计算机网络第二次实验实验报告

舒文炫

2021 年 9 月 19 日

目录

Chapter 1

实验内容

本次实验，我们将更深入的了解 http 协议更详细的内容，主要分为五个方面:

- 1.The Basic HTTP GET/response interaction
- 2.The HTTP CONDITIONAL GET/response interaction
- 3.Retrieving Long Documents
- 4.HTML Documents with Embedded Objects
- 5.HTTP Authentication

下面我将依次进行这几项实验。

Chapter 2

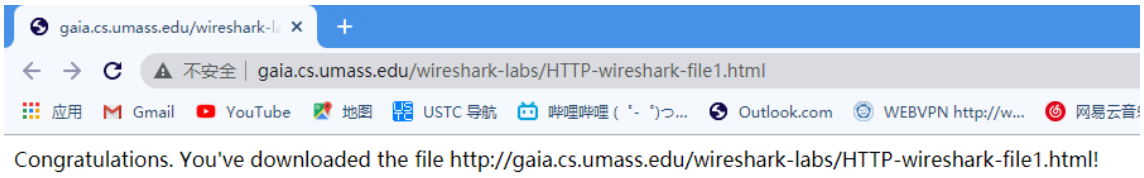
实验过程

2.1 The Basic HTTP GET/response interaction

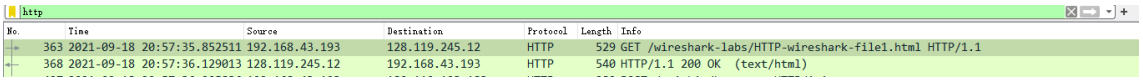
2.1.1 过程截图

在本小节，将通过浏览器下载一个非常简单 (短且只有文本) 的 HTML 文件，使用 wireshark 抓包获得 http 报文的基本信息.

首先，打开浏览器，打开 wireshark 进行捕获，浏览器中输入网址 http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html 网址打开界面如下

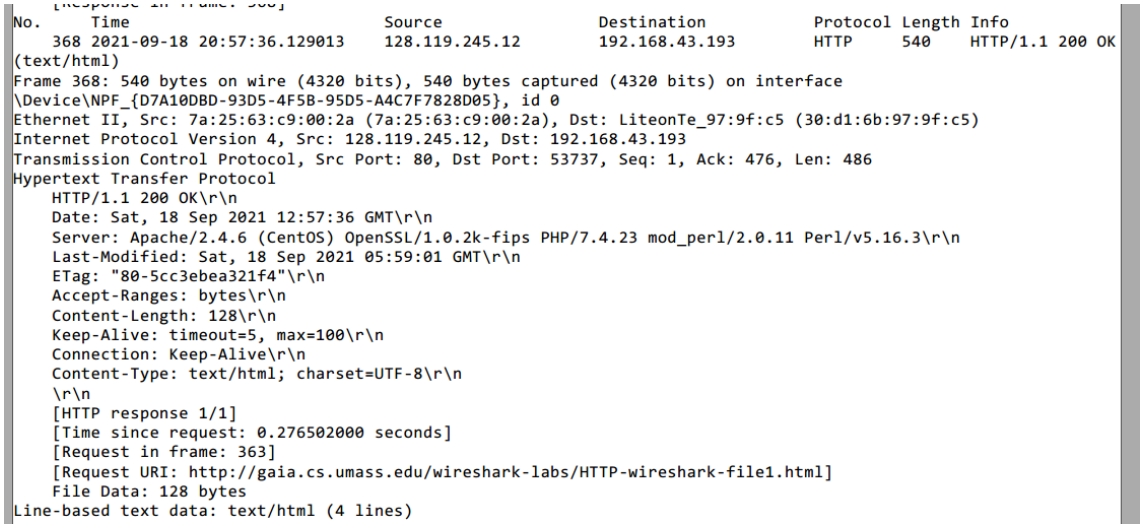


停止捕获，可以在抓包列表中看到这样两行, 这是关于这个 HTML 的 get 和 response 消息



将这两行的消息打印到 pdf 中截图如下

```
No.     Time                           Source                Destination           Protocol Length Info
    363 2021-09-18 20:57:35.852511     192.168.43.193        128.119.245.12        HTTP     529    GET /wireshark-
labs/HTTP-wireshark-file1.html HTTP/1.1
Frame 363: 529 bytes on wire (4232 bits), 529 bytes captured (4232 bits) on interface
\Device\NPF_{D7A10DBD-93D5-4F5B-95D5-A4C7F7828D05}, id 0
Ethernet II, Src: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5), Dst: 7a:25:63:c9:00:2a (7a:25:63:c9:00:2a)
Internet Protocol Version 4, Src: 192.168.43.193, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 53737, Dst Port: 80, Seq: 1, Ack: 1, Len: 475
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 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
93.0.4577.82 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/
*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: zh-CN,zh;q=0.9\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: 368]
```



做这个小节的时候，其实一开始没找到对应的包，就刷新了那个页面，结果后面都没出现想要的内容，后来清了缓存才解决，具体原因在第二小节。

2.1.2 小节思考题

1. Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?

我的浏览器 HTTP 版本是 1.1, 服务器的 HTTP 版本是 1.1，这个可以从我打印到 pdf 的信息的截图中看到，浏览器的 HTTP 版本从这里看出：GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1
服务器的 HTTP 版本从这里看出：HTTP/1.1 200 OK

2. What languages (if any) does your browser indicate that it can accept to the server?

在 get 消息里面，有一行 Accept-Language: zh-CN,zh;q=0.9，这个表示我的浏览器接受简体中文。

3. What is the IP address of your computer? Of the gaia.cs.umass.edu server?

这个可以直接在抓包的列表中看到,编号 363,get 请求的 source 即为我电脑的 IP 地址,为 192.168.43.193。destination 即为服务器的地址，为 128.119.245.12。

4. What is the status code returned from the server to your browser?

这个查找 response 消息，里面有一行 HTTP/1.1 200 OK，200 即为服务器返回的状态码，表示这个请求被许可，我们获得了这个 HTML 文件。

5. When was the HTML file that you are retrieving last modified at the server?

这个查找 response 消息，里面有一行 Last-Modified: Sat, 18 Sep 2021 05:59:01 GMT，GMT 是格林尼治时间。

6. How many bytes of content are being returned to your browser?

这个查找 response 消息，里面有一行 File Data: 128 bytes，表示文件内容有 128bytes。

7. By inspecting the raw data in the packet content window, do you see any headers within the data

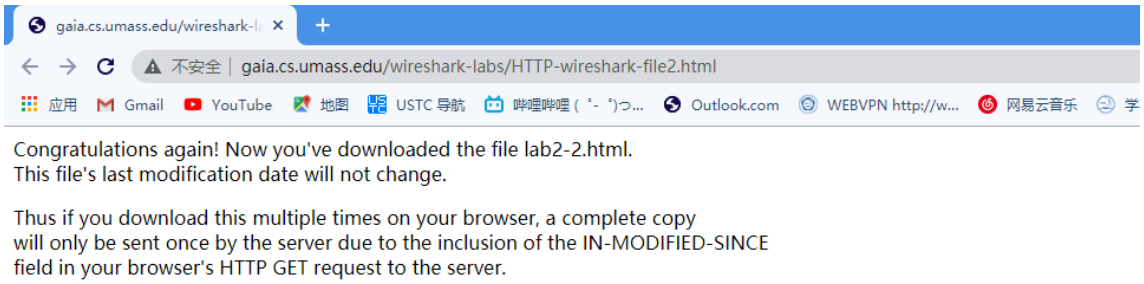
that are not displayed in the packet-listing window? If so, name one.

在抓包列表里面只有包的序号，时间，源，目的地，协议名称，长度以及简单的信息，原始数据不在其中的 headers，比如 Connection。

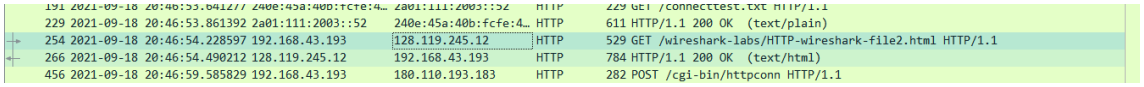
2.2 The HTTP CONDITIONAL GET/response interaction

2.2.1 过程截图

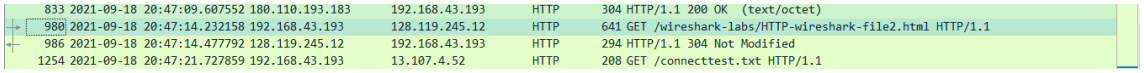
这一小节我们将看到浏览器的缓存机制，从而存在一个条件 GET，首先我打开浏览器，启动了 wireshark 捕获，然后输入网址 http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html，打开了如下的 html 文件



此时 wireshark 抓到的包如下：



然后点击刷新当前页面，wireshark 抓到的包如下：



我将这两次的具体内容打印到了 pdf 里面，截图如下：
这是第一次 get 请求对应的信息：

```
No.     Time                           Source                Destination           Protocol Length Info
    254 2021-09-18 20:46:54.228597     192.168.43.193        128.119.245.12        HTTP     529    GET /wireshark
labs/HTTP-wireshark-file2.html HTTP/1.1
Frame 254: 529 bytes on wire (4232 bits), 529 bytes captured (4232 bits) on interface
\Device\NPF_{D7A10DBD-93D5-4F5B-95D5-A4C7F7828D05}, id 0
Ethernet II, Src: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5), Dst: 7a:25:63:c9:00:2a (7a:25:63:c9:00:2a)
Internet Protocol Version 4, Src: 192.168.43.193, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 49697, Dst Port: 80, Seq: 1, Ack: 1, Len: 475
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 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
93.0.4577.82 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/
*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: zh-CN,zh;q=0.9\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: 266]
No.     Time                           Source                Destination           Protocol Length Info
    266 2021-09-18 20:46:54.490212     128.119.245.12        192.168.43.193        HTTP     784    HTTP/1.1 200 O
(text/html)
Frame 266: 784 bytes on wire (6272 bits), 784 bytes captured (6272 bits) on interface
\Device\NPF_{D7A10DBD-93D5-4F5B-95D5-A4C7F7828D05}, id 0
Ethernet II, Src: 7a:25:63:c9:00:2a (7a:25:63:c9:00:2a), Dst: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5)
Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.43.193
Transmission Control Protocol, Src Port: 80, Dst Port: 49697, Seq: 1, Ack: 476, Len: 730
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Date: Sat, 18 Sep 2021 12:46:54 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.23 mod_perl/2.0.11 Perl/v5.16.3\r\n
    Last-Modified: Sat, 18 Sep 2021 05:59:01 GMT\r\n
    ETag: "173-5cc3ebea31a24"\r\n
    Accept-Ranges: bytes\r\n
    Content-Length: 371\r\n
    Keep-Alive: timeout=5, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=UTF-8\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.261615000 seconds]
    [Request in frame: 254]
    [Request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
    File Data: 371 bytes
```

这是第二次对应的信息:

```
No.     Time                           Source                Destination           Protocol Length Info
    980 2021-09-18 20:47:14.232158     192.168.43.193        128.119.245.12        HTTP     641    GET /wireshark-
labs/HTTP-wireshark-file2.html HTTP/1.1
Frame 980: 641 bytes on wire (5128 bits), 641 bytes captured (5128 bits) on interface
\Device\NPF_{D7A10DBD-93D5-4F5B-95D5-A4C7F7828D05}, id 0
Ethernet II, Src: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5), Dst: 7a:25:63:c9:00:2a (7a:25:63:c9:00:2a)
Internet Protocol Version 4, Src: 192.168.43.193, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 64955, Dst Port: 80, Seq: 1, Ack: 1, Len: 587
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 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
93.0.4577.82 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/
*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: zh-CN,zh;q=0.9\r\n
    If-None-Match: "173-5cc3ebea31a24"\r\n
    If-Modified-Since: Sat, 18 Sep 2021 05:59:01 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: 986]
No.     Time                           Source                Destination           Protocol Length Info
    986 2021-09-18 20:47:14.477792     128.119.245.12        192.168.43.193        HTTP     294    HTTP/1.1 304
Not Modified
Frame 986: 294 bytes on wire (2352 bits), 294 bytes captured (2352 bits) on interface
\Device\NPF_{D7A10DBD-93D5-4F5B-95D5-A4C7F7828D05}, id 0
Ethernet II, Src: 7a:25:63:c9:00:2a (7a:25:63:c9:00:2a), Dst: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5)
Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.43.193
Transmission Control Protocol, Src Port: 80, Dst Port: 64955, Seq: 1, Ack: 588, Len: 240
Hypertext Transfer Protocol
    HTTP/1.1 304 Not Modified\r\n
    Date: Sat, 18 Sep 2021 12:47:14 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.23 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: "173-5cc3ebea31a24"\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.245634000 seconds]
    [Request in frame: 980]
    [Request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
```

2.2.2 小节思考题

8. Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE” line in the HTTP GET?

观察第一次的 get 消息，里面没有 IF-MODIFIED-SINCE 行。

9. Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?

观察第一次的 response 消息可以看到有 Content-Length: 371 和 Content-Type: text/html; charset=UTF-8 这两行，表示内容长度 371bytes，内容是文本，采用 UTF-8 编码。从而可以看出确实服务器返回了文件的内容。

10. Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE:” line in the HTTP GET? If so, what information follows the “IF-MODIFIED-SINCE:” header?

观察第二次 get 的消息，里面有 If-Modified-Since: Sat, 18 Sep 2021 05:59:01 GMT，表示询问是否在那个时候修改了。

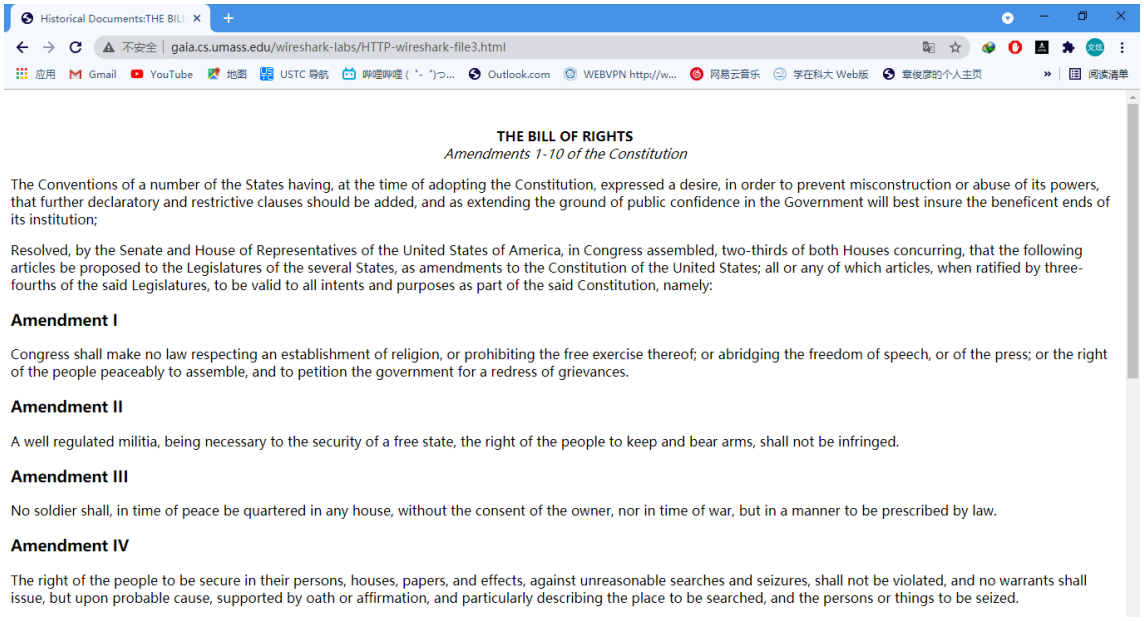
11. What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.

观察第二次 response 消息，有一行 HTTP/1.1 304 Not Modified，即返回的状态码是 304，后面跟着 not modified，表示没有修改，这里我没用找到有关文件内容描述的信息，这是因为本地将该文件缓存了，并且在服务器端这个文件并没有修改，从而显示的是本地缓存的文件，服务器也就没用必要返回文件内容。

2.3 Retrieving Long Documents

2.3.1 过程截图

在本小节，将通过浏览器下载一个大文件，先打开浏览器，启动 wireshark 捕获，输入网址 http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html 打开了如下内容，这个内容很长，我只截出一部分来展示。



Historical Documents:THE BILL

不安全 | gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html

THE BILL OF RIGHTS

Amendments 1-10 of the Constitution

The Conventions of a number of the States having, at the time of adopting the Constitution, expressed a desire, in order to prevent misconstruction or abuse of its powers, that further declaratory and restrictive clauses should be added, and as extending the ground of public confidence in the Government will best insure the beneficent ends of its institution;

Resolved, by the Senate and House of Representatives of the United States of America, in Congress assembled, two-thirds of both Houses concurring, that the following articles be proposed to the Legislatures of the several States, as amendments to the Constitution of the United States; all or any of which articles, when ratified by three-fourths of the said Legislatures, to be valid to all intents and purposes as part of the said Constitution, namely:

Amendment I

Congress shall make no law respecting an establishment of religion, or prohibiting the free exercise thereof; or abridging the freedom of speech, or of the press; or the right of the people peaceably to assemble, and to petition the government for a redress of grievances.

Amendment II

A well regulated militia, being necessary to the security of a free state, the right of the people to keep and bear arms, shall not be infringed.

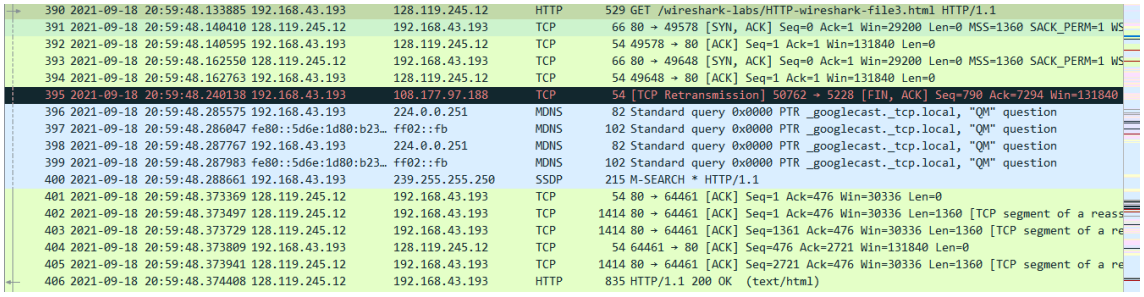
Amendment III

No soldier shall, in time of peace be quartered in any house, without the consent of the owner, nor in time of war, but in a manner to be prescribed by law.

Amendment IV

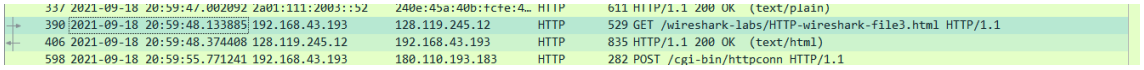
The right of the people to be secure in their persons, houses, papers, and effects, against unreasonable searches and seizures, shall not be violated, and no warrants shall issue, but upon probable cause, supported by oath or affirmation, and particularly describing the place to be searched, and the persons or things to be seized.

然后抓包的列表中出现如下内容，这里有四个 TCP 进行文件的传输，跟在了 httpGET 的后面：



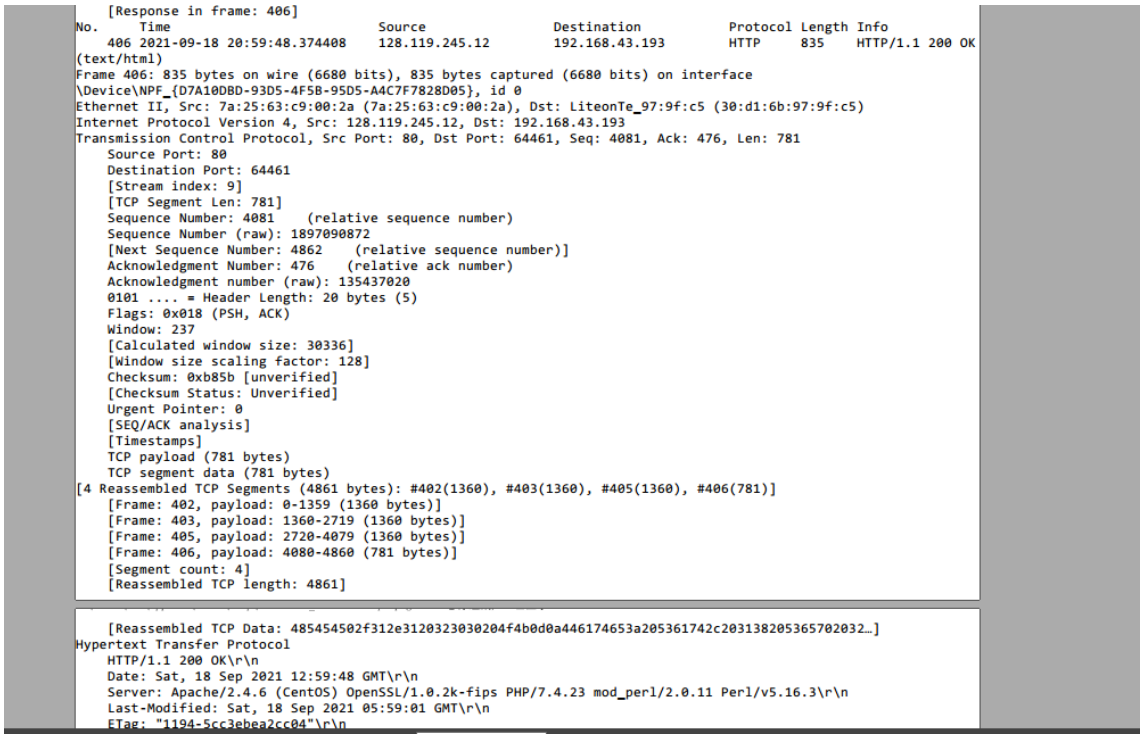
390	2021-09-18 20:59:48.133885	192.168.43.193	128.119.245.12	HTTP	529 GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
391	2021-09-18 20:59:48.140410	128.119.245.12	192.168.43.193	TCP	66 80 → 49578 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1360 SACK_PERM=1 WS
392	2021-09-18 20:59:48.140595	192.168.43.193	128.119.245.12	TCP	54 49578 → 80 [ACK] Seq=1 Ack=1 Win=131840 Len=0
393	2021-09-18 20:59:48.162550	128.119.245.12	192.168.43.193	TCP	66 80 → 49648 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1360 SACK_PERM=1 WS
394	2021-09-18 20:59:48.162763	192.168.43.193	128.119.245.12	TCP	54 49648 → 80 [ACK] Seq=1 Ack=1 Win=131840 Len=0
395	2021-09-18 20:59:48.240138	192.168.43.193	108.177.97.188	TCP	54 [TCP Retransmission] 50762 → 5228 [FIN, ACK] Seq=790 Ack=7294 Win=131840
396	2021-09-18 20:59:48.285575	192.168.43.193	224.0.0.251	MDNS	82 Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question
397	2021-09-18 20:59:48.286047	fe80::5d6e:1d80:b23…	ff02::fb	MDNS	102 Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question
398	2021-09-18 20:59:48.287767	192.168.43.193	224.0.0.251	MDNS	82 Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question
399	2021-09-18 20:59:48.287983	fe80::5d6e:1d80:b23…	ff02::fb	MDNS	102 Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question
400	2021-09-18 20:59:48.288661	192.168.43.193	239.255.255.250	SSDP	215 M-SEARCH * HTTP/1.1
401	2021-09-18 20:59:48.373369	128.119.245.12	192.168.43.193	TCP	54 80 → 64461 [ACK] Seq=1 Ack=476 Win=30336 Len=0
402	2021-09-18 20:59:48.373497	128.119.245.12	192.168.43.193	TCP	1414 80 → 64461 [ACK] Seq=1 Ack=476 Win=30336 Len=1360 [TCP segment of a reass
403	2021-09-18 20:59:48.373729	128.119.245.12	192.168.43.193	TCP	1414 80 → 64461 [ACK] Seq=1361 Ack=476 Win=30336 Len=1360 [TCP segment of a re
404	2021-09-18 20:59:48.373809	192.168.43.193	128.119.245.12	TCP	54 64461 → 80 [ACK] Seq=476 Ack=2721 Win=131840 Len=0
405	2021-09-18 20:59:48.373941	128.119.245.12	192.168.43.193	TCP	1414 80 → 64461 [ACK] Seq=2721 Ack=476 Win=30336 Len=1360 [TCP segment of a re
406	2021-09-18 20:59:48.374408	128.119.245.12	192.168.43.193	HTTP	835 HTTP/1.1 200 OK (text/html)

过滤出 HTTP 的结果如下:



337	2021-09-18 20:59:47.002092	2a01:111:2003::52	240e:45a:40b:fcfe:4…	HTTP	611 HTTP/1.1 200 OK (text/plain)
390	2021-09-18 20:59:48.133885	192.168.43.193	128.119.245.12	HTTP	529 GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
406	2021-09-18 20:59:48.374408	128.119.245.12	192.168.43.193	HTTP	835 HTTP/1.1 200 OK (text/html)
598	2021-09-18 20:59:55.771241	192.168.43.193	180.110.193.183	HTTP	282 POST /cgi-bin/httpconn HTTP/1.1

我将结果打印到了 pdf 里面，这里只截出我们做思考题需要的部分



```
    [Response in frame: 406]
No.     Time                    Source                Destination           Protocol Length Info
    406 2021-09-18 20:59:48.374408    128.119.245.12        192.168.43.193        HTTP     835    HTTP/1.1 200 OK
(text/html)
Frame 406: 835 bytes on wire (6680 bits), 835 bytes captured (6680 bits) on interface
\Device\NPF_{D7A10DBD-93D5-4F5B-95D5-A4C7F7828D05}, id 0
Ethernet II, Src: 7a:25:63:c9:00:2a (7a:25:63:c9:00:2a), Dst: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5)
Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.43.193
Transmission Control Protocol, Src Port: 80, Dst Port: 64461, Seq: 4081, Ack: 476, Len: 781
    Source Port: 80
    Destination Port: 64461
    [Stream index: 9]
    [TCP Segment Len: 781]
    Sequence Number: 4081    (relative sequence number)
    Sequence Number (raw): 1897090872
    [Next Sequence Number: 4862    (relative sequence number)]
    Acknowledgment Number: 476    (relative ack number)
    Acknowledgment number (raw): 135437020
    0101 .... = Header Length: 20 bytes (5)
    Flags: 0x018 (PSH, ACK)
    Window: 237
    [Calculated window size: 30336]
    [Window size scaling factor: 128]
    Checksum: 0xb85b [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
    [SEQ/ACK analysis]
    [Timestamps]
    TCP payload (781 bytes)
    TCP segment data (781 bytes)
[4 Reassembled TCP Segments (4861 bytes): #402(1360), #403(1360), #405(1360), #406(781)]
    [Frame: 402, payload: 0-1359 (1360 bytes)]
    [Frame: 403, payload: 1360-2719 (1360 bytes)]
    [Frame: 405, payload: 2720-4079 (1360 bytes)]
    [Frame: 406, payload: 4080-4860 (781 bytes)]
    [Segment count: 4]
    [Reassembled TCP length: 4861]
    [Reassembled TCP Data: 485454502f312e3120323030204f4b0d0a446174653a205361742c20313820536570202032…]
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Date: Sat, 18 Sep 2021 12:59:48 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.23 mod_perl/2.0.11 Perl/v5.16.3\r\n
    Last-Modified: Sat, 18 Sep 2021 05:59:01 GMT\r\n
    ETag: "1194-5cc3ebea2cc04"\r\n
```

2.3.2 小节思考题

12. How many HTTP GET request messages did your browser send? Which packet number in the trace contains the GET message for the Bill or Rights?

如果只看关于获取这个 html 文件的 get 请求，就只有我截出来的一个，其他有很多 http 请求是进行 contenttest，而且只要不停止抓包，会一直生成，与本实验没有什么关系，数这个也没有什么意义。从抓包列表中看到第 390 号储存了 GET 消息。

13. Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request?

从我截出来的那一段可以看到 response 消息是在第 406 号。

14. What is the status code and phrase in the response?

查看我打印出的 pdf 截图，里面有一行 HTTP/1.1 200 OK，即状态码 200，表示 OK。

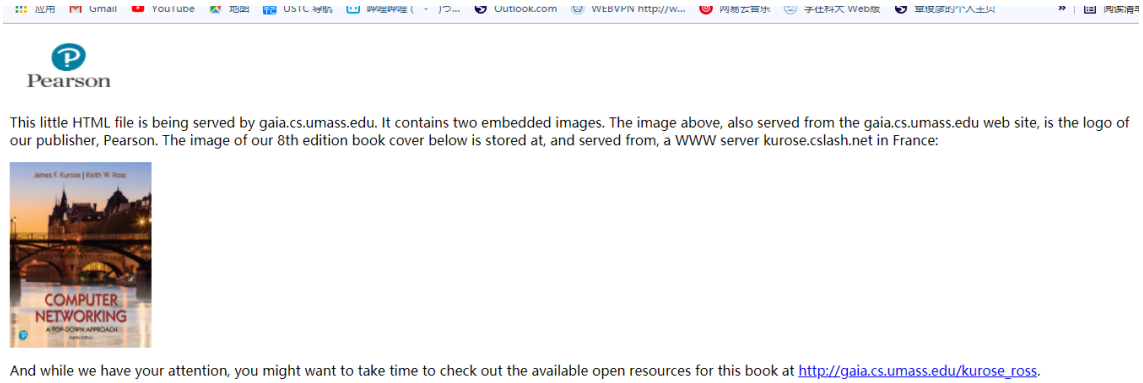
15. How many data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights?

查看我打印出的 pdf 截图，上面 TCP 部分有一个 Segment count: 4，即有 4 个 TCP 报文段，而且再上面四行可以看到每一段传了多少内容。

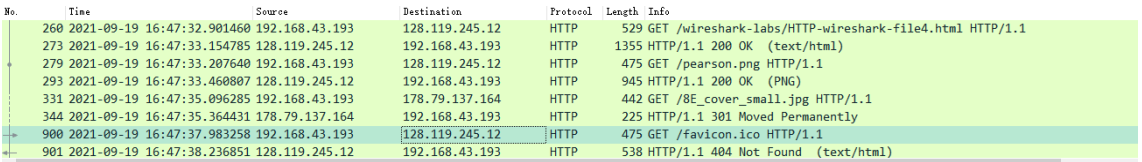
2.4 HTML Documents with Embedded Objects

2.4.1 过程截图

本小节将进行有嵌入内容的 html 文件的下载，这里面的文件将不再只有文本，还会包含两张图片。首先打开浏览器，启动 wireshark 捕获，输入网址 http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html，下面是浏览器显示截图：



然后抓包的列表有下列内容：



No.	Time	Source	Destination	Protocol	Length	Info
260	2021-09-19 16:47:32.901460	192.168.43.193	128.119.245.12	HTTP	529	GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
273	2021-09-19 16:47:33.154785	128.119.245.12	192.168.43.193	HTTP	1355	HTTP/1.1 200 OK (text/html)
279	2021-09-19 16:47:33.207640	192.168.43.193	128.119.245.12	HTTP	475	GET /pearson.png HTTP/1.1
293	2021-09-19 16:47:33.460807	128.119.245.12	192.168.43.193	HTTP	945	HTTP/1.1 200 OK (PNG)
331	2021-09-19 16:47:35.096285	192.168.43.193	178.79.137.164	HTTP	442	GET /8E_cover_small.jpg HTTP/1.1
344	2021-09-19 16:47:35.364431	178.79.137.164	192.168.43.193	HTTP	225	HTTP/1.1 301 Moved Permanently
900	2021-09-19 16:47:37.983258	192.168.43.193	128.119.245.12	HTTP	475	GET /favicon.ico HTTP/1.1
901	2021-09-19 16:47:38.236851	128.119.245.12	192.168.43.193	HTTP	538	HTTP/1.1 404 Not Found (text/html)

2.4.2 小节思考题

16. How many HTTP GET request messages did your browser send? To which Internet addresses were these GET requests sent?

观察抓包的列表,这里相关的 GET 的请求一共有四个,260 号 GET 请求 html 文件,发往 128.119.245.12,273 号请求 pearson.png 发往 128.119.245.12，331 号 GET 请求一个 jpg 文件，发往 178.79.137.164，900 号 GET 请求一个 favicon.ico, 发往 128.119.245.12

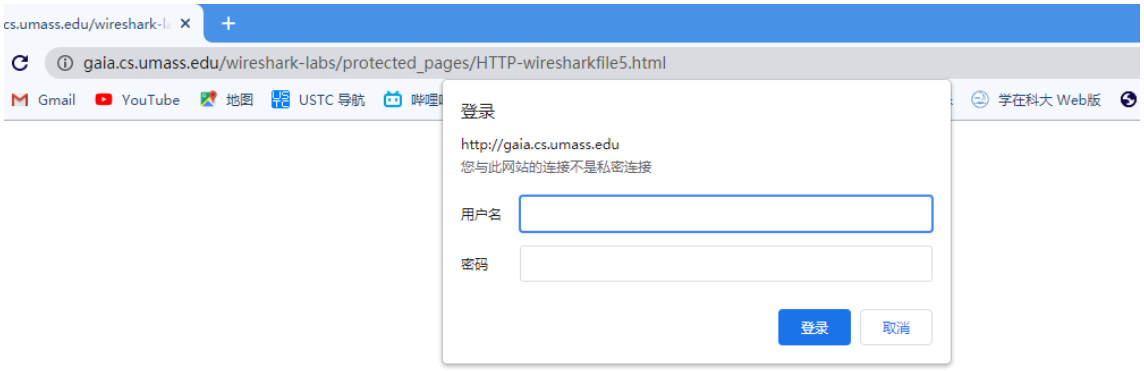
17. Can you tell whether your browser downloaded the two images serially, or whether they were downloaded from the two web sites in parallel? Explain.

观察抓包的列表，发现那两个请求图片的 GET 请求 ip 地址不一样，从而它们是平行的从两个网址下载的。

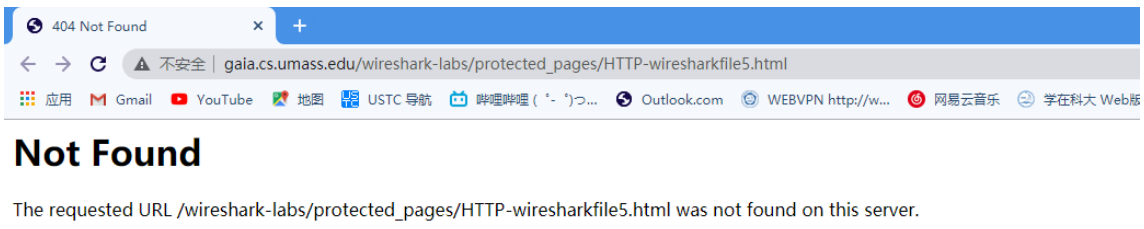
2.5 HTTP Authentication

2.5.1 过程截图

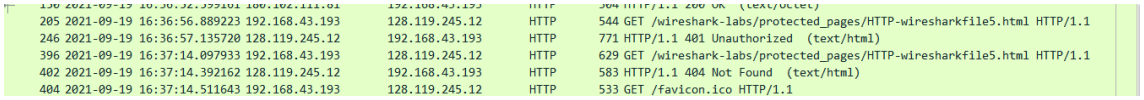
这一小节,我将访问那种带密码保护的 http 网址,并进行抓包观察这一类的网址不一样的地方。先打开浏览器,打开 wireshark 捕获,输入网址 http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wiresharkfile5.html，这时弹出一个输入密码的窗口如下：



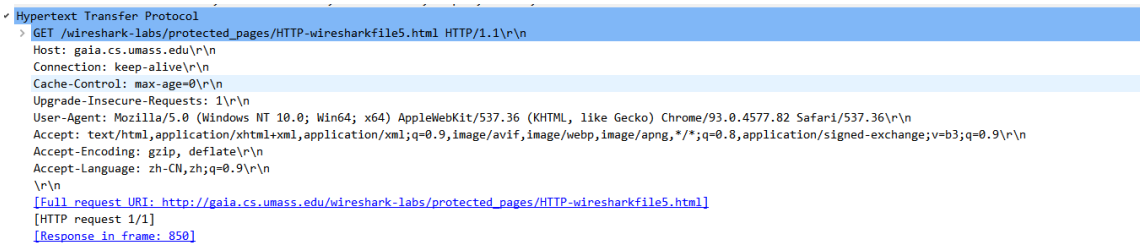
输入账号 wireshark-students，密码 network，进入了网页，不过可惜的是得到了如下的显示，不清楚是这个文件被从服务器上删除了，还是需要其他的方法得到，不过对完成本实验来说，已经足够了：



下面是抓包截图，上面是第一次打开，发现需要输入密码，下面那个 get 是输完了密码，去访问结果 404 了：



这是第一次 get 的具体信息：



这是第二次 get 的具体信息：

2.5.2 小节思考题

18. What is the server’s response (status code and phrase) in response to the initial HTTP GET message from your browser?

从上面抓包列表的截图可以看到，第一次 get 的响应状态码是 401，后面跟着说明 Unauthorized, 表示未授权。

19. When your browser’s sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?

我们对比两次 get 的具体信息的截图，发现第二次多了一个 Authorization 行
Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=
Credentials: wireshark-students:network

Chapter 3

实验总结

本次实验通过 wireshark 抓包，观察了各种各样的 html 文件请求响应的报文，对 http 报文有了更深刻的认识。