递给上层程序，其它的包一律丢弃。

一般来说，混杂模式不会影响网卡的正常工作，多在网络监听工具上使用。F

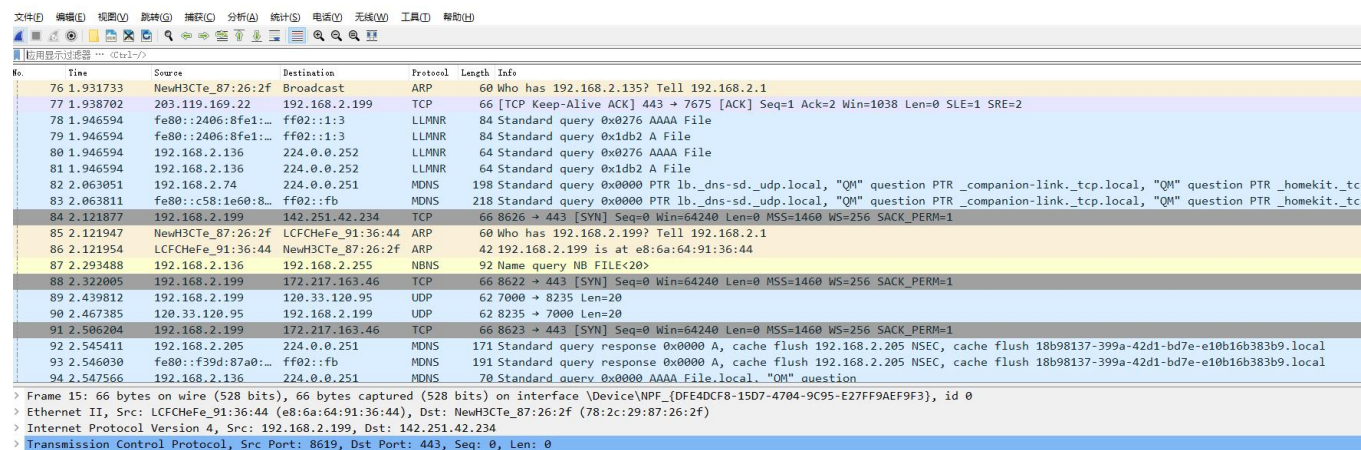
## 2.4 如何开启混杂模式？



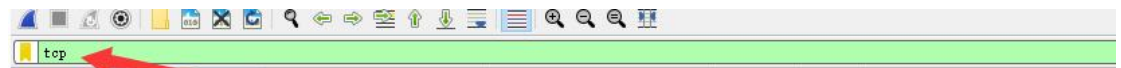
这样就开启了。默认就是开启混杂模式。←

## 章节三：Wireshark 过滤器使用

### 3.1 开启以上的混淆模式，抓取接口上使用混杂模式直接进行抓包



例 1：对 TCP 协议的包进行筛选



No.	Time	Source	Destination	Protocol	Length	Info
232	8.184731	192.168.2.199	49.79.227.194	TCP	66	8632 → 443 [SYN] Seq=
234	8.205747	203.119.169.22	192.168.2.199	TCP	66	[TCP Keep-Alive ACK]
235	8.209307	49.79.227.194	192.168.2.199	TCP	66	443 → 8632 [SYN, ACK]
236	8.209480	192.168.2.199	49.79.227.194	TCP	54	8632 → 443 [ACK] Seq=
237	8.219349	192.168.2.199	49.79.227.194	TLSv1...	571	Client Hello
238	8.245330	49.79.227.194	192.168.2.199	TCP	60	443 → 8632 [ACK] Seq=
239	8.246021	49.79.227.194	192.168.2.199	TLSv1...	1494	Server Hello
240	8.246021	49.79.227.194	192.168.2.199	TCP	1494	443 → 8632 [ACK] Seq=
241	8.246021	49.79.227.194	192.168.2.199	TLSv1...	538	Certificate, Server K
242	8.246119	192.168.2.199	49.79.227.194	TCP	54	8632 → 443 [ACK] Seq=
243	8.250364	49.79.227.194	192.168.2.199	TCP	538	[TCP Spurious Retrans
244	8.250437	192.168.2.199	49.79.227.194	TCP	66	[TCP Dup ACK 242#1] 8
245	8.251573	192.168.2.199	49.79.227.194	TLSv1...	180	Client Key Exchange,
247	8.276977	49.79.227.194	192.168.2.199	TLSv1...	105	Change Cipher Spec, E


## 例 2：筛选出 ACK 相关的包

SYN=1、ACK=0：客户端请求向服务端建立连接。

top.flags.ack == 0 and top.flags.syn == 1						
No.	Time	Source	Destination	Protocol	Length	Info
19	0.323381	192.168.2.199	172.217.163.42	TCP	66	4088 → 443 [SYN] Seq=0 Win=64240 L
274	5.456481	192.168.2.199	142.251.42.234	TCP	66	4093 → 443 [SYN] Seq=0 Win=64240 L
285	5.681220	192.168.2.199	142.251.42.234	TCP	66	4094 → 443 [SYN] Seq=0 Win=64240 L
361	8.577532	192.168.2.199	183.47.124.53	TCP	66	4102 → 80 [SYN] Seq=0 Win=64240 Le
398	9.390642	192.168.2.199	142.251.42.234	TCP	66	4104 → 443 [SYN] Seq=0 Win=64240 L
574	12.324649	192.168.2.199	142.251.42.234	TCP	66	4106 → 443 [SYN] Seq=0 Win=64240 L
581	12.390663	192.168.2.199	142.251.42.234	TCP	66	[TCP Retransmission] [TCP Port num
697	15.324844	192.168.2.199	142.251.42.234	TCP	66	[TCP Retransmission] [TCP Port num
751	17.457575	192.168.2.199	172.217.163.42	TCP	66	4110 → 443 [SYN] Seq=0 Win=64240 L
756	17.681275	192.168.2.199	172.217.163.42	TCP	66	4111 → 443 [SYN] Seq=0 Win=64240 L
774	18.391136	192.168.2.199	142.251.42.234	TCP	66	[TCP Retransmission] [TCP Port num
838	20.457687	192.168.2.199	172.217.163.42	TCP	66	[TCP Retransmission] [TCP Port num

## 例 3：抓取指定条件的包

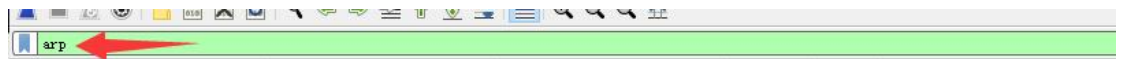
tcp.flags.fin == 1 当 FIN=1 时，表明数据已经发送完毕，要求释放连接



No.	Time	Source	Destination	Protocol	Length	Info
251	8.303416	192.168.2.199	49.79.227.194	TCP	54	8632 → 443 [FIN, ACK] S
253	8.328437	49.79.227.194	192.168.2.199	TCP	60	443 → 8632 [FIN, ACK] S
590	21.066498	192.168.2.199	180.163.203.54	TCP	54	2917 → 80 [FIN, ACK] Se
593	21.366702	192.168.2.199	180.163.203.54	TCP	54	[TCP Retransmission] 29
603	21.967827	192.168.2.199	180.163.203.54	TCP	54	[TCP Retransmission] 29
631	23.168159	192.168.2.199	180.163.203.54	TCP	54	[TCP Retransmission] 29
653	25.568481	192.168.2.199	180.163.203.54	TCP	54	[TCP Retransmission] 29
769	30.368578	192.168.2.199	180.163.203.54	TCP	54	[TCP Retransmission] 29
828	33.355944	192.168.2.199	182.254.116.117	TCP	54	8656 → 80 [FIN, ACK] Se
856	33.390715	182.254.116.117	192.168.2.199	TCP	60	80 → 8656 [FIN, ACK] Se
897	33.490130	192.168.2.199	183.47.99.22	TCP	54	8658 → 443 [FIN, ACK] S
900	33.491625	192.168.2.199	183.47.99.22	TCP	54	8657 → 443 [FIN, ACK] S
902	33.508883	183.47.99.22	192.168.2.199	TCP	60	443 → 8658 [FIN, ACK] S


## 例 4：筛选出 ARP 数据包





No.	Time	Source	Destination	Protocol	Length	Info
4	0.063466	ExtremeN_68:00:8c	Broadcast	ARP	60	Who has 192.168.2.1
31	0.907336	NewH3CTe_87:26:2f	Broadcast	ARP	60	Who has 192.168.2.1
76	1.931733	NewH3CTe_87:26:2f	Broadcast	ARP	60	Who has 192.168.2.1
85	2.121947	NewH3CTe_87:26:2f	LCFCHeFe_91:36:44	ARP	60	Who has 192.168.2.1
86	2.121954	LCFCHeFe_91:36:44	NewH3CTe_87:26:2f	ARP	42	192.168.2.199 is at
101	2.927047	NewH3CTe_87:26:2f	Broadcast	ARP	60	Who has 192.168.2.1
128	3.928459	NewH3CTe_87:26:2f	Broadcast	ARP	60	Who has 192.168.2.1
150	4.951858	NewH3CTe_87:26:2f	Broadcast	ARP	60	Who has 192.168.2.1
164	5.947188	NewH3CTe_87:26:2f	Broadcast	ARP	60	Who has 192.168.2.1
198	6.884490	HuaweiTe_ee:a0:cc	Broadcast	ARP	60	Who has 192.168.2.1
200	6.947121	NewH3CTe_87:26:2f	Broadcast	ARP	60	Who has 192.168.2.1

### 例 5：筛选出 udp 属于传输层的数据包




No.	Time	Source	Destination	Protocol	Length	Info
19360	16.805174540	124.134.234.1	192.168.1.93	UDP	75	5545 → 30633 Len=33
19361	16.807749867	110.230.253.1	192.168.1.93	UDP	75	57879 → 30633 Len=33
19362	16.812799737	192.168.1.93	110.16.0.46	UDP	1107	30633 → 14336 Len=1065
19363	16.812806957	192.168.1.93	113.3.88.59	UDP	1107	30633 → 34472 Len=1065
19364	16.812831858	192.168.1.93	124.134.234.156	UDP	1107	30633 → 5545 Len=1065
19365	16.812911088	192.168.1.93	110.230.253.108	UDP	1107	30633 → 57879 Len=1065
19366	16.816183342	114.248.70.60	192.168.1.93	UDP	75	58581 → 30633 Len=33
19367	16.817372873	58.21.216.151	192.168.1.93	UDP	75	30528 → 30633 Len=33
19368	16.817822637	113.6.115.233	192.168.1.93	UDP	75	36972 → 30633 Len=33
19369	16.818718248	113.3.88.59	192.168.1.93	UDP	79	34472 → 30633 Len=37
19370	16.818721474	182.117.65.31	192.168.1.93	UDP	75	21778 → 30633 Len=33
19371	16.825698806	110.230.253.1	192.168.1.93	UDP	75	57879 → 30633 Len=33

我们使用过滤器输入“udp”以筛选出 udp 报文。但是为什么输入 udp 之后出现那么多种协议呢?原因就是 oicq 以及 dns 都是基于 udp 的传输层之上的协议-

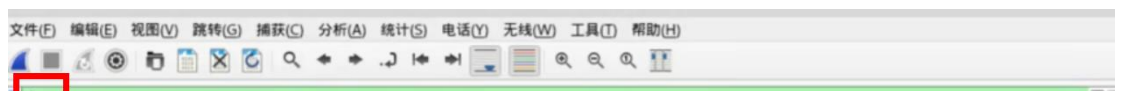
扩展:客户端向 DNS 服务器查询域名,一般返回的内容都不超过 512 字节,用 UDP 传输即可。不用经过三次握手,这样 DNS 服务器负载更低,响应更快。理论上说,客户端也可以指定向 DNS 服务器查询时用 TCP,但事实上,很多 DNS 服务器进行配置的时候,仅支持 UDP 查询包。-

### 例 6：http 请求



No.	Time	Source	Destination	Protocol	Length	Info
6523	108.589670808	192.168.1.53	42.101.56.31	HTTP	479	GET /5foIcy0a2gI2n2jgoY3K/
6524	108.590270053	192.168.1.53	42.101.56.31	HTTP	477	GET /5foIcy0a2gI2n2jgoY3K/
6525	108.590721764	192.168.1.53	42.101.56.31	HTTP	461	GET /5foIcy0a2gI2n2jgoY3K/
6528	108.592475697	42.101.56.31	192.168.1.53	HTTP	1633	HTTP/1.1 200 OK (PNG)
6537	108.595911630	42.101.56.31	192.168.1.53	HTTP	4947	HTTP/1.1 200 OK (PNG)

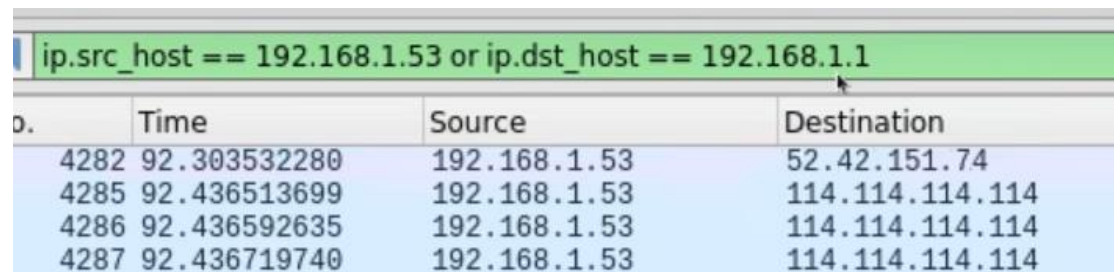
### 例 6：dns 数据包



No.	Time	Source	Destination	Protocol	Length	Info
16203	13.990289757	192.168.1.1	192.168.1.53	DNS	104	Standard query response 0x2bf5 AAAA ss0.baidu.com CNAME
16206	13.992200561	192.168.1.1	192.168.1.53	DNS	120	Standard query response 0x295f A ss1.baidu.com CNAME ssl
16207	13.992310208	192.168.1.1	192.168.1.53	DNS	107	Standard query response 0x3f4a AAAA ss1.baidu.com CNAME
16239	14.003432665	192.168.1.1	192.168.1.53	DNS	120	Standard query response 0x90b8 A ss2.baidu.com CNAME ssl
16241	14.003465036	192.168.1.1	192.168.1.53	DNS	104	Standard query response 0x9dd7 AAAA ss2.baidu.com CNAME
16242	14.003468561	192.168.1.1	192.168.1.53	DNS	120	Standard query response 0x5b76 A ss3.baidu.com CNAME ssl
16250	14.004390898	192.168.1.1	192.168.1.53	DNS	104	Standard query response 0x9e7b AAAA ss3.baidu.com CNAME
17074	14.630318133	192.168.1.53	192.168.1.1	DNS	72	Standard query 0x3d49 A trace.qq.com
17075	14.630321381	192.168.1.53	192.168.1.1	DNS	72	Standard query 0x0877 AAAA trace.qq.com
17085	14.637289577	192.168.1.53	192.168.1.1	DNS	75	Standard query 0xfec6 A pingfore.qq.com
17087	14.637459246	192.168.1.53	192.168.1.1	DNS	75	Standard query 0x4bd2 AAAA pingfore.qq.com
17089	14.639919574	192.168.1.1	192.168.1.53	DNS	157	Standard query response 0x3d49 A trace.qq.com CNAME btra

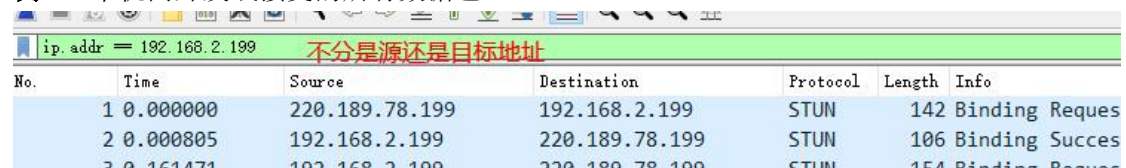
### 例 7：数据包条件筛选

其实我们不仅可以对协议类型进行筛选，我们还有跟多的筛选条件，比如源地址目的地址等等例 6:筛选源地址是 192.168.1.53 或目的地址是 192.168.1.1



No.	Time	Source	Destination
4282	92.303532280	192.168.1.53	52.42.151.74
4285	92.436513699	192.168.1.53	114.114.114.114
4286	92.436592635	192.168.1.53	114.114.114.114
4287	92.436719740	192.168.1.53	114.114.114.114

### 例 8：本机向外发或接受的所有数据包



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	220.189.78.199	192.168.2.199	STUN	142	Binding Request
2	0.000805	192.168.2.199	220.189.78.199	STUN	106	Binding Success
3	0.161471	192.168.2.199	220.189.78.199	STUN	154	Binding Request

## 章节四：

### 4.1 Arp 协议解读

协议分析的时候我们关闭混淆模式，避免一些干扰的数据包存在

常用协议分析-ARP 协议(英语:Address Resolution Protocol，细与：AKP) 定一个通过解析网层地址来找寻数据链路层地址的网络传输协议，它在 IPv4 中极其重要。ARP 是通过网络地址来定位 MAC 地址。

主机向目标机器发送信息时，ARP 请求广播到局域网络上的所有主机，并接收返回消息，以此确定目标的物理地址；收到返回消息后将该 IP 地址和物理地址，存入本机 ARP 缓存中并保留一定时间，下次请求时直接查询 ARP 缓存以节约资源。

### 4.2 如果找到 ARP 包？

arp						
No.	Time	Source	Destination	Protocol	Length	Info
130...	161.445462	NewH3CTe_87:26:2f	LCFCHeFe_91:36:44	ARP	60	Who has 192.168.2.199? Tell 192.168.2.1
130...	161.445480	LCFCHeFe_91:36:44	NewH3CTe_87:26:2f	ARP	42	192.168.2.199 is at e8:6a:64:91:36:44
130...	161.463224	NewH3CTe_87:26:2f	Broadcast	ARP	60	Who has 192.168.2.195? Tell 192.168.2.1
130...	161.538587	HuaweiTe_ee:a0:cc	Broadcast	ARP	60	Who has 192.168.2.1? Tell 192.168.2.80
130...	161.569594	NewH3CTe_87:26:2f	Broadcast	ARP	60	Who has 192.168.2.79? Tell 192.168.2.1
132...	161.905826	NewH3CTe_87:26:2f	Broadcast	ARP	60	Who has 192.168.2.145? Tell 192.168.2.1
132...	162.205460	NewH3CTe_87:26:2f	Broadcast	ARP	60	Who has 192.168.2.66? Tell 192.168.2.1
132...	162.375574	NewH3CTe_87:26:2f	Broadcast	ARP	60	Who has 192.168.2.128? Tell 192.168.2.1
132...	162.455532	NewH3CTe_87:26:2f	Broadcast	ARP	60	Who has 192.168.2.195? Tell 192.168.2.1
132...	162.565473	NewH3CTe_87:26:2f	Broadcast	ARP	60	Who has 192.168.2.79? Tell 192.168.2.1
132...	163.242078	NewH3CTe_87:26:2f	Broadcast	ARP	60	Who has 172.17.0.1? Tell 172.17.1.1
132...	163.455518	NewH3CTe_87:26:2f	Broadcast	ARP	60	Who has 192.168.2.195? Tell 192.168.2.1

> Frame 13044: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF\_{DFE4DCF8-15D7-4704-9...}

> Ethernet II, Src: NewH3CTe\_87:26:2f (78:2c:29:87:26:2f), Dst: LCFCHeFe\_91:36:44 (e8:6a:64:91:36:44)

✓ Address Resolution Protocol (request) 发的请求包/request

Hardware type: Ethernet (1) 硬件类型-标识

Protocol type: IPv4 (0x0800) 网络层协议

Hardware size: 6 硬件地址长度-MAC地址长度 6字节 48位

Protocol size: 4 协议地址长度4字节, 32位

Opcode: request (1) 操作码类型: 1请求 2响应

Sender MAC address: NewH3CTe\_87:26:2f (78:2c:29:87:26:2f) 源MAC地址与IP, 发送者是网关

Sender IP address: 192.168.2.1

Target MAC address: 00:00:00\_00:00:00 (00:00:00:00:00:00)

Target IP address: 192.168.2.199 目标本机的MAC 不知道, 目标IP是本机

## 章节五：ICMP 抓包与解读

先 Ping 一个地址，获得 ICMP 包

```
C:\Users\10762>ping 192.168.2.1 /t

正在 Ping 192.168.2.1 具有 32 字节的数据:
来自 192.168.2.1 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.2.1 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.2.1 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.2.1 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.2.1 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.2.1 的回复: 字节=32 时间<1ms TTL=128
```

再筛选过滤 icmp 格式包

icmp						
No.	Time	Source	Destination	Protocol	Length	Info
92	2.207190	192.168.2.199	121.14.77.201	ICMP	74	Echo (ping) request id=0x0001, seq=141/36096, ttl=64 (reply in 93)
93	2.227365	121.14.77.201	192.168.2.199	ICMP	74	Echo (ping) reply id=0x0001, seq=141/36096, ttl=52 (request in 92)

> Frame 92: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF\_{959D08A3-3EB0-43E2-A24D-D47B9D856284}, id 0

> Ethernet II, Src: LCFCHeFe\_91:36:44 (e8:6a:64:91:36:44), Dst: NewH3CTe\_87:26:2f (78:2c:29:87:26:2f)

> Internet Protocol Version 4, Src: 192.168.2.199, Dst: 121.14.77.201 基于IP协议

✓ Internet Control Message Protocol 这是ICMP

Type: 8 (Echo (ping) request) 发的是ping请求 8代表请求包, 0代表响应包

Code: 0

Checksum: 0x4cce [correct] 检验盒: 数据包完整性检验

[Checksum Status: Good] 检验状态

Identifier (BE): 1 (0x0001)

Identifier (LE): 256 (0x0100)

Sequence Number (BE): 141 (0x008d)

Sequence Number (LE): 36096 (0x8d00)

[Response frame: 93]

> Data (32 bytes)



```
Internet Control Message Protocol
Type: 0 (Echo (ping) reply)
Code: 0 协议类型0代码0表示回显应答报文
Checksum: 0x852d [correct] 校验和
[Checksum Status: Good]
Identifier (BE): 2792 (0x0ae8)
Identifier (LE): 59402 (0xe80a)
Sequence number (BE): 1 (0x0001)
Sequence number (LE): 256 (0x0100)
[Request frame: 51] 请求帧的序列号
[Response time: 10.136 ms] 响应时间
Timestamp from icmp data: May 15, 2019 11:43:53.000000000 CST
[Timestamp from icmp data (relative): 0.677497174 seconds]
Data (48 bytes) 填充数据, 共48字节
```

工作过程:-

本机发送一个 ICMP Echo Request 的包

接受方返回一个 ICMP Echo Reply, 包含了接受到数据拷贝和一些其他指令-

## 章节六: 常用协议分析-TCP 的 3 次握手协议

清空数据包然后筛选 tcp 开始抓包 e



选中一个包, 进行解读



```

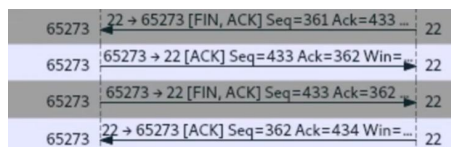
Transmission Control Protocol, Src Port: 53209, Dst Port: 22,
  Source Port: 53209 源端口
  Destination Port: 22 目的端口
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence number: 1 (relative sequence number) Seq=1 等于上一帧的确认序列号
  [Next sequence number: 1 (relative sequence number)]
  Acknowledgment number: 1 (relative ack number) ACK=1 确认序号1, 上一帧的序号+1
  0101 .... = Header Length: 20 bytes (5)
  Flags: 0x010 (ACK) Flags=ACK
  Window size value: 8212
  [Calculated window size: 2102272]
  [Window size scaling factor: 256]
  Checksum: 0x8277 [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  [SEQ/ACK analysis]
  [Timestamps]

  Flags: 0x010 (ACK)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    .... 0... = Congestion Window Reduced (CWR): Not set
    .... .0.. = ECN-Echo: Not set
    .... ..0. = Urgent: Not set
    .... ...1 = Acknowledgment: Set
    .... .... 0... = Push: Not set

```

## 章节 7: Tcp 连接断开的 4 次挥手

我们分析一下过程,我们在终端输入 EXIT 实际上是在我们 Kali 上执行的命令,表示我们 SSHD 的 Server 端向客户端发起关闭链接请求。



我们分析一下过程,我们在终端输入 EXIT 实际上是在我们 Kali 上执行的命令,表示我们 SSHD 的 Server 端向客户端发起关闭链接请求。

### 第一次挥手:

服务端发送一个[FIN+ACK],表示自己没有数据要发送了,想断开连接,并进入 FIN\_WAIT\_1 状态

### 第二次挥手:

客户端收到 FIN 后,知道不会再有数据从服务端传来,发送 ACK 进行确认,确认序号为收到序号+1(与 SYN 相同,一个 FIN 占用一个序号),客户端进入 CLOSE\_WAIT 状态.

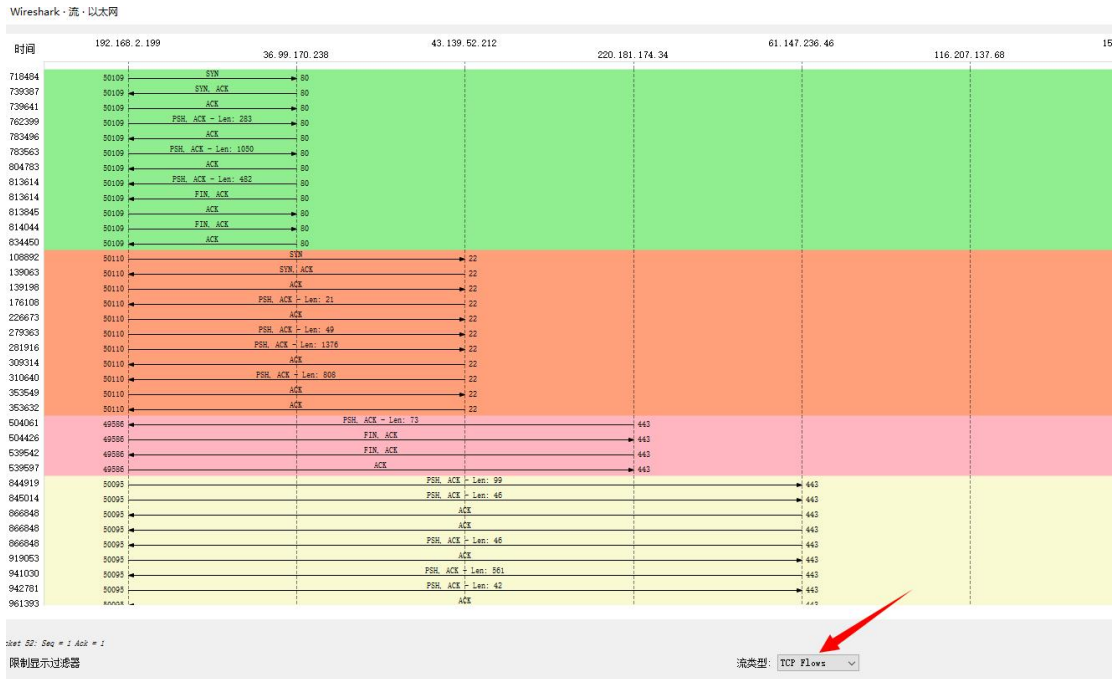
### 第三次挥手:

客户端发送[FIN+ACK]给对方,表示自己没有数据要发送了,客户端进入 LAST\_ACK 状态,然后直接断开 TCP 会话的连接,释放相应的资源。

### 第四次挥手:

服务端收到了客户端的 FIN 信令后,进入 TIMED\_WAIT 状态并发往 ACK 确认消息。服务端在 TIMED\_WAIT 状态下,等待一段时间,没有数据到来,就认为对面已经收到了自己发送的 ACK 并正确关闭了进入 CLOSE 状态,自己也断开了 TCP 连接,释放所有资源。当客户端收到服务端的 ACK 回应后,会进入 CLOSE 状态并关闭本端的会话接口,释放相应资源。

## 数据流的统计



## 章节 8：常用协议分析-HTTP 协议

HTTP 是 TCP 的上层协议，所以我们过滤 TCP 的数据会包含 HTTP 协议的数据包

No.	Time	Source	Destination	Protocol	Length	Info
102	2.711014	1.192.192.201	192.168.2.199	TCP	60	80 → 51955 [FIN, ACK] Seq=442 A
103	2.711264	192.168.2.199	1.192.192.201	TCP	54	51955 → 80 [ACK] Seq=1013 Ack=4
104	2.711555	192.168.2.199	1.192.192.201	TCP	54	51955 → 80 [FIN, ACK] Seq=1013
105	2.737231	1.192.192.201	192.168.2.199	TCP	60	80 → 51955 [ACK] Seq=443 Ack=10
106	2.821790	192.168.2.199	1.192.192.201	TCP	66	51956 → 80 [SYN] Seq=0 Win=6424
107	2.843821	1.192.192.201	192.168.2.199	TCP	62	80 → 51956 [SYN, ACK] Seq=0 Ack
108	2.844045	192.168.2.199	1.192.192.201	TCP	54	51956 → 80 [ACK] Seq=1 Ack=1 Wi
109	2.860139	192.168.2.199	1.192.192.201	TCP	337	51956 → 80 [PSH, ACK] Seq=1 Ack
110	2.881746	1.192.192.201	192.168.2.199	TCP	60	80 → 51956 [ACK] Seq=1 Ack=284
111	2.881876	192.168.2.199	1.192.192.201	HTTP	1112	POST /cloudquery.php HTTP/1.1
112	2.903512	1.192.192.201	192.168.2.199	TCP	60	80 → 51956 [ACK] Seq=1 Ack=1342
113	2.911683	1.192.192.201	192.168.2.199	HTTP	536	HTTP/1.1 200 OK
114	2.911683	1.192.192.201	192.168.2.199	TCP	60	80 → 51956 [FIN, ACK] Seq=483 A
115	2.911829	192.168.2.199	1.192.192.201	TCP	54	51956 → 80 [ACK] Seq=1342 Ack=4

## 章节 9：黑客利用 wireshak 获取用户名和密码实战

准备以下物料，安装好即可

- 1, NGINX
- 2, VM 虚拟机
- 3, kali 操作系统





