

## 实验内容

- 1.基本HTTP GET/response交互
- 2.HTTP条件Get/response交互
- 3.检索长文件
- 4.具有嵌入对象的HTML文档
- 5.HTTP认证

## 实验步骤

### 1.基本HTTP GET/response交互

先开浏览器稍等一下（原因后面会解释），然后打开 `wireshark`，点击开始捕获之后，在浏览器输入 URL

<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html>

捕获停止后的界面如下

这是数据包51号--请求消息

No.	Time	Source	Destination	Protocol	Length	Info
51	2020-04-04 21:40:06.468598824	192.168.43.30	128.119.245.12	HTTP	523	GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1
55	2020-04-04 21:40:06.814924955	128.119.245.12	192.168.43.30	HTTP	552	HTTP/1.1 200 OK (text/html)
77	2020-04-04 21:40:07.257508083	192.168.43.30	128.119.245.12	HTTP	461	GET /favicon.ico HTTP/1.1
87	2020-04-04 21:40:08.248782980	128.119.245.12	192.168.43.30	HTTP	550	HTTP/1.1 404 Not Found (text/html)

Frame 51: 523 bytes on wire (4184 bits), 523 bytes captured (4184 bits) on interface 0
Ethernet II, Src: IntelCor-11:2c:59 (9c:da:3e:11:2c:59), Dst: f6:b7:b3:8f:c3:92 (f6:b7:b3:8f:c3:92)
Internet Protocol Version 4, Src: 192.168.43.30, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 36738, Dst Port: 80, Seq: 1, Ack: 1, Len: 457
Hypertext Transfer Protocol
GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n
Host: gaia.cs.umass.edu\r\n
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86 Safari/537.36\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3\r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8\r\n
\r\n
[Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html]
[HTTP request 1/1]
[Response in frame: 55]

绿色部分的 Internet Protocol Version 4 其实就是 IPv4 的缩写，接着的 Transmission Control Protocol 就是 TCP 的全称，而且在这一行还显示了源地址跟目的地址的端口号等信息，最后的 Hypertext Transfer Protocol 就是 HTTP 的全称。

这是数据包55号--http响应消息

No.	Time	Source	Destination	Protocol	Length	Info
51	2020-04-04 21:40:06.468598824	192.168.43.30	128.119.245.12	HTTP	552	GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1
55	2020-04-04 21:40:06.814924955	128.119.245.12	192.168.43.30	HTTP	552	HTTP/1.1 200 OK (text/html)
77	2020-04-04 21:40:07.257508083	192.168.43.30	128.119.245.12	HTTP	461	GET /favicon.ico HTTP/1.1
87	2020-04-04 21:40:08.248782980	128.119.245.12	192.168.43.30	HTTP	550	HTTP/1.1 404 Not Found (text/html)

<p>Frame 55: 552 bytes on wire (4416 bits), 552 bytes captured (4416 bits) on interface 0</p> <p>Ethernet II, Src: f6:b7:b3:8f:c3:92 (f6:b7:b3:8f:c3:92), Dst: IntelCor_11:2c:59 (9c:da:3e:11:2c:59)</p> <p>Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.43.30</p> <p>Transmission Control Protocol, Src Port: 80, Dst Port: 36738, Seq: 1, Ack: 458, Len: 486</p> <p>Hypertext Transfer Protocol</p> <p>HTTP/1.1 200 OK\r\n</p> <p>Date: Sat, 04 Apr 2020 13:40:06 GMT\r\n</p> <p>Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/5.4.16 mod_perl/2.0.11 Perl/v5.16.3\r\n</p> <p>Last-Modified: Sat, 04 Apr 2020 05:59:02 GMT\r\n</p> <p>Etag: "80-5a270bb15be41"\r\n</p> <p>Accept-Ranges: bytes\r\n</p> <p>Content-Length: 128\r\n</p> <p>Keep-Alive: timeout=5, max=100\r\n</p> <p>Connection: Keep-Alive\r\n</p> <p>Content-Type: text/html; charset=UTF-8\r\n</p> <p>\r\n</p> <p>[HTTP response 1/2]</p> <p>[Time since request: 0.346326131 seconds]</p> <p>[Request in frame: 51]</p> <p>[Next request in frame: 77]</p> <p>[Next response in frame: 87]</p> <p>[Request URI: http://gaia.cs.umass.edu/favicon.ico]</p> <p>File Data: 128 bytes</p> <p>Line-based text data: text/html (4 lines)</p>
--

## 回答问题

1.您的浏览器是否运行HTTP版本1.0或1.1?服务器运行什么版本的HTTP?

浏览器跟服务器都是运行的HTTP1.1版本

2.您的浏览器会从服务器接受哪种语言(如果有的话)?

在请求消息图中的 Accept-Language 显示的是 zh-CN，也就是中文啦

3.您的计算机的IP地址是什么?gaia.cs.umass.edu服务器地址呢?

计算机的IP地址: 192.168.43.0  
gaia.cs.umass.edu服务器地址: 128.119.245.12

4.服务器返回到浏览器的状态代码是什么?

200，代表OK

5.服务器上HTML文件的最近一次修改是什么时候?

在响应消息的图中显示的是  
Last-Modified: Sat, 04 Apr 2020 05:59:02 GMT\r\n

6.服务器返回多少字节的内容到您的浏览器?

在响应消息的图中显示的是128字节  
Accept-Ranges: bytes  
Content-Length: 128\r\n

7.通过检查数据包内容窗口中的原始数据,你是否看到有协议头在数据包列表窗口中未显示?如果是,请举一个例子。

TCP

## 2.HTTP条件Get/response交互

先清除浏览器缓存，打开浏览器后，开始捕获，输入上次的URL，成功显示后刷新或者重新进入上次的网址，停止捕获后的界面如下。

No.	Time	Source	Destination	Protocol	Length	Info
462	2020-04-04 22:20:27.684432164	192.168.43.30	128.119.245.12	HTTP	523	GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
467	2020-04-04 22:20:28.009770596	128.119.245.12	192.168.43.30	HTTP	796	HTTP/1.1 200 OK (text/html)
485	2020-04-04 22:20:33.509834962	192.168.43.30	128.119.245.12	HTTP	635	GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
488	2020-04-04 22:20:33.846920105	128.119.245.12	192.168.43.30	HTTP	306	HTTP/1.1 304 Not Modified

回答问题

8.检查第一个从您浏览器到服务器的HTTP GET请求的内容。您在HTTP GET中看到了“IF-MODIFIED-SINCE”行吗？

第一个GET请求中并没有看到

```
Frame 462: 523 bytes on wire (4184 bits), 523 bytes captured (4184 bits) on interface 0
Ethernet II, Src: IntelCor_11:2c:59 (9c:da:3e:11:2c:59), Dst: f6:b7:b3:8f:c3:92 (f6:b7:b3:8f:c3:92)
Internet Protocol Version 4, Src: 192.168.43.30, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 37280, Dst Port: 80, Seq: 1, Ack: 1, Len: 457
Hypertext Transfer Protocol
  GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
  Host: gaia.cs.umass.edu\r\n
  Connection: keep-alive\r\n
  Upgrade-Insecure-Requests: 1\r\n
  User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86 Safari/537.36\r\n
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3\r\n
  Accept-Encoding: gzip, deflate\r\n
  Accept-Language: zh-CN,zh;q=0.9,en;q=0.8\r\n
  \r\n
  [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
  [HTTP request 1/1]
  [Response in frame: 467]
```

9.检查服务器响应的内容,服务器是否显式返回文件的内容?你是怎么知道的?

是的。在 Line-based text data 里面可以看到

```
File data: 314 bytes
Line-based text data: text/html (10 lines)
<html>\n
\n
Congratulations again! Now you've downloaded the file lab2-2.html. <br>\n
This file's last modification date will not change. <p>\n
Thus if you download this multiple times on your browser, a complete copy <br>\n
will only be sent once by the server due to the inclusion of the IN-MODIFIED-SINCE<br>\n
field in your browser's HTTP GET request to the server.\n
\n
</html>\n
```

10.现在,检查第二个HTTP GET请求的内容。您在HTTP GET中看到了“IF-MODIFIED-SINCE:”行吗?如果是,“IF-MODIFIED-SINCE:”头后面包含哪些信息?

If-Modified-Since: Sat, 04 Apr 2020 05:59:02 GMT\r\n

包含的是上次服务器更新的时间。（因为gaia.cs.umass.edu服务器将这个特定文件的最后修改时间设置为当前时间,并且每分钟执行一次）

```

Frame 485: 635 bytes on wire (5080 bits), 635 bytes captured (5080 bits) on interface 0
Ethernet II, Src: IntelCor_11:2c:59 (9c:da:3e:11:2c:59), Dst: f6:b7:b3:8f:c3:92 (f6:b7:b3:8f:c3:92)
Internet Protocol Version 4, Src: 192.168.43.30, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 37282, Dst Port: 80, Seq: 1, Ack: 1, Len: 569
Hypertext Transfer Protocol
  GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    Connection: keep-alive\r\n
    Cache-Control: max-age=0\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: zh-CN,zh;q=0.9,en;q=0.8\r\n
    If-None-Match: "173-5a270bb15aea1"\r\n
    If-Modified-Since: Sat, 04 Apr 2020 05:59:02 GMT\r\n
  \r\n
[Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
[HTTP request 1/1]
[Response in frame: 488]

```

11. 针对第二个HTTP GET,从服务器响应的HTTP状态码和短语是什么?服务器是否明确地返回文件的内容?请解释。

状态码短语: 306 HTTP/1.1 304 Not Modified  
 服务并没有明确返回文件内容

No.	Time	Source	Destination	Protocol	Length	Info
462	2020-04-04 22:20:27.684432164	192.168.43.30	128.119.245.12	HTTP	523	GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
467	2020-04-04 22:20:28.009770596	128.119.245.12	192.168.43.30	HTTP	796	HTTP/1.1 200 OK (text/html)
485	2020-04-04 22:20:33.509834962	192.168.43.30	128.119.245.12	HTTP	635	GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
488	2020-04-04 22:20:33.846926105	128.119.245.12	192.168.43.30	HTTP	306	HTTP/1.1 304 Not Modified

```

Frame 488: 306 bytes on wire (2448 bits), 306 bytes captured (2448 bits) on interface 0
Ethernet II, Src: f6:b7:b3:8f:c3:92 (f6:b7:b3:8f:c3:92), Dst: IntelCor_11:2c:59 (9c:da:3e:11:2c:59)
Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.43.30
Transmission Control Protocol, Src Port: 80, Dst Port: 37282, Seq: 1, Ack: 570, Len: 240
Hypertext Transfer Protocol
  HTTP/1.1 304 Not Modified\r\n
    Date: Sat, 04 Apr 2020 14:20:33 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/5.4.16 mod_perl/2.0.11 Perl/v5.16.3\r\n
    Connection: Keep-Alive\r\n
    Keep-Alive: timeout=5, max=100\r\n
    ETag: "173-5a270bb15aea1"\r\n
  \r\n
[HTTP response 1/1]
[Time since request: 0.337085143 seconds]
[Request in frame: 485]
[Request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]

```

### 3.检索长文件

这次的URL

<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html>

捕获后的截图如下

No.	Time	Source	Destination	Protocol	Length	Info
617	2020-04-04 22:34:39.002087647	192.168.43.30	128.119.245.12	HTTP	523	GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
628	2020-04-04 22:34:39.436326166	128.119.245.12	192.168.43.30	HTTP	4927	HTTP/1.1 200 OK (text/html)

```

Frame 628: 4927 bytes on wire (39416 bits), 4927 bytes captured (39416 bits) on interface 0
Ethernet II, Src: f6:b7:b3:8f:c3:92 (f6:b7:b3:8f:c3:92), Dst: IntelCor_11:2c:59 (9c:da:3e:11:2c:59)
Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.43.30
Transmission Control Protocol, Src Port: 80, Dst Port: 37584, Seq: 1, Ack: 458, Len: 4861
Hypertext Transfer Protocol
  Line-based text data: text/html (98 lines)

```

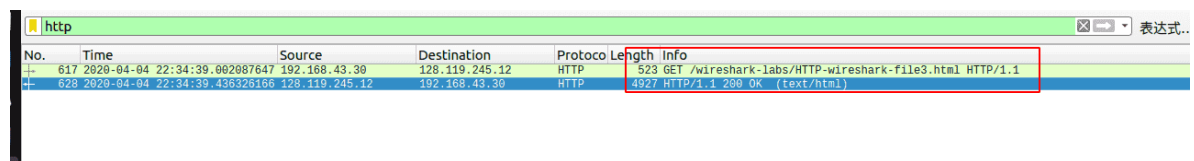
不知道为什么我检索的长文件，并没有显示TCP数据包之间分段的信息，在TCP的头消息中包的数量为1.....，本来按照书上的说法，这次的文件比较大，应该是有多个TCP数据包的，后来重复几次之后，发现还是1个TCP数据包

在我们的例子中,HTML文件相当长,4500字节太大,一个TCP数据包不能容纳。因此,单个HTTP响应消息由TCP分成几个部分,每个部分包含在单独的TCP报文段中。在Wireshark的最新版本中,Wireshark将每个TCP报文段指定为独立的数据包,并且单个HTTP响应在多个TCP数据包之间分段的事实由Wireshark显示的Info列的“重组PDU的TCP段”指示。

## 回答问题

12.您的浏览器发送多少HTTP GET请求消息?哪个数据包包含了美国权利法案的消息?

发送了1个HTTP GET请求消息，数据包620包含了美国权利法案的消息



No.	Time	Source	Destination	Protocol	Length	Info
617	2020-04-04 22:34:39.002887647	192.168.43.30	128.119.245.12	HTTP	523	GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
620	2020-04-04 22:34:39.436326166	128.119.245.12	192.168.43.30	HTTP	4927	HTTP/1.1 200 OK (text/html)

13.哪个数据包包含响应HTTP GET请求的状态码和短语?

620

14.响应中的状态码和短语是什么?

200 OK

15.需要多少包含数据的TCP段来执行单个HTTP响应和权利法案文本?

我这里包含了1个

## 4.具有嵌入对象的HTML文档

这次的URL

<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html>

最终的实验结果

No.	Time	Source	Destination	Protocol	Length	Info
192	2020-04-04 22:55:01.794970335	192.168.43.30	128.119.245.12	HTTP	523	GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
196	2020-04-04 22:55:02.098617898	128.119.245.12	192.168.43.30	HTTP	1139	HTTP/1.1 200 OK (text/html)
198	2020-04-04 22:55:02.170997937	192.168.43.30	128.119.245.12	HTTP	461	GET /pearson.png HTTP/1.1
206	2020-04-04 22:55:03.630666584	128.119.245.12	192.168.43.30	HTTP	941	HTTP/1.1 200 OK (PNG)
214	2020-04-04 22:55:05.474372539	192.168.43.30	128.119.245.12	HTTP	475	GET /~kurose/cover_5th_ed.jpg HTTP/1.1

Frame 206: 941 bytes on wire (7528 bits), 941 bytes captured (7528 bits) on interface 0 Ethernet II, Src: f6:b7:b3:8f:c3:92 (f6:b7:b3:8f:c3:92), Dst: IntelCor_11:2c:59 (9c:da:3e:11:2c:59) Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.43.30 Transmission Control Protocol, Src Port: 80, Dst Port: 37782, Seq: 3810, Ack: 853, Len: 875 [2 Reassembled TCP Segments (3611 bytes): #206(2736), #206(875)] Hypertext Transfer Protocol Portable Network Graphics PNG Signature: 89504e470d0a1a0a Image Header (IHDR) Palette (PLTE) Image data chunk (IDAT) Image Trailer (IEND)
--

## 回答问题

16.您的浏览器发送了几个HTTP GET请求消息?这些GET请求发送到哪个IP地址?

3个请求。发送到地址128.119.245.12。

17.浏览器从两个网站串行还是并行下载了两张图片?请说明。

并行。（官方是这么说的）

No.	Time	Source	Destination	Protocol	Length	Info
192	2020-04-04 22:55:01.794970335	192.168.43.30	128.119.245.12	HTTP	523	GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
196	2020-04-04 22:55:02.098617898	128.119.245.12	192.168.43.30	HTTP	1139	HTTP/1.1 200 OK (text/html)
198	2020-04-04 22:55:02.170997937	192.168.43.30	128.119.245.12	HTTP	461	GET /pearson.png HTTP/1.1
206	2020-04-04 22:55:03.630666584	128.119.245.12	192.168.43.30	HTTP	941	HTTP/1.1 200 OK (PNG)
214	2020-04-04 22:55:05.474372539	192.168.43.30	128.119.245.12	HTTP	475	GET /~kurose/cover_5th_ed.jpg HTTP/1.1

## 5.HTTP认证

这次的URL

[http://gaia.cs.umass.edu/wireshark-labs/protected\\_pages/HTTP-wireshark-file5.html](http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html)

捕获截图



# HTTP请求消息

## ❖ HTTP协议有两类消息

- 请求消息(request)
- 响应消息(response)

## ❖ 请求消息

- ASCII: 人直接可读

request line  
(GET, POST, HEAD commands)

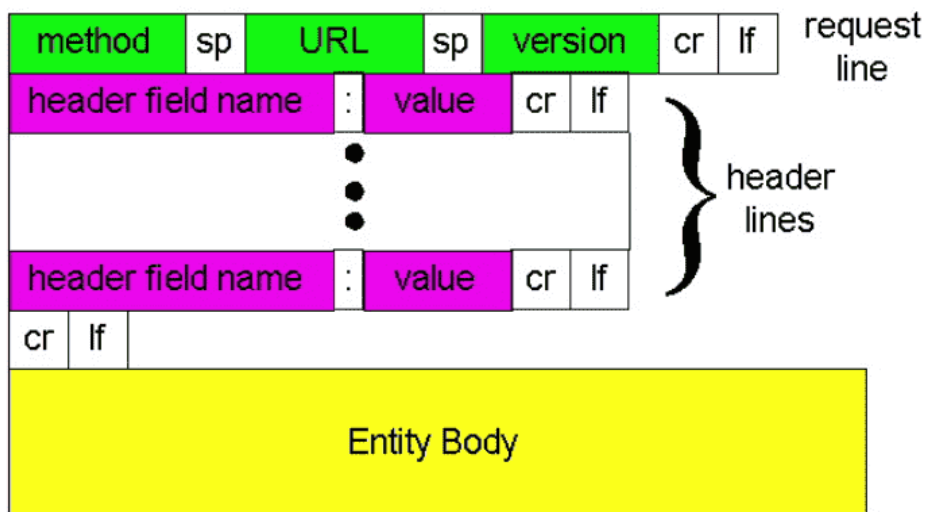
header lines

Carriage return line feed indicates end of message

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close
Accept-language: fr
```

(extra carriage return, line feed)

# HTTP请求消息的通用格式



# HTTP响应消息

status line  
(protocol status code status phrase)

header lines

data, e.g., requested HTML file

```
HTTP/1.1 200 OK
Connection: close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 .....
Content-Length: 6821
Content-Type: text/html
```

data data data data data ...



## 参考资料

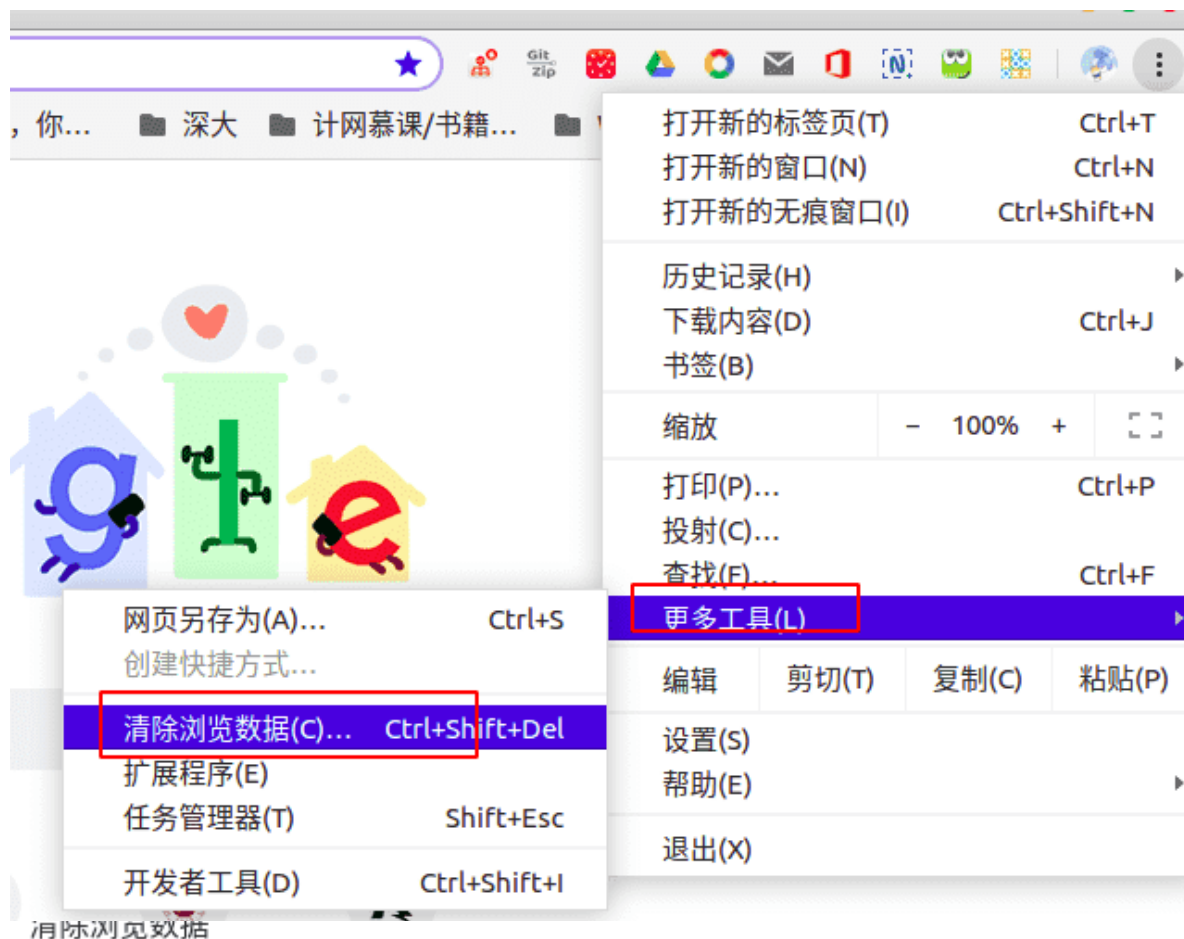
- [Wireshark实验HTTP](#)
- [HTTP验证方案](#)
- [HTTP与HTTPS的区别](#)

## 附录

### 1.谷歌浏览器清除缓存

(记录一下防止以后忘了)

来到 **更多工具** -> **清除浏览数据**。选择 **过去一个小时**。



基本 高级

时间范围 过去一小时

☒ 浏览记录  
清除所有登录过的设备上的历史记录。您的 Google 帐号在 [myactivity.google.com](https://myactivity.google.com) 上可能有其他形式的浏览记录。

☐ Cookie 及其他网站数据  
会致使您从大多数网站退出。但您的 Google 帐号仍会保持登录状态，以便清除您的已同步数据。

☒ 缓存的图片和文件  
释放了 3.5 MB。当您下次访问时，某些网站的加载速度可能会更慢。

取消 清除数据

## 2.HTTP验证方案

参考资料第二点，是一个对HTTP验证方案的简单介绍

接下来，客户端发出另一个请求，这次包括一个Authentication标头字段，其中包含适用于服务器身份验证质询的客户端凭据。

如果服务器接受凭据，它将返回请求的页面。否则，它将返回另一个“401未经授权”响应，以通知客户端身份验证已失败。

您输入的用户名(wireshark-students)和密码(network)按照客户端HTTP GET消息中请求头的“Authorization:Basic”的字符串(d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=)编码。

所谓用户名和密码可能“加密”，只是以一种称为Base64格式的格式进行编码。打开在线[解码工具](#)，输入HTTP GET消息中请求头的 Authorization:Basic 的字符串，也就是 d2lyZXNoYXJrLXN0dWRlbnRzMz0ldHdvcm0=。选择解码 decode。

最后转换出来的 ASCII 码，用户名字跟密码都有了。

Source data from the Base64 string:

wireshark-students:network

Type (or copy-paste) some text to a textbox below. The text can be a Base64 string to decode or any string to encode to a Base64

d2lyZXNoYXJrLXN0dWR1bnRzOm5ldHdvcmcs=

## 4.HTTP与HTTPS

参考资料第三点，就截取一些片段吧

HTTPS协议的主要作用可以分为两种：一种是建立一个信息安全通道，来保证数据传输的安全；另一种就是确认网站的真实性。

简单来说，HTTPS协议是由SSL+HTTP协议构建的可进行加密传输、身份认证的网络协议，要比http协议安全。

HTTPS和HTTP的区别主要如下：

- 1、https协议需要到ca申请证书，一般免费证书较少，因而需要一定费用。
- 2、http是超文本传输协议，信息是明文传输，https则是具有安全性的ssl加密传输协议。
- 3、http和https使用的是完全不同的连接方式，用的端口也不一样，前者是80，后者是443。
- 4、http的连接很简单，是无状态的；HTTPS协议是由SSL+HTTP协议构建的可进行加密传输、身份认证的网络协议，比http协议安全。

HTTP使用TCP三次握手建立连接，客户端和服务端需要交换3个包；HTTPS除了TCP的三个包，还要加上ssl握手需要的9个包，所以一共是12个包。