

实验2：配置Web服务器，编写简单页面，分析交互过程

姓名：丁彦添

学号：1911406

实验要求

(1) 搭建Web服务器（自由选择系统），并制作简单的 Web 页面，包含简单文本信息（至少包含专业、学号、姓名）和自己的LOGO。

(2) 通过浏览器获取自己编写的Web页面，使用 Wireshark 捕获浏览器与 Web 服务器的交互过程，并进行简单的分析说明。

搭建Web服务器，制作简单的Web页面

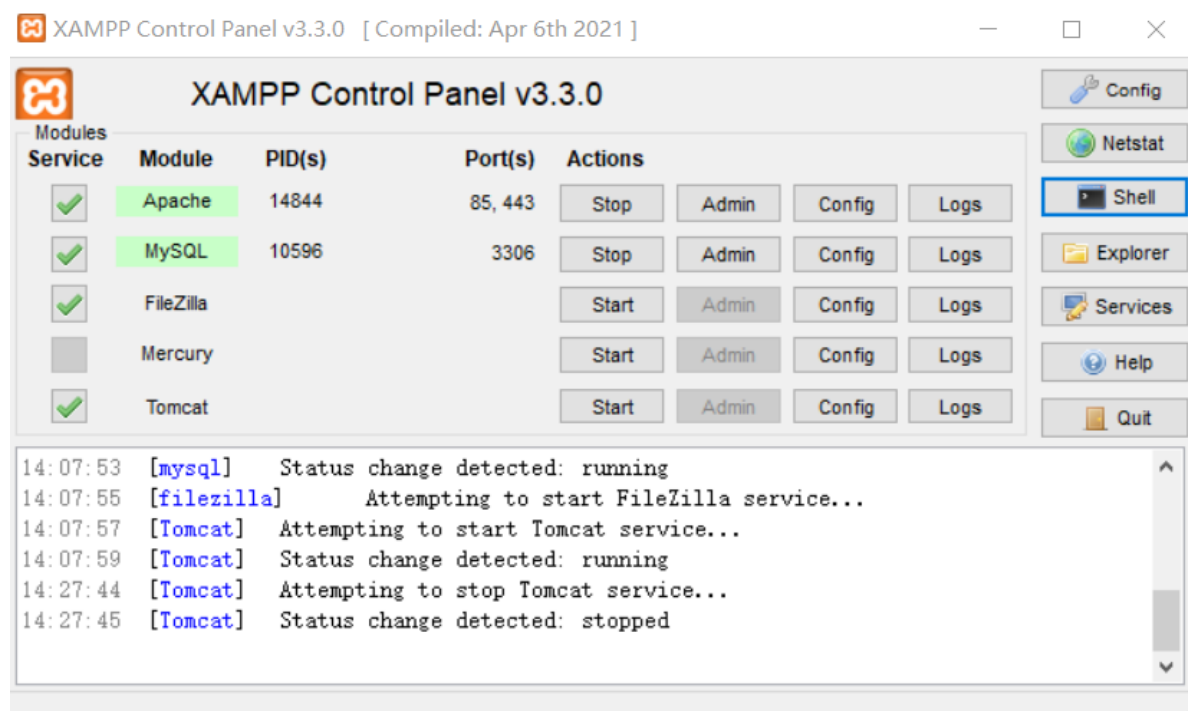
采用环境

Web 服务器运行环境：Windows 10 XAMPP Apache v3.3.0

Wireshark 运行环境：Parrot OS 5.0 Wireshark v3.4.4

Web 服务器搭建

使用 XAMPP 下的 Apache 功能。



选择 Apache 的 start，服务器就可以运行了。

这里需要注意，Apache 的默认端口是 80 端口，为了避免某些软件会占用 Apache 的 80 端口，需要在 Apache 的配置文件中修改 Listen 和 localhost 的端口号。本次实验将 Listen 和 localhost 的端口号改为了 85。

Listen 修改端口：

```
#  
# Listen: Allows you to bind Apache to specific IP addresses and/or  
# ports, instead of the default. See also the <VirtualHost>  
# directive.  
#  
# Change this to Listen on specific IP addresses as shown below to  
# prevent Apache from glomming onto all bound IP addresses.  
#  
#Listen 12.34.56.78:80  
Listen 85
```

localhost 修改端口：

```
#  
# ServerName gives the name and port that the server uses to identify itself.  
# This can often be determined automatically, but we recommend you specify  
# it explicitly to prevent problems during startup.  
#  
# If your host doesn't have a registered DNS name, enter its IP address here.  
#  
ServerName localhost:85
```

制作简单的Web页面

本次实验由于时间紧张，仅完成了作业要求的部分

内容仅仅包括专业、学号、姓名、少量文字和一张图片（png类型）。大致截图如下：

计算机网络 Lab 2 配置Web服务器，编写简单页面，分析交互过程

姓名：丁彦添

学号：1911406

专业：计算机科学与技术

实验要求：

(1) 搭建Web服务器（自由选择系统），并制作简单的Web页面，包含简单文本信息（至少包含专业、学号、姓名）和自己的LOGO。

(2) 通过浏览器获取自己编写的Web页面，使用Wireshark捕获浏览器与Web服务器的交互过程，并进行简单的分析说明。

(3) 提交实验报告。

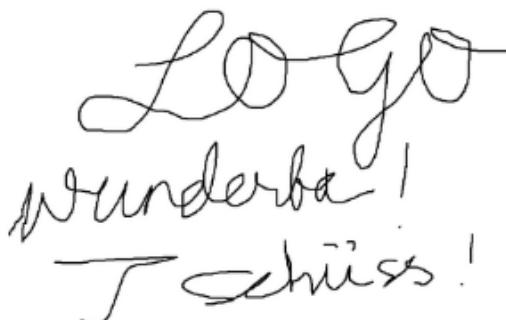
评分标准：

(1) 功能实现：Web服务器搭建（提交HTML文档）（20分）

Wireshark捕获交互过程（提交捕获文件）（30分）

(2) 演示并讲解（20分）(3) 实验报告（30分）

简单乱画个logo



Logo
Wunderba!
Tschüss!

webpage很简陋，见谅~

具体的 HTML 代码此处不贴出来，见附件。

将 HTML 及相关文件放入 XAMPP\htdocs 相关目录下，就可以访问了。

通过浏览器获取自己编写的Web页面

将服务器端计算机和客户端计算机连接同一个局域网下。

查看服务器端 IPv4 地址为：10.130.249.231

通过客户端 Web 浏览器输入“IP地址:端口号”（此处应该填：10.130.249.231:85）即可访问服务器端的 Web 页面。

WireShark 捕获交互过程

在客户端终端输入命令 `sudo wireshark` 打开 Wireshark 网络分析器。

由于是 WLAN 连接，选择 WLAN。

为了防止捕获的包冲突，仅捕获来自服务器端口号为 85 的包。只需要在过滤器 filters 中输入：

`ip.addr == 10.130.249.231 and (tcp.port in {85} or udp.port in {85})`。

ip.addr == 10.130.249.231 and (tcp.port in {85} or udp.port in {85})						
No.	Time	Source	Destination	Protocol	Length	Info
59	22.930261435	192.168.137.181	10.130.249.231	TCP	74	51012 → 85 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3167702040 TSe...
60	22.931842833	10.130.249.231	192.168.137.181	TCP	66	85 → 51012 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
61	22.931922787	192.168.137.181	10.130.249.231	TCP	54	51012 → 85 [ACK] Seq=1 Ack=1 Win=64256 Len=0
62	22.932311518	192.168.137.181	10.130.249.231	HTTP	438	GET / HTTP/1.1
63	22.933684819	10.130.249.231	192.168.137.181	TCP	54	85 → 51012 [ACK] Seq=1 Ack=385 Win=262144 Len=0
64	22.940514380	10.130.249.231	192.168.137.181	TCP	590	85 → 51012 [ACK] Seq=1 Ack=385 Win=262144 Len=536 [TCP segment of a reassembled ...]
65	22.940663292	192.168.137.181	10.130.249.231	TCP	54	51012 → 85 [ACK] Seq=385 Ack=537 Win=64128 Len=0
66	22.940778122	10.130.249.231	192.168.137.181	TCP	590	85 → 51012 [ACK] Seq=537 Ack=385 Win=262144 Len=536 [TCP segment of a reassembled ...]
67	22.940799700	10.130.249.231	192.168.137.181	HTTP	214	HTTP/1.1 200 OK (text/html)
68	22.940818280	192.168.137.181	10.130.249.231	TCP	54	51012 → 85 [ACK] Seq=385 Ack=1073 Win=64128 Len=0
69	22.940891662	192.168.137.181	10.130.249.231	TCP	54	51012 → 85 [ACK] Seq=385 Ack=1233 Win=64000 Len=0
70	23.044331139	192.168.137.181	10.130.249.231	HTTP	399	GET /icons/blank.gif HTTP/1.1
71	23.044506817	192.168.137.181	10.130.249.231	TCP	54	51012 → 85 [FIN, ACK] Seq=730 Ack=1233 Win=64128 Len=0
72	23.045866385	10.130.249.231	192.168.137.181	TCP	54	85 → 51012 [ACK] Seq=1233 Ack=730 Win=262400 Len=0
73	23.045902890	10.130.249.231	192.168.137.181	TCP	54	85 → 51012 [ACK] Seq=1233 Ack=731 Win=262400 Len=0
74	23.047165634	10.130.249.231	192.168.137.181	HTTP	510	HTTP/1.1 200 OK (GIF89a)
75	23.047219146	192.168.137.181	10.130.249.231	TCP	54	51012 → 85 [RST] Seq=731 Win=0 Len=0
76	23.047288111	10.130.249.231	192.168.137.181	TCP	54	85 → 51012 [FIN, ACK] Seq=1689 Ack=731 Win=262400 Len=0
77	23.047325792	192.168.137.181	10.130.249.231	TCP	54	51012 → 85 [RST] Seq=731 Win=0 Len=0
78	23.047384413	10.130.249.231	192.168.137.181	TCP	66	[TCP Dup ACK 73#1] 85 → 51012 [ACK] Seq=1690 Ack=731 Win=262400 Len=0 SLE=385 SR...
79	23.047420706	192.168.137.181	10.130.249.231	TCP	54	51012 → 85 [RST] Seq=731 Win=0 Len=0
80	23.047482380	10.130.249.231	192.168.137.181	TCP	54	[TCP Dup ACK 73#2] 85 → 51012 [ACK] Seq=1690 Ack=731 Win=262400 Len=0
81	23.047518896	192.168.137.181	10.130.249.231	TCP	54	51012 → 85 [RST] Seq=731 Win=0 Len=0
82	23.051448760	192.168.137.181	10.130.249.231	TCP	74	51014 → 85 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3167702186 TSe...
83	23.053162426	10.130.249.231	192.168.137.181	TCP	66	85 → 51014 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
84	23.053221343	192.168.137.181	10.130.249.231	TCP	54	51014 → 85 [RST] Seq=1 Win=0 Len=0
87	25.855530926	192.168.137.181	10.130.249.231	TCP	74	51016 → 85 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3167705907 TSe...
88	26.873089578	10.130.249.231	192.168.137.181	TCP	66	85 → 51016 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
89	26.873144753	192.168.137.181	10.130.249.231	TCP	54	51016 → 85 [ACK] Seq=1 Ack=1 Win=64256 Len=0

Internet Protocol Version 4, Src: 192.168.137.181, Dst: 10.130.249.231

Transmission Control Protocol, Src Port: 51012, Dst Port: 85, Seq: 0, Len: 0

Source Port: 51012

Destination Port: 85

Stream index: 0

0000	c2 23 43 d7 66 4f d0 df	9a da 03 74 08 00 45 00	-#C f0..
0010	00 3c 06 a4 00 00 00 06	e5 50 c0 a8 89 b5 0a 82	<- 0 0
0020	f9 e7 c7 44 00 55 3e df	ce 89 00 00 00 00 a0 02	..D U>
0030	fa f0 20 5d 00 00 02 04	05 b4 04 02 08 0a bc cf	..]
0040	4c 18 00 00 00 00 01 03	03 07	L

交互过程的具体分析

TCP 三次握手——建立连接

首先分析本地和服务器建立连接的过程，即TCP的三次握手过程。下面的一个图片简要地讲述了客户端和服务器通过TCP协议进行连接的过程。

第一次握手

用户端向 Web 服务器端发送第一次握手请求。

此处可以看到用户端向服务器端发送的序列号是：1054854793，由用户端产生的初始序列号作为 SYN 被置为1（Flags=0x002），发送到服务器端。

ip.addr == 10.130.249.231 and (tcp.port in {85} or udp.port in {85})						
No.	Time	Source	Destination	Protocol	Length	Info
59	22.930261435	192.168.137.181	10.130.249.231	TCP	74	51012 → 85 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3167702040 TSe...
60	22.931842833	10.130.249.231	192.168.137.181	TCP	66	85 → 51012 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
61	22.931922787	192.168.137.181	10.130.249.231	TCP	54	51012 → 85 [ACK] Seq=1 Ack=1 Win=64256 Len=0
62	22.932311518	192.168.137.181	10.130.249.231	HTTP	438	GET / HTTP/1.1
63	22.933684819	10.130.249.231	192.168.137.181	TCP	54	85 → 51012 [ACK] Seq=1 Ack=385 Win=262144 Len=0
64	22.940514380	10.130.249.231	192.168.137.181	TCP	590	85 → 51012 [ACK] Seq=1 Ack=385 Win=262144 Len=536 [TCP segment of a reassembled ...]
65	22.940663292	192.168.137.181	10.130.249.231	TCP	54	51012 → 85 [ACK] Seq=385 Ack=537 Win=64128 Len=0
66	22.940778122	10.130.249.231	192.168.137.181	TCP	590	85 → 51012 [ACK] Seq=537 Ack=385 Win=262144 Len=536 [TCP segment of a reassembled ...]
67	22.940799700	10.130.249.231	192.168.137.181	HTTP	214	HTTP/1.1 200 OK (text/html)
68	22.940818280	192.168.137.181	10.130.249.231	TCP	54	51012 → 85 [ACK] Seq=385 Ack=1073 Win=64128 Len=0
69	22.940891662	192.168.137.181	10.130.249.231	TCP	54	51012 → 85 [ACK] Seq=385 Ack=1233 Win=64000 Len=0

Frame 59: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface wlan0, id 0

Ethernet II, Src: LiteonTe...da:03:74 (d0:df:9a:da:03:74), Dst: c2:23:43:d7:66:4f (c2:23:43:d7:66:4f)

Internet Protocol Version 4, Src: 192.168.137.181, Dst: 10.130.249.231

Transmission Control Protocol, Src Port: 51012, Dst Port: 85, Seq: 0, Len: 0

Source Port: 51012

Destination Port: 85

Stream index: 0

[TCP Segment Len: 0]

Sequence Number: 0 (relative sequence number)

Sequence Number (raw): 1054854793

[Next Sequence Number: 1 (relative sequence number)]

Acknowledgment Number: 0

Acknowledgment number (raw): 0

1010 ... = Header Length: 40 bytes (10)

Flags: 0x002 (SYN)

Window: 64240

[Calculated window size: 64240]

Checksum: 0x205d [unverified]

[Checksum Status: Unverified]

Urgent Pointer: 0

Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale

Timestamps

0000	c2 23 43 d7 66 4f d0 df	9a da 03 74 08 00 45 00	-#C f0..
0010	00 3c 06 a4 00 00 00 06	e5 50 c0 a8 89 b5 0a 82	<- 0 0
0020	f9 e7 c7 44 00 55 3e df	ce 89 00 00 00 00 a0 02	..D U>
0030	fa f0 20 5d 00 00 02 04	05 b4 04 02 08 0a bc cf	..]
0040	4c 18 00 00 00 00 01 03	03 07	L

用户端 51012 端口向服务器 85 端口发送 TCP 连接请求，进行第一次握手。

检查数据包的具体内容，发现：

Source Port: 51012 Destination Port: 85，表示该报文的源端口和目的端口。校验和状态为：Unverified，未校验。

除此之外还能看到时间戳 Timestamp 等相关信息。

```
Transmission Control Protocol, Src Port: 51012, Dst Port: 85, Seq: 0, Len: 0
  Source Port: 51012
  Destination Port: 85
  [Stream index: 9]
  [TCP Segment Len: 0]
  Sequence Number: 0 (relative sequence number)
  Sequence Number (raw): 1054854793
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 0
  Acknowledgment number (raw): 0
  1010 .... = Header Length: 40 bytes (10)
  ▶ Flags: 0x002 (SYN)
  Window: 64240
  [Calculated window size: 64240]
  Checksum: 0x205d [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  ▶ Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
    ▶ TCP Option - Maximum segment size: 1460 bytes
    ▶ TCP Option - SACK permitted
    ▶ TCP Option - Timestamps: TSval 3167702040, TSecr 0
      Kind: Time Stamp Option (8)
      Length: 10
      Timestamp value: 3167702040
      Timestamp echo reply: 0
    ▶ TCP Option - No-Operation (NOP)
    ▶ TCP Option - Window scale: 7 (multiply by 128)
  ▶ [Timestamps]
    [Time since first frame in this TCP stream: 0.000000000 seconds]
    [Time since previous frame in this TCP stream: 0.000000000 seconds]
```

第二次握手

服务器接收到用户的TCP连接请求的第一次握手后，进行相应，即第二次握手。

可以看到 Source Address: 10.130.249.231（服务器的 IP 地址） Destination Address: 192.168.137.181（用户的 IP 地址）

No.	Time	Source	Destination	Protocol	Length	Info
59	22.930261435	192.168.137.181	10.130.249.231	TCP	74	51012 → 85 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3167702040 TSe...
60	22.931842833	10.130.249.231	192.168.137.181	TCP	66	85 → 51012 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
61	22.931922787	192.168.137.181	10.130.249.231	TCP	54	51012 → 85 [ACK] Seq=1 Ack=1 Win=64256 Len=0

Fragment Offset: 0
Time to Live: 127
Protocol: TCP (6)
Header Checksum: 0xf417 [validation disabled]
[Header checksum status: Unverified]
Source Address: 10.130.249.231
Destination Address: 192.168.137.181

```
Transmission Control Protocol, Src Port: 85, Dst Port: 51012, Seq: 0, Ack: 1, Len: 0
  Source Port: 85
  Destination Port: 51012
  [Stream index: 9]
  [TCP Segment Len: 0]
  Sequence Number: 0 (relative sequence number)
  Sequence Number (raw): 374943798
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 1054854794
  1000 .... = Header Length: 32 bytes (8)
  ▶ Flags: 0x012 (SYN, ACK)
  Window: 8192
  [Calculated window size: 8192]
  Checksum: 0xe4a5 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  ▶ Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
    ▶ TCP Option - Maximum segment size: 1460 bytes
    ▶ TCP Option - No-Operation (NOP)
    ▶ TCP Option - Window scale: 8 (multiply by 256)
    ▶ TCP Option - No-Operation (NOP)
    ▶ TCP Option - No-Operation (NOP)
    ▶ TCP Option - SACK permitted
  ▶ [SEQ/ACK analysis]
  ▶ [Timestamps]
    [Time since first frame in this TCP stream: 0.001581398 seconds]
    [Time since previous frame in this TCP stream: 0.001581398 seconds]
```

此外，发现服务器将用户端发送的序列号 1054854793 加一，变成 1054854794 存在了 Acknowledgment number (raw) 中。同时服务器也初始化了自己的序列号，为 374943798，存在了 sequence number 中，作为 ACK 发送给用户端。

Flags 位被置为 0x012，其中包含了 SYN 和 ACK，表示服务器确认序列号有效并允许与该用户建立一个新的 TCP 连接。

第三次握手

connect 返回一个连接建立，即 established 状态发送给服务器。对应地，服务器也建立一个新的文件描述符和客户端通信。客户端将服务器端发送的序列号加1，作为 ACK 发送给Server。这里就可以看到是 374943799，相比上一次服务器发送过来的序列号要多1。

No.	Time	Source	Destination	Protocol	Length	Info
59	22.930261435	192.168.137.181	10.130.249.231	TCP	74	51012 → 85 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3167702040 TSe...
60	22.931842833	10.130.249.231	192.168.137.181	TCP	66	85 → 51012 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
61	22.931922787	192.168.137.181	10.130.249.231	TCP	54	51012 → 85 [ACK] Seq=1 Ack=1 Win=64256 Len=0

0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 40
Identification: 0x06a5 (1701)
▸ Flags: 0x40, Don't fragment
Fragment Offset: 0
Time to Live: 64
Protocol: TCP (6)
Header Checksum: 0xe563 [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.137.181
Destination Address: 10.130.249.231

▾ Transmission Control Protocol, Src Port: 51012, Dst Port: 85, Seq: 1, Ack: 1, Len: 0
Source Port: 51012
Destination Port: 85
[Stream index: 9]
[TCP Segment Len: 0]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 1954854794
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 374943799

0101 = Header Length: 20 bytes (5)
▸ Flags: 0x010 (ACK)
Window: 502
[Calculated window size: 64256]
[Window size scaling factor: 128]
Checksum: 0x4383 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
▸ [SEQ/ACK analysis]
▾ [Timestamps]
[Time since first frame in this TCP stream: 0.001661352 seconds]
[Time since previous frame in this TCP stream: 0.000079954 seconds]

此数据包，Flags被置为 0x010，包含的内容是 ACK，符合 TCP 第三次握手的标志位规定。

源地址为：192.168.137.181 目的地址为：10.130.249.231 和第一次握手时一样。

经过三次握手，用户端和服务器端的 TCP 连接建立了可靠的全双工连接，开始传送数据，接下来将分析数据的传输过程。

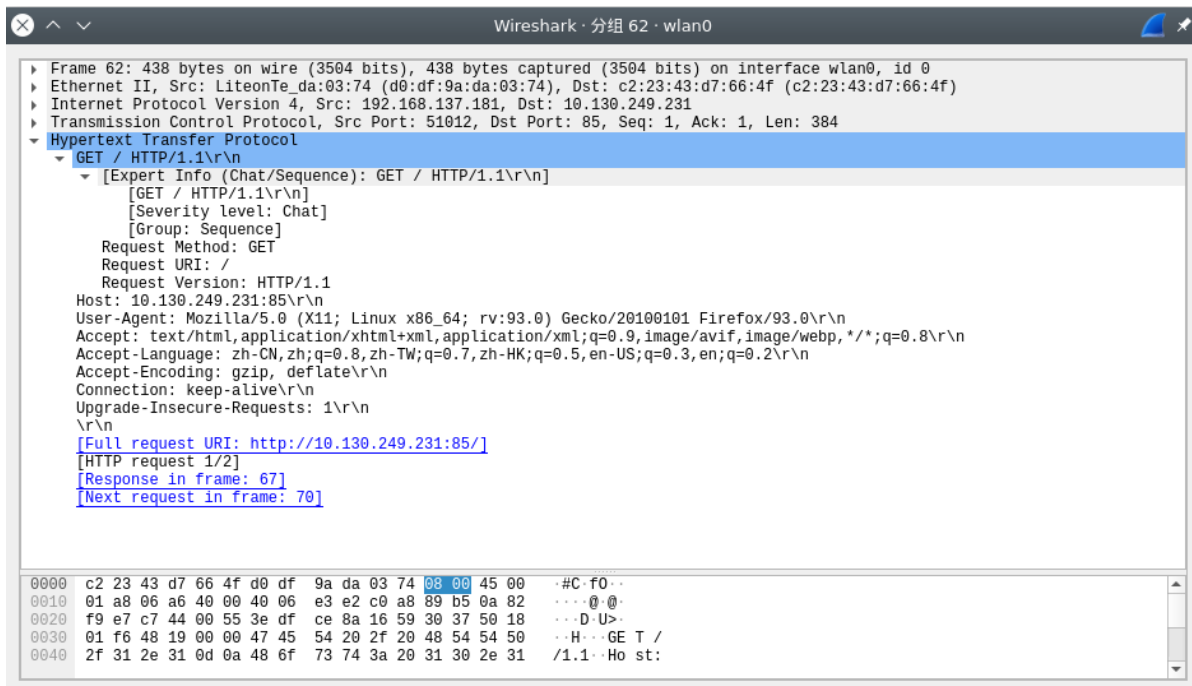
获取信息

获取文字信息

用户向服务器首先发送一个Get请求，其中，可以看到，报文中“GET/HTTP/1.1”的部分，通过明文方式显示出传输协议相关的信息，这是由于http的get请求就是明文传输，使用的协议是版本http1.1。

61	22.931922787	192.168.137.181	10.130.249.231	TCP	54	51012 → 85 [ACK] Seq=1 Ack=1 Win=642
62	22.932311518	192.168.137.181	10.130.249.231	HTTP	438	GET / HTTP/1.1
63	22.933684819	10.130.249.231	192.168.137.181	TCP	54	85 → 51012 [ACK] Seq=1 Ack=385 Win=2

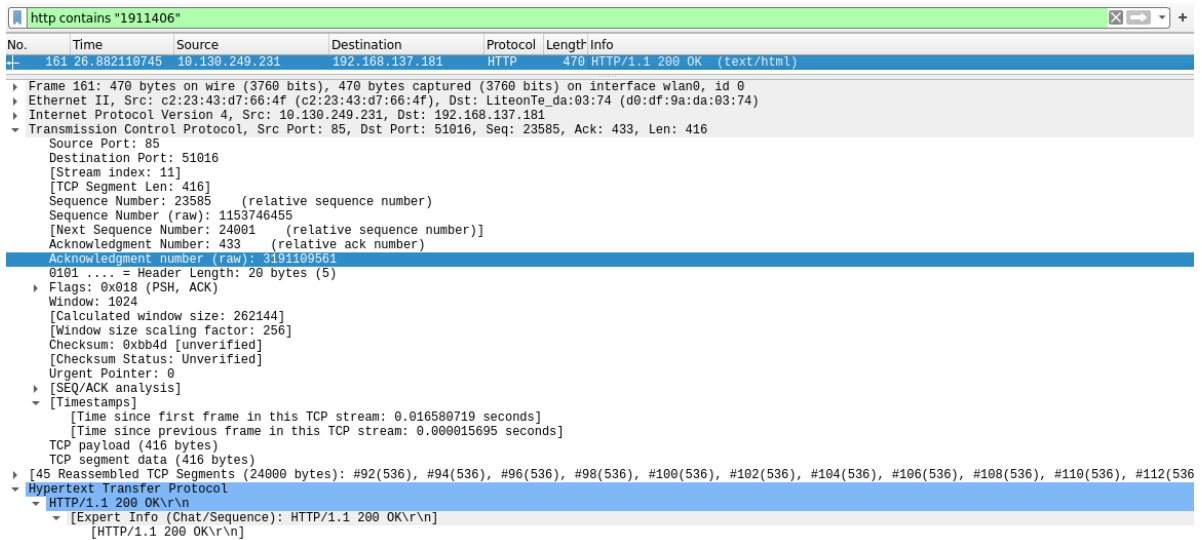
在请求头中也可以看到用户端的相关信息。使用的操作系统版本：Linux，x86_64 等等，以及使用的浏览器为 Firefox。下面的 accept 相关部分指明了可接受的文件类型，语言，编码等等。



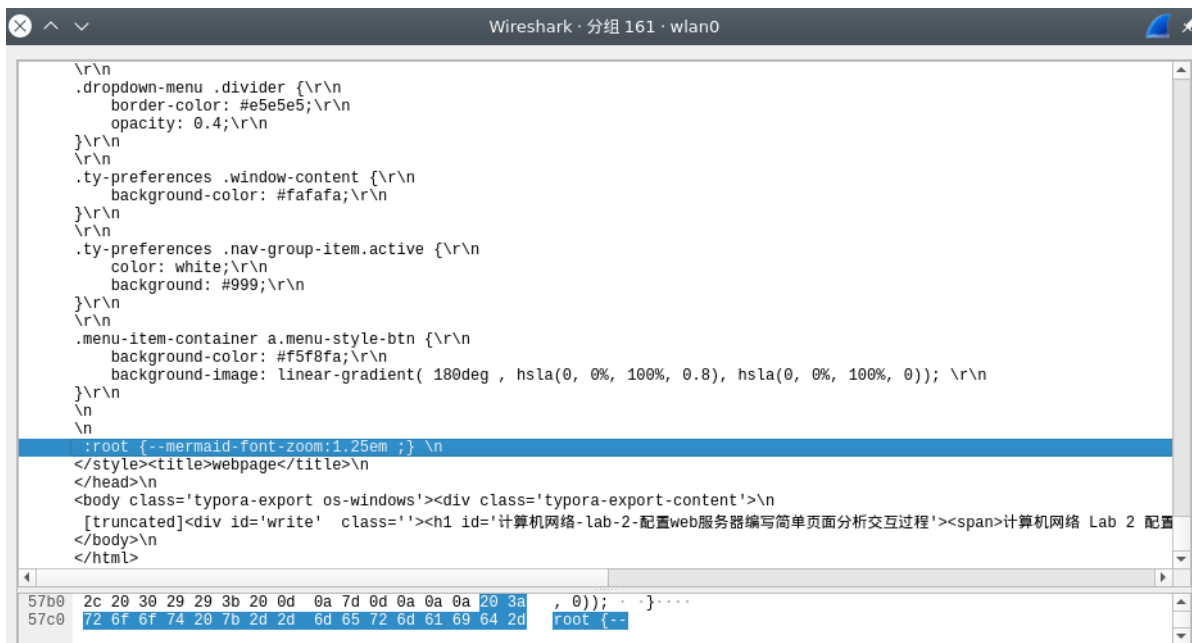
查询发送的文字信息，使用在 filter 处输入网页包含的信息（学号：1911406）

http contains "1911406"

找到协议类型 HTTP，且报文内容包含“1911406”的包。服务器向用户返回状态码 200，表示 OK，请求成功。



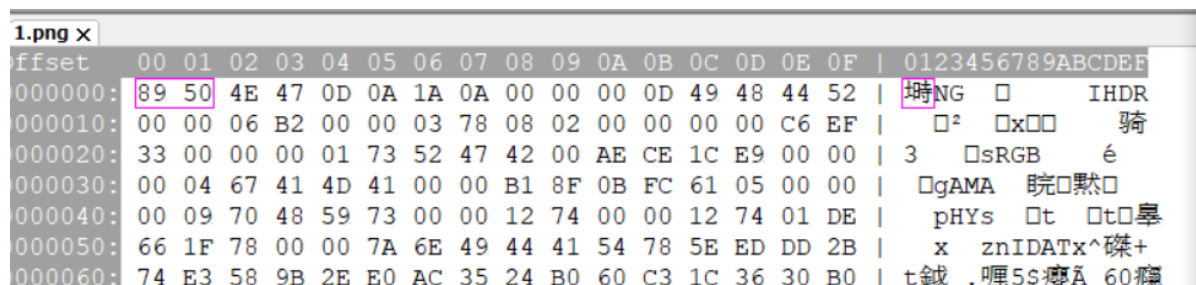
在响应体中，我们就可以看到之前自己写的html文档被完全地传输了过来。



获取图片信息

请求头和响应头都会进行一个标识。

看传输的内容。首先用一个二进制编辑器 wxMEdit 在服务器端打开 1.png，如下，我们只需要记住第一行包含 IHDR，第三行包含 sRGB。



我们可以看出捕获的包中，也包含刚才说的“第一行包含 IHDR，第三行包含 sRGB”，这就表示我们在响应体中接收到了这个图片。

TCP 四次挥手——断开连接

第一次挥手

源 IP 地址为 192.168.137.181，目的 IP 地址为 10.130.249.231。是用户向服务器发送的请求。

ip.addr == 10.130.249.231 and (tcp.port in {85} or udp.port in {85})						
No.	Time	Source	Destination	Protocol	Length	Info
1051	201.730968390	10.130.249.231	192.168.137.181	TCP	54	85 → 51030 [ACK] Seq=1234 Ack=386 Win=262144 Len=0
1050	201.724762201	192.168.137.181	10.130.249.231	TCP	54	51030 → 85 [ACK] Seq=386 Ack=1234 Win=64128 Len=0
1049	201.724651524	192.168.137.181	10.130.249.231	TCP	54	51030 → 85 [FIN, ACK] Seq=385 Ack=1233 Win=64128 Len=0
1048	201.724577372	10.130.249.231	192.168.137.181	TCP	54	85 → 51030 [FIN, ACK] Seq=1233 Ack=385 Win=262144 Len=0
1036	196.723457191	192.168.137.181	10.130.249.231	TCP	54	51030 → 85 [ACK] Seq=385 Ack=1233 Win=64128 Len=0
▶ Frame 1049: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface wlan0, id 0 ▶ Ethernet II, Src: LiteonTe_da:03:74 (d0:df:9a:da:03:74), Dst: c2:23:43:d7:66:4f (c2:23:43:d7:66:4f) ▶ Internet Protocol Version 4, Src: 192.168.137.181, Dst: 10.130.249.231 ▶ Transmission Control Protocol, Src Port: 51030, Dst Port: 85, Seq: 385, Ack: 1233, Len: 0						
Source Port: 51030 Destination Port: 85 [Stream index: 47] [TCP Segment Len: 0] Sequence Number: 385 (relative sequence number) Sequence Number (raw): 2196706106 [Next Sequence Number: 386 (relative sequence number)] Acknowledgment Number: 1233 (relative ack number) Acknowledgment number (raw): 3227568484 0101 = Header Length: 20 bytes (5) Flags: 0x011 (FIN, ACK) 000. = Reserved: Not set ...0 = Nonce: Not set0 = Congestion Window Reduced (CWR): Not set0 = ECN-Echo: Not set0 = Urgent: Not set1 = Acknowledgment: Set0 = Push: Not set0 = Reset: Not set0 = Syn: Not set1 = Fin: Set [TCP Flags:A...] Window: 501 [Calculated window size: 64128] [Window size scaling factor: 128] Checksum: 0x777c [unverified] [Checksum Status: Unverified] Urgent Pointer: 0 [Timestamps] [Time since first frame in this TCP stream: 5.015932207 seconds]						

用户和服务通信完毕过后，用户调用close，开始四次挥手，用户端发送一个 FIN，用来关闭用户到服务器的数据传输。注意到此报文序号为 2196706106。

第二次挥手

源 IP 地址为 10.130.249.231，目的 IP 地址为 192.168.137.181。是服务器向用户发送的请求。

ip.addr == 10.130.249.231 and (tcp.port in {85} or udp.port in {85})						
No.	Time	Source	Destination	Protocol	Length	Info
1051	201.730968390	10.130.249.231	192.168.137.181	TCP	54	85 → 51030 [ACK] Seq=1234 Ack=386 Win=262144 Len=0
1050	201.724762201	192.168.137.181	10.130.249.231	TCP	54	51030 → 85 [ACK] Seq=386 Ack=1234 Win=64128 Len=0
1049	201.724651524	192.168.137.181	10.130.249.231	TCP	54	51030 → 85 [FIN, ACK] Seq=385 Ack=1233 Win=64128 Len=0
1048	201.724577372	10.130.249.231	192.168.137.181	TCP	54	85 → 51030 [FIN, ACK] Seq=1233 Ack=385 Win=262144 Len=0
1036	196.723457191	192.168.137.181	10.130.249.231	TCP	54	51030 → 85 [ACK] Seq=385 Ack=1233 Win=64128 Len=0
▶ Frame 1051: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface wlan0, id 0 ▶ Ethernet II, Src: LiteonTe_da:03:74 (d0:df:9a:da:03:74), Dst: LiteonTe_da:03:74 (d0:df:9a:da:03:74) ▶ Internet Protocol Version 4, Src: 10.130.249.231, Dst: 192.168.137.181 ▶ Transmission Control Protocol, Src Port: 85, Dst Port: 51030, Seq: 1234, Ack: 386, Len: 0						
Source Port: 85 Destination Port: 51030 [Stream index: 47] [TCP Segment Len: 0] Sequence Number: 1234 (relative sequence number) Sequence Number (raw): 3227568485 [Next Sequence Number: 1234 (relative sequence number)] Acknowledgment Number: 386 (relative ack number) Acknowledgment number (raw): 2196706107 0101 = Header Length: 20 bytes (5) Flags: 0x010 (ACK) 000. = Reserved: Not set ...0 = Nonce: Not set0 = Congestion Window Reduced (CWR): Not set0 = ECN-Echo: Not set0 = Urgent: Not set1 = Acknowledgment: Set0 = Push: Not set0 = Reset: Not set0 = Syn: Not set0 = Fin: Not set [TCP Flags:A...] Window: 1024 [Calculated window size: 262144] [Window size scaling factor: 256] Checksum: 0x7570 [unverified] [Checksum Status: Unverified] Urgent Pointer: 0 [SEQ/ACK analysis] [Timestamps]						

第二次挥手，服务器收到用户发来的 FIN 之后，发送的序号 2196706107 比第一次挥手的正好多一，作为 ACK 确认报文的序号，服务器进入CLOSE_WAIT状态。

第三次挥手

源 IP 地址为 10.130.249.231，目的 IP 地址为 192.168.137.181。是服务器向用户发送的请求。

ip.addr == 10.130.249.231 and (tcp.port in {85} or udp.port in {85})						
No.	Time	Source	Destination	Protocol	Length	Info
1051	201.730968390	10.130.249.231	192.168.137.181	TCP	54	85 → 51030 [ACK] Seq=1234 Ack=386 Win=262144 Len=0
1050	201.724762201	192.168.137.181	10.130.249.231	TCP	54	51030 → 85 [ACK] Seq=386 Ack=1234 Win=64128 Len=0
1049	201.724651524	192.168.137.181	10.130.249.231	TCP	54	51030 → 85 [FIN, ACK] Seq=385 Ack=1233 Win=64128 Len=0
1048	201.724577372	10.130.249.231	192.168.137.181	TCP	54	85 → 51030 [FIN, ACK] Seq=1233 Ack=385 Win=262144 Len=0
1036	196.723457191	192.168.137.181	10.130.249.231	TCP	54	51030 → 85 [ACK] Seq=385 Ack=1233 Win=64128 Len=0
▶ Frame 1048: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface wlan0, id 0 ▶ Ethernet II, Src: c2:23:43:d7:66:4f (c2:23:43:d7:66:4f), Dst: LiteonTe_da:03:74 (d0:df:9a:da:03:74) ▶ Internet Protocol Version 4, Src: 10.130.249.231, Dst: 192.168.137.181 ▶ Transmission Control Protocol, Src Port: 85, Dst Port: 51030, Seq: 1233, Ack: 385, Len: 0 Source Port: 85 Destination Port: 51030 [Stream index: 47] [TCP Segment Len: 0] Sequence Number: 1233 (relative sequence number) Sequence Number (raw): 3227568484 [Next Sequence Number: 1234 (relative sequence number)] Acknowledgment Number: 385 (relative ack number) Acknowledgment number (raw): 2196706106 0101 = Header Length: 20 bytes (5) Flags: 0x011 (FIN, 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 ...0 = Reset: Not set ...0 = Syn: Not set ...1 = Fin: Set [TCP Flags:A...] Window: 1024 [Calculated window size: 262144] [Window size scaling factor: 256] Checksum: 0x7571 [unverified] [Checksum Status: Unverified] Urgent Pointer: 0 [Timestamps] [Time since first frame in this TCP stream: 5.015858055 seconds]						

第三次挥手，服务器向用户发送 FIN 报文并且服务器关闭连接。

第四次挥手

源 IP 地址为 192.168.137.181，目的 IP 地址为 10.130.249.231。是用户向服务器发送的请求。

ip.addr == 10.130.249.231 and (tcp.port in {85} or udp.port in {85})						
No.	Time	Source	Destination	Protocol	Length	Info
1051	201.730968390	10.130.249.231	192.168.137.181	TCP	54	85 → 51030 [ACK] Seq=1234 Ack=386 Win=262144 Len=0
1050	201.724762201	192.168.137.181	10.130.249.231	TCP	54	51030 → 85 [ACK] Seq=386 Ack=1234 Win=64128 Len=0
1049	201.724651524	192.168.137.181	10.130.249.231	TCP	54	51030 → 85 [FIN, ACK] Seq=385 Ack=1233 Win=64128 Len=0
1048	201.724577372	10.130.249.231	192.168.137.181	TCP	54	85 → 51030 [FIN, ACK] Seq=1233 Ack=385 Win=262144 Len=0
1036	196.723457191	192.168.137.181	10.130.249.231	TCP	54	51030 → 85 [ACK] Seq=385 Ack=1233 Win=64128 Len=0
▶ Frame 1050: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface wlan0, id 0 ▶ Ethernet II, Src: LiteonTe_da:03:74 (d0:df:9a:da:03:74), Dst: c2:23:43:d7:66:4f (c2:23:43:d7:66:4f) ▶ Internet Protocol Version 4, Src: 192.168.137.181, Dst: 10.130.249.231 ▶ Transmission Control Protocol, Src Port: 51030, Dst Port: 85, Seq: 386, Ack: 1234, Len: 0 Source Port: 51030 Destination Port: 85 [Stream index: 47] [TCP Segment Len: 0] Sequence Number: 386 (relative sequence number) Sequence Number (raw): 2196706107 [Next Sequence Number: 386 (relative sequence number)] Acknowledgment Number: 1234 (relative ack number) Acknowledgment number (raw): 3227568485 0101 = Header Length: 20 bytes (5) 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 ...0 = Reset: Not set ...0 = Syn: Not set ...0 = Fin: Not set [TCP Flags:A...] Window: 501 [Calculated window size: 64128] [Window size scaling factor: 128] Checksum: 0x777b [unverified] [Checksum Status: Unverified] Urgent Pointer: 0 [SEQ/ACK analysis] [Timestamps]						

第四次挥手，用户接收到服务器发送的 FIN 报文并向服务器返回 ACK 报文，并进入定时等待。之后用户端关闭关闭连接。