

实验内容

- Ubuntu18.10安装 wireshark
- 运行wireshark并且进行抓包

实验步骤

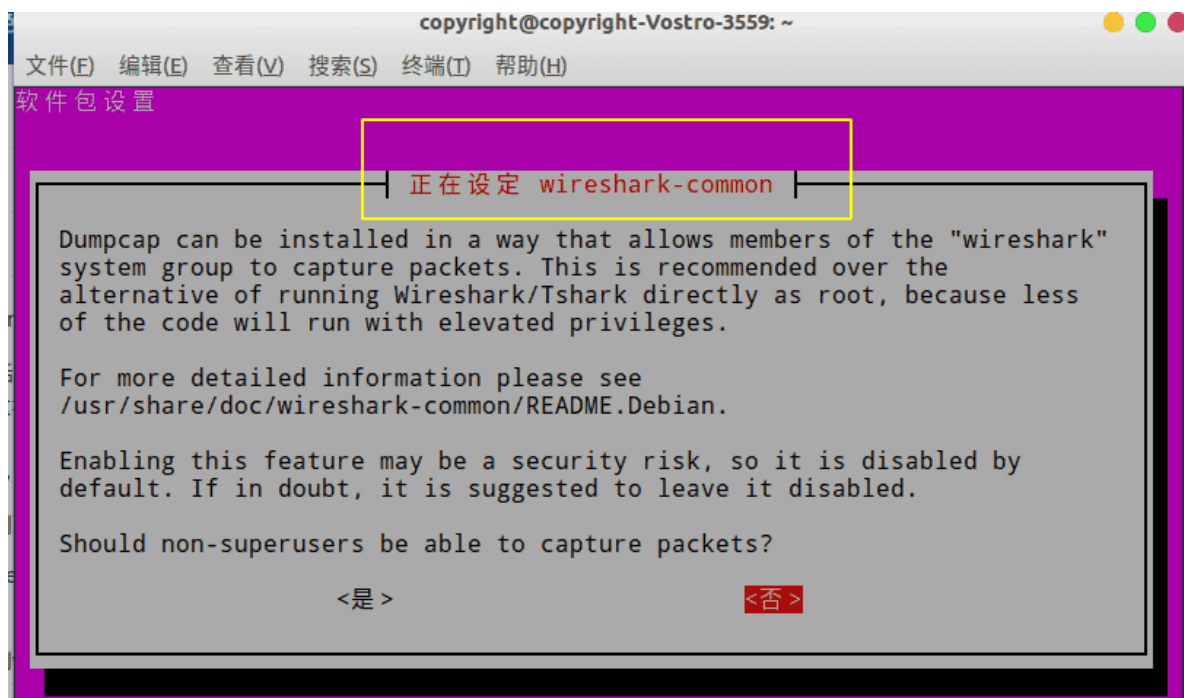
安装wireshark

- 环境：Ubuntu18.10
- 版本：Stable Wireshark
- 参考：[链接](#)

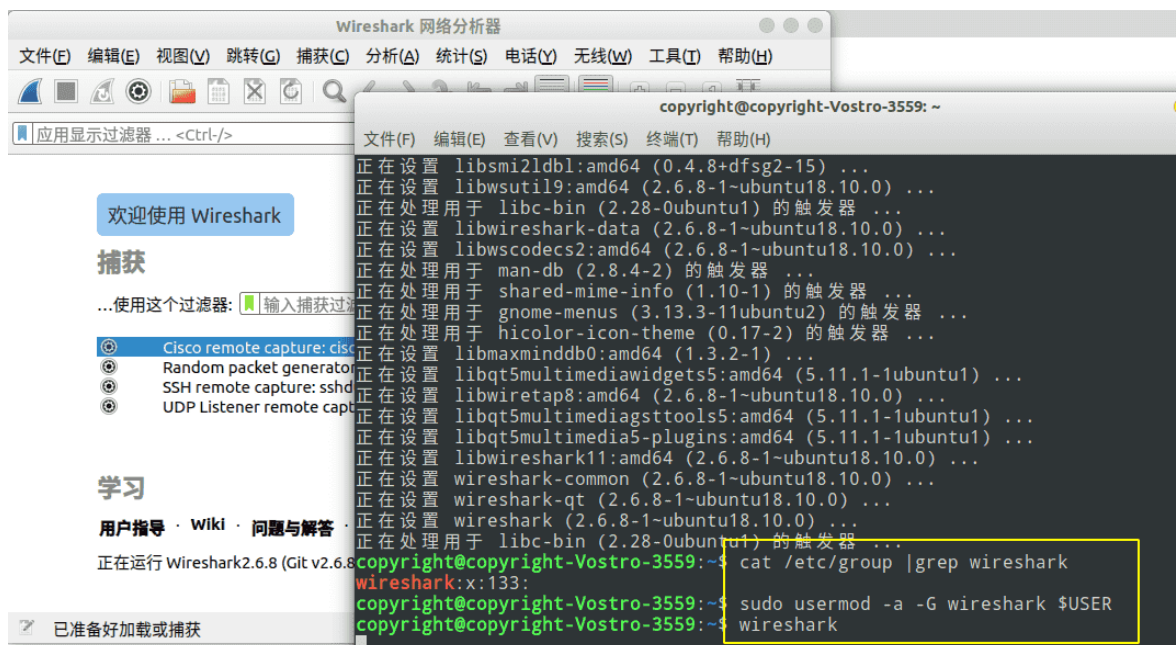
命令行安装

```
copyright@copyright-Vostro-3559:~$ sudo apt install wireshark
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不需要了：
  dwz libarchive-cpio-perl libfile-stripnondeterminism-perl
  libmail-sendmail-perl libsys-hostname-long-perl po-debconf
使用 'sudo apt autoremove'来卸载它(它们)。
将会同时安装下列软件：
  libmaxminddb0 libnl-route-3-200 libqt5multimedia5 libqt5multimedia5-plug
```

安装 wireshark-common 选择 yes

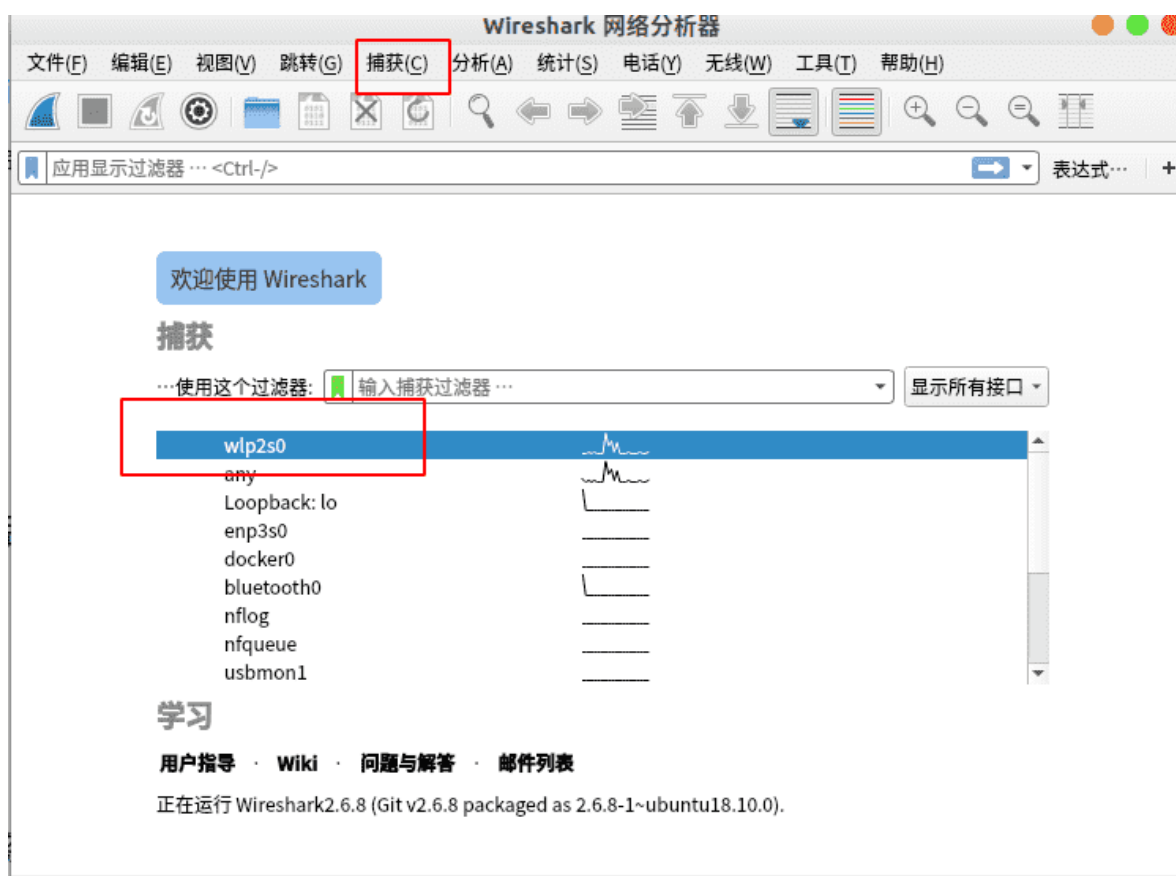


添加到 \$USER 变量，以后运行时不用加 sudo 权限

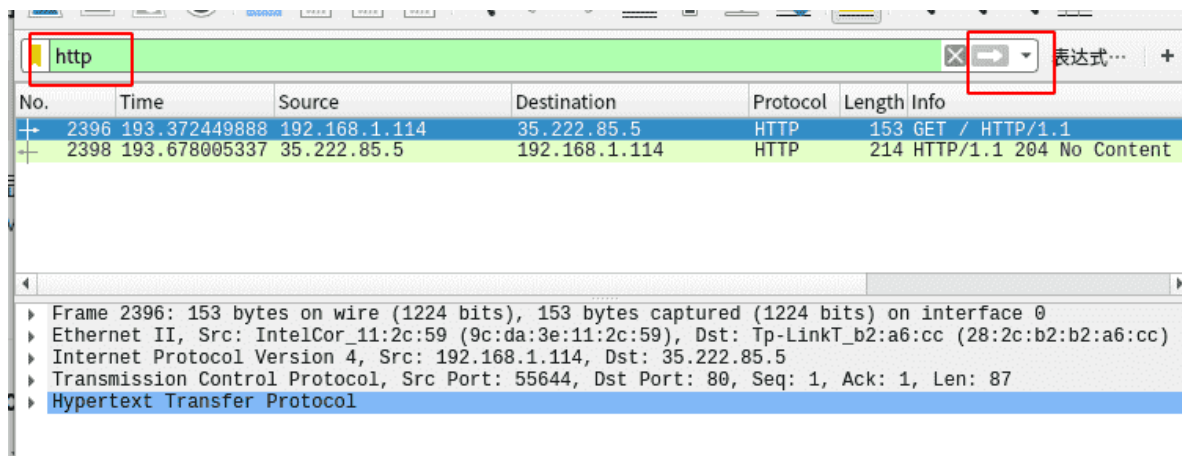


运行wireshark

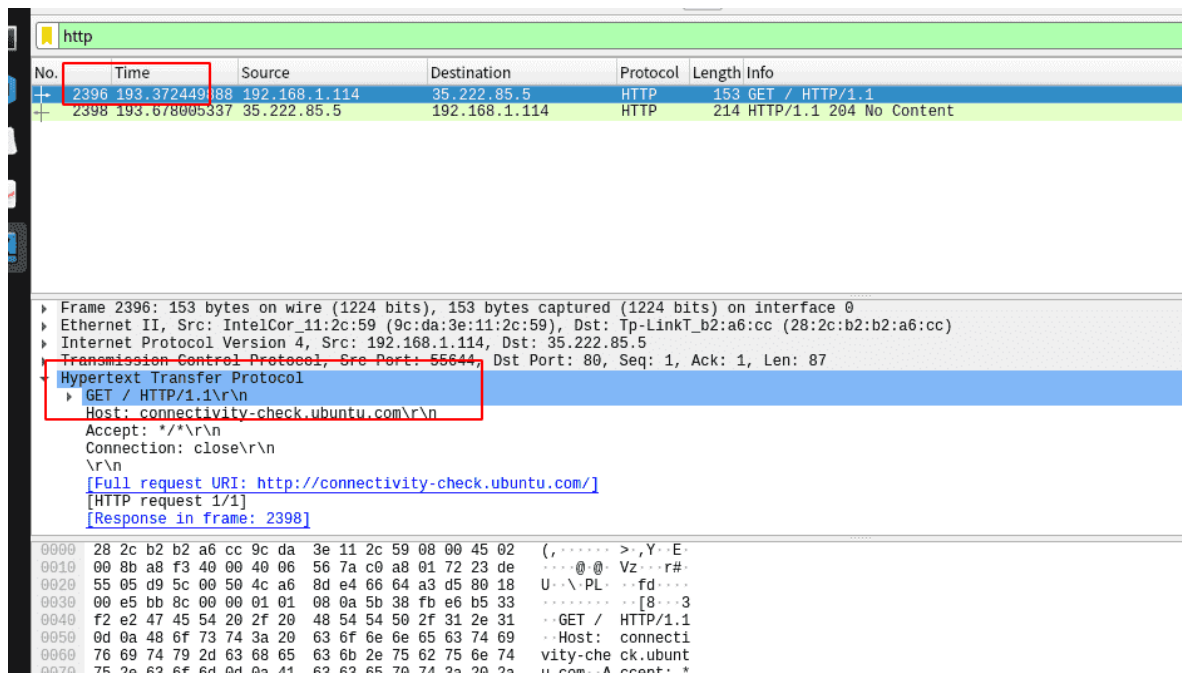
命令行启动软件，因为现在是WiFi环境，选择无线网卡 wlp2s0，点击捕获->开始。



在浏览器输入一个 HTTP 协议的网页（最好是之前没浏览过的，好像本地缓存过的地址不会被 wireshark 抓到？）。输入完成后，点击 wireshark 中的捕获->停止，在过滤器列表输入 http (小写)，并点击右边的箭头。



捕获的 HTTP 的请求消息



捕获的 HTTP 响应消息（紫色箭头的方向可以区分是请求消息还是响应消息）

The image shows a Wireshark packet capture of an HTTP response. The packet list at the top shows packet 2398 as an HTTP response from 35.222.85.5 to 192.168.1.114. The packet details pane shows the HTTP response structure, including the status code 204. The packet bytes pane shows the raw data.

No.	Time	Source	Destination	Protocol	Length	Info
2396	193.372449888	192.168.1.114	35.222.85.5	HTTP	153	GET / HTTP/1.1
2398	193.678005337	35.222.85.5	192.168.1.114	HTTP	214	HTTP/1.1 204 No Content

Frame 2398: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) on interface 0
 Ethernet II, Src: Tp-LinkT_b2:a6:cc (28:2c:b2:b2:a6:cc), Dst: IntelCor_11:2c:59 (9c:da:3e:11:2c:59)
 Internet Protocol Version 4, Src: 35.222.85.5, Dst: 192.168.1.114
 Transmission Control Protocol, Src Port: 80, Dst Port: 55644, Seq: 1, Ack: 88, Len: 148
 Hypertext Transfer Protocol
 HTTP/1.1 204 No Content
 Date: Mon, 30 Mar 2020 15:50:04 GMT
 Server: Apache/2.4.18 (Ubuntu)
 X-NetworkManager-Status: online
 Connection: close
 [HTTP response 1/1]
 [Time since request: 0.305555449 seconds]

图中的响应状态码是204

204 No Content

服务器成功处理了请求，但不需要返回任何实体内容，并且希望返回更新了元信息。响应可能通过实体头部的形式，返回新的或更新后的元信息。如果存在这些头部信息，则应当与所请求的变量相呼应。如果客户端是浏览器的话，那么用户浏览器应保留发送了该请求的页面，而不产生任何文档视图上的变化，即使按照规范新的或更新后的元信息应当被应用到用户浏览器活动视图中的文档。由于204响应被禁止包含任何消息体，因此它始终以消息头后的第一个空行结尾。

实验结果

- 列出上述出现在未过滤的分组列表窗口的协议列中的3种不同的协议。

在捕获的过程中还出现 TCP ， UDP ， TLSV1.2 ， TLSV1.3 这几种协议

The image shows a Wireshark packet capture with various protocols. The packet list at the top shows several packets with different protocols. The packet details pane shows the structure of a TLSV1.3 packet.

No.	Time	Source	Destination	Protocol	Length	Info
78	15.103984009		59.149.170.8	TCP	321	[TCP Retransmission]
79	15.232002231		59.149.170.8	TCP	127	[TCP Retransmission]
80	15.236953312		192.168.1.114	TCP	66	443 → 44652 [ACK]
81	15.236979501		192.168.1.114	TLSv1.3	127	Application Data
82	15.237007160		59.149.170.8	TCP	66	44652 → 443 [ACK]
83	15.237355188		59.149.170.8	TLSv1.3	677	Application Data
84	15.450701317		192.168.1.114	TCP	66	443 → 44652 [ACK]

应用显示过滤器: <Ctrl-/> 表达式: +

No.	Time	Source	Destination	Protocol	Length	Info
96	15.879371157	192.168.1.114	59.149.170.8	TLSv1.3	127	Application Data
97	16.079071444	59.149.170.8	192.168.1.114	TCP	66	443 → 44652 [ACK]
98	17.119991633	192.168.1.114	59.149.170.8	TCP	127	[TCP Retransmission]
99	17.169371258	59.149.170.8	192.168.1.114	TCP	66	443 → 44270 [ACK]
100	17.819990492	192.168.1.130	255.255.255.255	UDP	42	56342 → 21161 Len: 32
101	20.223965670	192.168.1.114	52.114.74.43	TCP	66	[TCP Retransmission]
102	26.367991411	192.168.1.114	117.18.237.29	TCP	66	57122 → 80 [ACK]

Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0

- 从HTTP GET消息发送到HTTP OK回复需要多长时间？

GET 消息发送时间

Arrival Time: Mar 30, 2020 23:50:04.019483996 CST

```
Frame 2396: 153 bytes on wire (1224 bits), 153 bytes captured (1224 bits) on interface 0
  Interface id: 0 (wlp2s0)
  Encapsulation type: Ethernet (1)
  Arrival Time: Mar 30, 2020 23:50:04.019483996 CST
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1585583404.019483996 seconds
  [Time delta from previous captured frame: 0.000420824 seconds]
  [Time delta from previous displayed frame: 0.000000000 seconds]
  [Time since reference or first frame: 193.372449888 seconds]
  Frame Number: 2396
  Frame Length: 153 bytes (1224 bits)
  Capture Length: 153 bytes (1224 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:ip:tcp:http]
  [Coloring Rule Name: HTTP]
```

这次实验收到的是 204 No Content 响应状态码

Arrival Time: Mar 30, 2020 23:50:04.325039445 CST

```
Frame 2398: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) on interface 0
  Interface id: 0 (wlp2s0)
  Encapsulation type: Ethernet (1)
  Arrival Time: Mar 30, 2020 23:50:04.325039445 CST
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1585583404.325039445 seconds
  [Time delta from previous captured frame: 0.000217327 seconds]
  [Time delta from previous displayed frame: 0.305555449 seconds]
  [Time since reference or first frame: 193.678005337 seconds]
  Frame Number: 2398
  Frame Length: 214 bytes (1712 bits)
  Capture Length: 214 bytes (1712 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  0000  9c da 3e 11 2c 59 28 2c b2 b2 a6 cc 08 00 45 00  ..>.,Y(, .....E.
```

所以从 HTTP GET 消息发送到 HTTP No Content 回复需要的时间为 0.305555449

参考资料

- [安装wireshark](#)
- [实验文档翻译](#)