重庆大学大数据与软件学院

上机实验报告

上机实践项目 HTTP 协议分析

课程名称 计算机网络

姓名	XXX	成绩	
学号	2021XXXX	教师	**XXX**
班级	软工 X 班	日期	2023/3/16

《计算机网络》上机实验报告

开课实验室：DS1502 **2023 年 3 月 16 日**

姓　名	XXX	年级、班级	2021 级软件工程 X 班	学号	**2021XXXX**
上机（项目）名称		HTTP 协议分析		指导教师	XXX
教师评语	教师签名： **年　月　日**				

一、上机目的

In this lab, we’ll explore several aspects of the HTTP protocol:

- the basic GET/response interaction;
- HTTP message formats;
- Retrieving large HTML files；
- Retrieving HTML files with embedded objects；
- HTTP authentication and security.

Before beginning these labs, you might want to review Section 2.2 of the text.1

二、基本原理

Wireshark: Wireshark is a popular network protocol analysis tool that captures network traffic and analyzes the communication process of various protocols. It is a free and open-source software that can run on multiple operating systems including Windows, Mac OS X, and Linux.Wireshark captures packets from a network interface and can analyze them in real-time or offline. It supports various protocols such as TCP, UDP, IP, HTTP, FTP, DNS, ICMP, and can also decode TLS and SSL encrypted traffic.By using Wireshark, users can perform deep analysis of network communication processes, including understanding the data type, protocol header information, packet transmission time, and relationships between different packets. This is extremely useful for network administrators and security professionals.

HTTP: HTTP (HyperText Transfer Protocol) is a protocol used to transfer data between web browsers and web servers. It is a client-server protocol, where the client sends HTTP requests to fetch resources, and the server responds with HTTP responses to transfer the requested resources.HTTP uses TCP as its transport protocol and is a stateless protocol, which means the server does not retain any state information about the client, and each request is independent.HTTP is commonly used to transfer resources such as HTML, CSS, JavaScript, images, and more, between web browsers and web servers. In addition to the basic HTTP requests and responses, HTTP defines many other features such as caching, authentication, security, and more.

Windows: Here are the steps to use Wireshark software on Windows for analyzing HTTP protocol:

・Download and install Wireshark software, and then open it.

・Select the network interface you want to monitor in Wireshark and click the "Start" button to begin packet capturing.

・Open a web browser and enter the website address you want to access.

・Stop packet capturing in Wireshark, and filter out HTTP protocol packets using a filter. In the filter, type "http," and then click the "Apply" button.

・Wireshark will display only HTTP protocol packets. Select an HTTP packet and view detailed information, including HTTP request and response headers, message bodies, and more.

・With the Wireshark software, we can gain a deeper understanding of the communication process of the HTTP protocol and view various parameters and data in the protocol. This is very useful for network management and security-related work.

三、使用的软件、硬件

Software: Windows11 && Wireshark && Firefox;

Hardware: Lenovo Legion R9000P2021H.

四、上机操作步骤

1. The Basic HTTP GET/response interaction

1) Start up your web browser.

2) Start up the Wireshark packet sniffer, as described in the Introductory lab (but don’t yet begin packet capture). Enter “http” (just the letters, not the quotation marks) in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window. (We’re only interested in the HTTP protocol here, and don’t want to see the clutter of all captured packets).

3) Wait a bit more than one minute (we’ll see why shortly), and then begin Wireshark packet capture.

4) Enter the following to your browser

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

Your browser should display the very simple, one-line HTML file.

5) Stop Wireshark packet capture.

2. The HTTP CONDITIONAL GET/response interaction

1) Start up your web browser, and make sure your browser’ s cache is cleared, as discussed above.

2) Start up the Wireshark packet sniffer

3) Enter the following URL into your browser http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html Your browser should display a very simple five-line HTML file.

4) Quickly enter the same URL into your browser again (or simply select the refresh button on your browser)

5) Stop Wireshark packet capture, and enter “http” in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window.

(Note: If you are unable to run Wireshark on a live network connection, you can use the http-ethereal-trace-2 packet trace to answer the questions below; see footnote 1. This trace file was gathered while performing the steps above on one of the author’s computers.)

3. Retrieving Long Documents

1) Start up your web browser, and make sure your browser’ s cache is cleared, as discussed above.

2) Start up the Wireshark packet sniffer

3) Enter the following URL into your browser http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html Your browser should display the rather lengthy US Bill of Rights.

4) Stop Wireshark packet capture, and enter “http” in the display-filter-specification window, so that only captured HTTP messages will be displayed.

(Note: If you are unable to run Wireshark on a live network connection, you can use the http-ethereal-trace-3 packet trace to answer the questions below; see footnote 1. This trace file was gathered while performing the steps above on one of the author’s computers.)

4. HTML Documents with Embedded Objects

1) Start up your web browser, and make sure your browser’ s cache is cleared, as discussed above.

2) Start up the Wireshark packet sniffer.

3) Enter the following URL into your browser http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html Your browser should display a short HTML file with two images. These two images are referenced in the base HTML file. That is, the images themselves are not contained in the HTML; instead the URLs for the images are contained in the downloaded HTML file. As discussed in the textbook, your browser will have to retrieve these logos from the indicated web sites. Our publisher’s logo is retrieved from the www.aw-bc.com web site. The image of our book’s cover is stored at the manic.cs.umass.edu server.

4)Stop Wireshark packet capture, and enter “http” in the display-filter-specification window, so that only captured HTTP messages will be displayed.

(Note: If you are unable to run Wireshark on a live network connection, you can use the http-ethereal-trace-4 packet trace to answer the questions below; see footnote 1. This trace file was gathered while performing the steps above on one of the author’ s computers.)

5. HTTP Authentication

1) Make sure your browser’s cache is cleared, as discussed above, and close down your browser. Then, start up your browser

2) Start up the Wireshark packet sniffer

3) Enter the following URL into your browser

http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wiresharkfile5.html Type the requested user name and password into the pop up box.

4) Stop Wireshark packet capture, and enter “http” in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window.

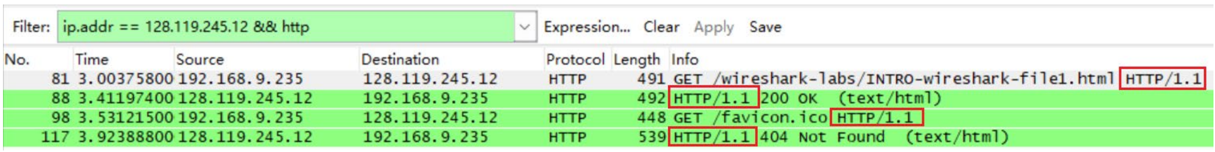
(Note: If you are unable to run Wireshark on a live network connection, you can use the http-ethereal-trace-5 packet trace to answer the questions below; see footnote 1. This trace file was gathered while performing the steps above on one of the author’s computers.)

五、过程原始记录(数据、图表、计算等)

1. The Basic HTTP GET/response interaction

Q1: Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?

A1: 我的浏览器和访问的服务器运行的 HTTP 协议版本均为 HTTP 1.1;

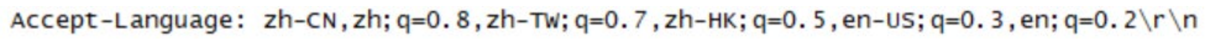


```
Filter: ip.addr == 128.119.245.12 && http          Expression... Clear Apply Save

No.  Time        Source          Destination     Protocol Length Info
  81 3.00375800 192.168.9.235   128.119.245.12  HTTP     491 GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1
  88 3.41197400 128.119.245.12  192.168.9.235   HTTP     492 HTTP/1.1 200 OK  (text/html)
  98 3.53121500 192.168.9.235   128.119.245.12  HTTP     448 GET /favicon.ico HTTP/1.1
 117 3.92388800 128.119.245.12  192.168.9.235   HTTP     539 HTTP/1.1 404 Not Found  (text/html)
```

Q2: What languages (if any) does your browser indicate that it can accept to the server?

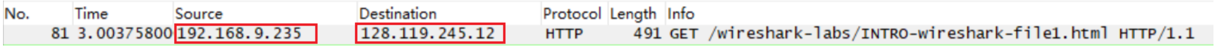
A2: 我的浏览器可以接受简体中文（zh-CN）、繁体中文-台湾（zh-TW）、繁体中文-香港（zh-HK）、英文-美国（en-US）；



```
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2\r\n
```

Q3: What is the IP address of your computer? Of the gaia.cs.umass.edu server?

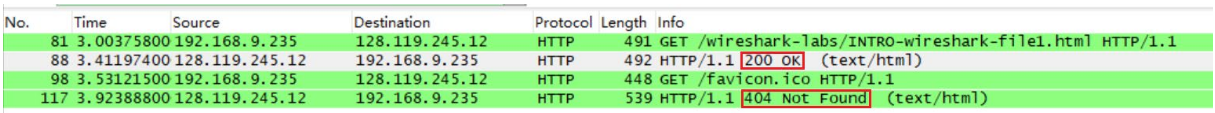
A3: 我的 IP 地址为 192.168.9.235, The gaia.cs.umass.edu server 的 IP 地址为 128.119.245.12;



```
No.  Time        Source          Destination     Protocol Length Info
  81 3.00375800 192.168.9.235   128.119.245.12  HTTP     491 GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1
```

Q4: What is the status code returned from the server to your browser?

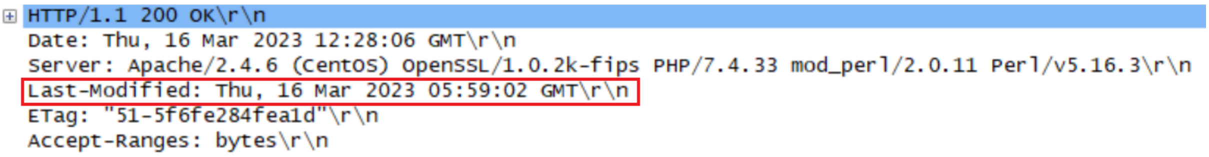
A4: 返回的状态码及短语为”200 OK” & ”404 Not Found”;



```
No.  Time        Source          Destination     Protocol Length Info
  81 3.00375800 192.168.9.235   128.119.245.12  HTTP     491 GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1
  88 3.41197400 128.119.245.12  192.168.9.235   HTTP     492 HTTP/1.1 200 OK  (text/html)
  98 3.53121500 192.168.9.235   128.119.245.12  HTTP     448 GET /favicon.ico HTTP/1.1
 117 3.92388800 128.119.245.12  192.168.9.235   HTTP     539 HTTP/1.1 404 Not Found  (text/html)
```

Q5: When was the HTML file that you are retrieving last modified at the server?

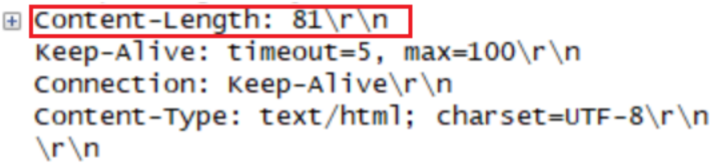
A5: 最后一次修改时间为：2023.3.16 05:59:02 星期四;



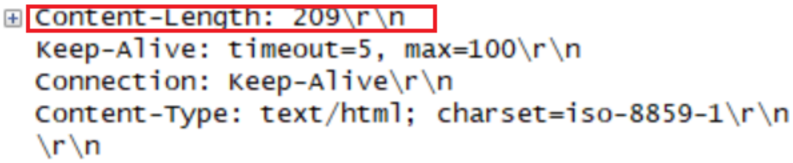
```
⊞ HTTP/1.1 200 OK\r\n
  Date: Thu, 16 Mar 2023 12:28:06 GMT\r\n
  Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3\r\n
  Last-Modified: Thu, 16 Mar 2023 05:59:02 GMT\r\n
  ETag: "51-5f6fe284fea1d"\r\n
  Accept-Ranges: bytes\r\n
```

Q6: How many bytes of content are being returned to your browser?

A6: 两次返回的内容分别有 81 字节和 209 字节;



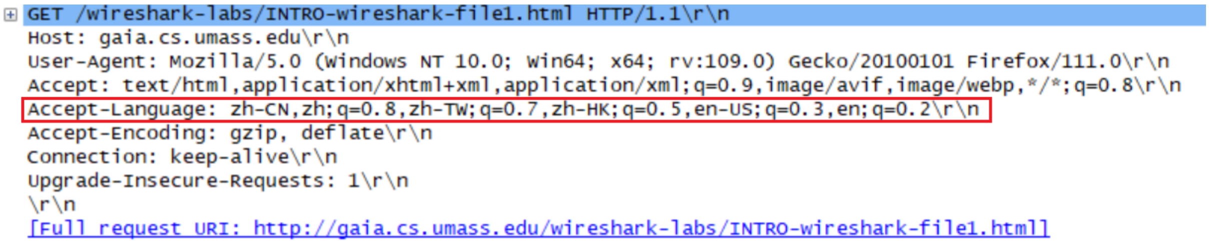
```
⊞ Content-Length: 81\r\n
  Keep-Alive: timeout=5, max=100\r\n
  Connection: Keep-Alive\r\n
  Content-Type: text/html; charset=UTF-8\r\n
  \r\n
```



```
⊞ Content-Length: 209\r\n
  Keep-Alive: timeout=5, max=100\r\n
  Connection: Keep-Alive\r\n
  Content-Type: text/html; charset=iso-8859-1\r\n
  \r\n
```

Q7: 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.

A7: 有，如 Accept-Language;

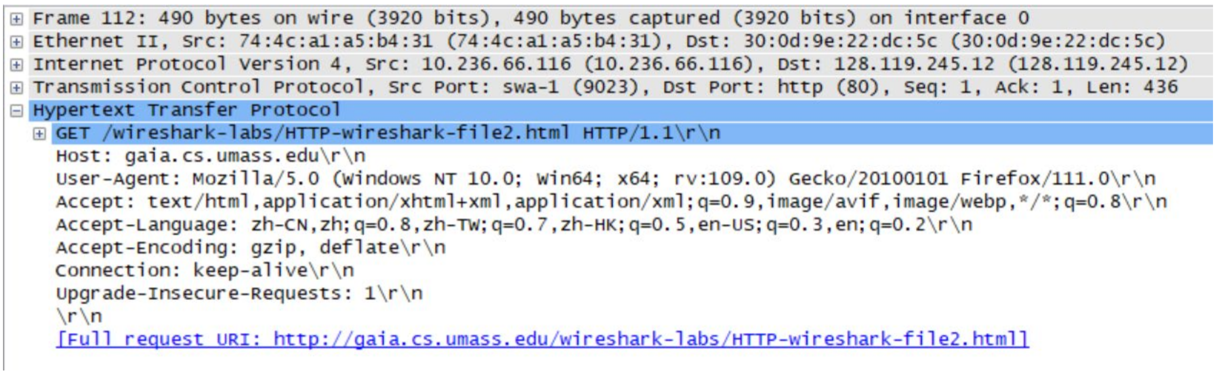


```
⊞ GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1\r\n
  Host: gaia.cs.umass.edu\r\n
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/111.0\r\n
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
  Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2\r\n
  Accept-Encoding: gzip, deflate\r\n
  Connection: keep-alive\r\n
  Upgrade-Insecure-Requests: 1\r\n
  \r\n
  [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html]
```

2. The HTTP CONDITIONAL GET/response interaction

Q8: 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?

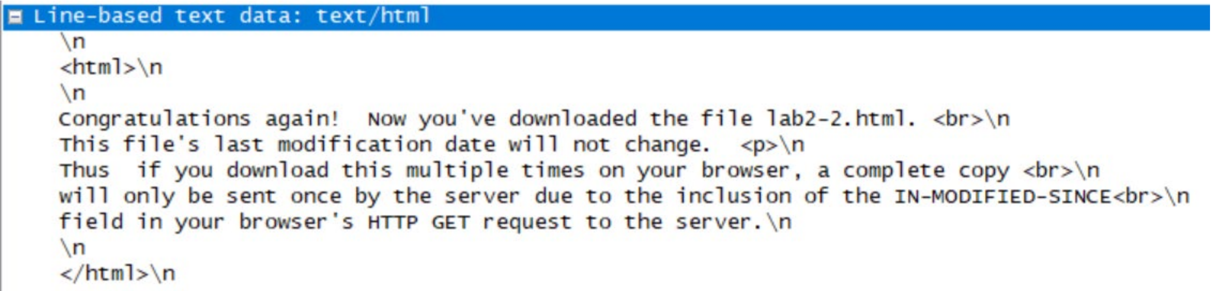
A8: 没有看到 "IF-MODIFIED-SINCE" line；



```
⊞ Frame 112: 490 bytes on wire (3920 bits), 490 bytes captured (3920 bits) on interface 0
⊞ Ethernet II, Src: 74:4c:a1:a5:b4:31 (74:4c:a1:a5:b4:31), Dst: 30:0d:9e:22:dc:5c (30:0d:9e:22:dc:5c)
⊞ Internet Protocol Version 4, Src: 10.236.66.116 (10.236.66.116), Dst: 128.119.245.12 (128.119.245.12)
⊞ Transmission Control Protocol, Src Port: swa-1 (9023), Dst Port: http (80), Seq: 1, Ack: 1, Len: 436
⊟ Hypertext Transfer Protocol
  ⊞ GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/111.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
    Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    \r\n
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
```

Q9: Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?

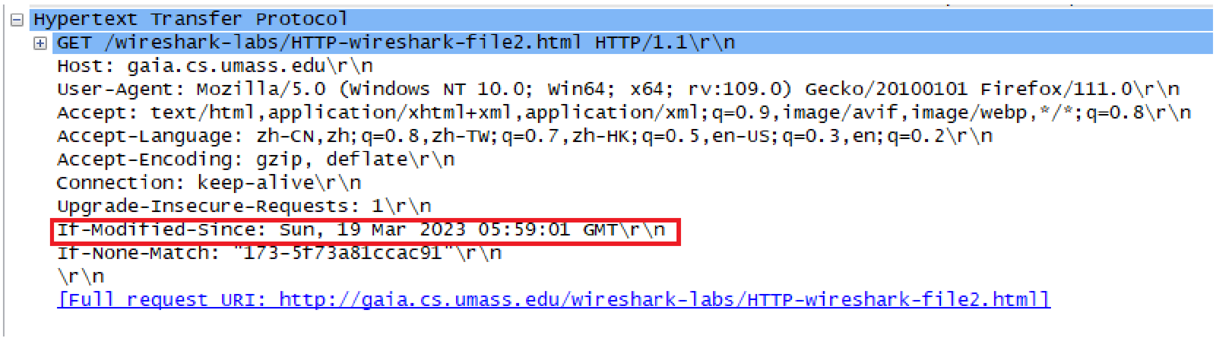
A9: 服务器显式返回了内容；



```
⊟ Line-based text data: text/html
    \n
    <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
```

Q10: 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?

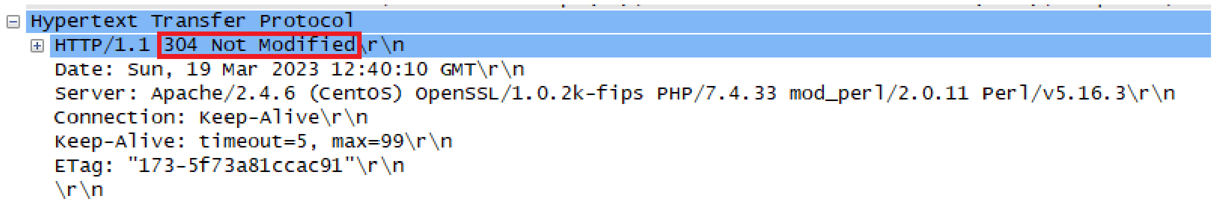
A10: 出现了“IF-MODIFIED-SINCE”字段，后面显示的信息为”Sun, 19 Mar 2023 05:59:02 GMT\r\n”;



```
⊟ Hypertext Transfer Protocol
  ⊞ GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/111.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
    Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    If-Modified-Since: Sun, 19 Mar 2023 05:59:01 GMT\r\n
    If-None-Match: "173-5f73a81ccac91"\r\n
    \r\n
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
```

Q11: 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.

A11: 返回的状态码及短语是：”304 Not Modified”，意思是没有修改，没有显式返回文件的内容，因为第一次访问时，我们的客户端已经获得了服务器端的内容，并在本地缓存了，而第二次请求 get 的时候，经验证，浏览器端缓存页面最后修改时间与服务器端时间一致，没有更新，所以返回 304 状态码，客户端接到之后，就直接把本地缓存文件显示到浏览器中。

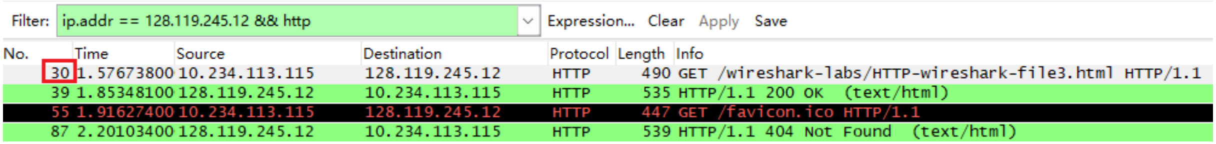


```
⊟ Hypertext Transfer Protocol
  ⊞ HTTP/1.1 304 Not Modified\r\n
    Date: Sun, 19 Mar 2023 12:40:10 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3\r\n
    Connection: Keep-Alive\r\n
    Keep-Alive: timeout=5, max=99\r\n
    ETag: "173-5f73a81ccac91"\r\n
    \r\n
```

3. Retrieving Long Documents

Q12: 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?

A12: 一共发出了 2 个 GET 请求， the packet number is 30;

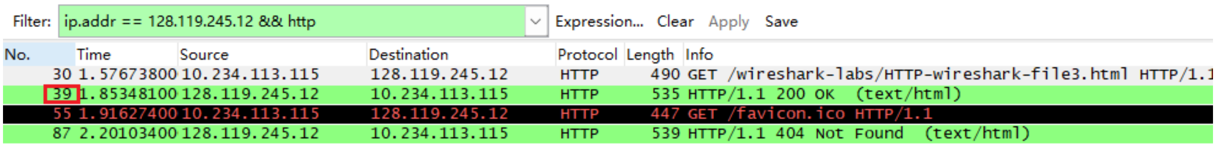


Filter: ip.addr == 128.119.245.12 && http

No.	Time	Source	Destination	Protocol	Length	Info
30	1.57673800	10.234.113.115	128.119.245.12	HTTP	490	GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
39	1.85348100	128.119.245.12	10.234.113.115	HTTP	535	HTTP/1.1 200 OK (text/html)
55	1.91627400	10.234.113.115	128.119.245.12	HTTP	447	GET /favicon.ico HTTP/1.1
87	2.20103400	128.119.245.12	10.234.113.115	HTTP	539	HTTP/1.1 404 Not Found (text/html)

Q13: Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request?

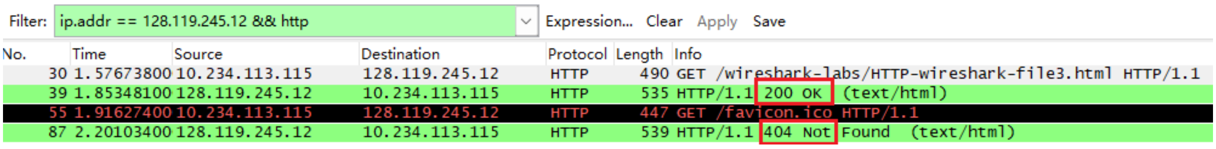
A13: The packet number is 39;



Filter: ip.addr == 128.119.245.12 && http

No.	Time	Source	Destination	Protocol	Length	Info
30	1.57673800	10.234.113.115	128.119.245.12	HTTP	490	GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
39	1.85348100	128.119.245.12	10.234.113.115	HTTP	535	HTTP/1.1 200 OK (text/html)
55	1.91627400	10.234.113.115	128.119.245.12	HTTP	447	GET /favicon.ico HTTP/1.1
87	2.20103400	128.119.245.12	10.234.113.115	HTTP	539	HTTP/1.1 404 Not Found (text/html)

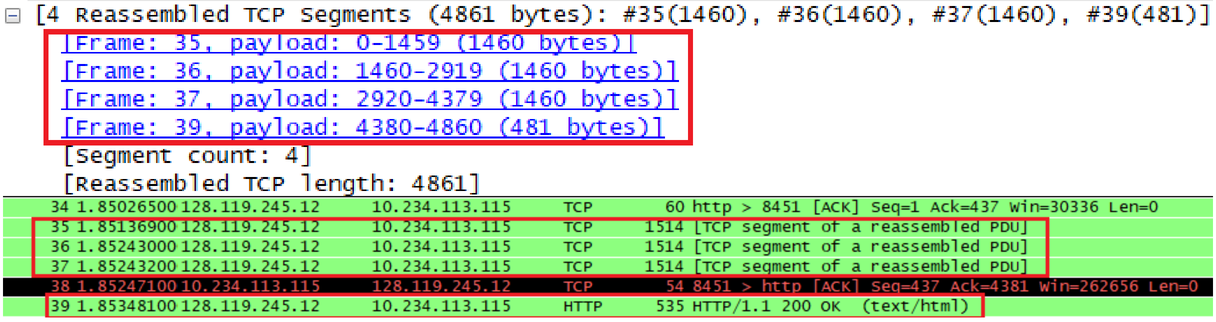
Q14: What is the status code and phrase in the response?
A14: 状态码及短语有两个，分别为”200 OK”和”404 Not Found”；



```
Filter: ip.addr == 128.119.245.12 && http                Expression... Clear Apply Save
No.   Time        Source          Destination     Protocol Length Info
   30 1.57673800 10.234.113.115  128.119.245.12  HTTP      490 GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
   39 1.85348100 128.119.245.12  10.234.113.115  HTTP      535 HTTP/1.1 200 OK  (text/html)
   55 1.91627400 10.234.113.115  128.119.245.12  HTTP      447 GET /favicon.ico HTTP/1.1
   87 2.20103400 128.119.245.12  10.234.113.115  HTTP      539 HTTP/1.1 404 Not Found  (text/html)
```

Q15: How many data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights?

A15: 共 4 个，如下图所示；

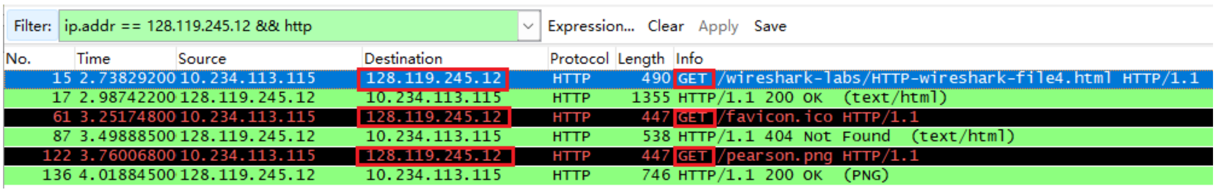


```
⊟ [4 Reassembled TCP Segments (4861 bytes): #35(1460), #36(1460), #37(1460), #39(481)]
    [Frame: 35, payload: 0-1459 (1460 bytes)]
    [Frame: 36, payload: 1460-2919 (1460 bytes)]
    [Frame: 37, payload: 2920-4379 (1460 bytes)]
    [Frame: 39, payload: 4380-4860 (481 bytes)]
    [Segment count: 4]
    [Reassembled TCP length: 4861]
   34 1.85026500 128.119.245.12  10.234.113.115  TCP       60 http > 8451 [ACK] Seq=1 Ack=437 Win=30336 Len=0
   35 1.85136900 128.119.245.12  10.234.113.115  TCP     1514 [TCP segment of a reassembled PDU]
   36 1.85243000 128.119.245.12  10.234.113.115  TCP     1514 [TCP segment of a reassembled PDU]
   37 1.85243200 128.119.245.12  10.234.113.115  TCP     1514 [TCP segment of a reassembled PDU]
   38 1.85247100 10.234.113.115  128.119.245.12  TCP       54 8451 > http [ACK] Seq=437 Ack=4381 Win=262656 Len=0
   39 1.85348100 128.119.245.12  10.234.113.115  HTTP     535 HTTP/1.1 200 OK  (text/html)
```

4. HTML Documents with Embedded Objects

Q16: How many HTTP GET request messages did your browser send? To which Internet addresses were these GET requests sent?

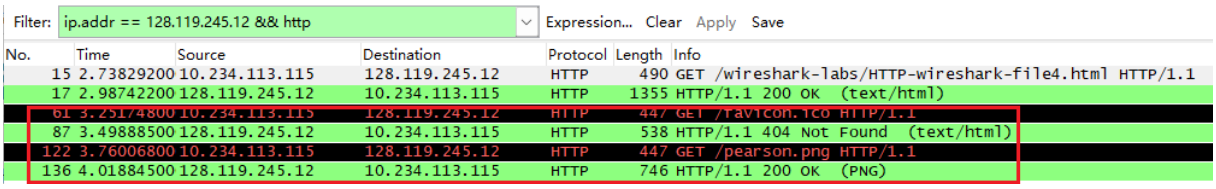
A16: 我的浏览器共发出 3 个 GET 请求，目的地址分别是：128.119.245.12，128.119.245.12，128.119.245.12；



```
Filter: ip.addr == 128.119.245.12 && http                Expression... Clear Apply Save
No.   Time        Source          Destination     Protocol Length Info
   15 2.73829200 10.234.113.115  128.119.245.12  HTTP      490 GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
   17 2.98742200 128.119.245.12  10.234.113.115  HTTP     1355 HTTP/1.1 200 OK  (text/html)
   61 3.25174800 10.234.113.115  128.119.245.12  HTTP      447 GET /favicon.ico HTTP/1.1
   87 3.49888500 128.119.245.12  10.234.113.115  HTTP      538 HTTP/1.1 404 Not Found  (text/html)
  122 3.76006800 10.234.113.115  128.119.245.12  HTTP      447 GET /pearson.png HTTP/1.1
  136 4.01884500 128.119.245.12  10.234.113.115  HTTP      746 HTTP/1.1 200 OK  (PNG)
```

Q17: Can you tell whether your browser downloaded the two images serially, or whether they were downloaded from the two web sites in parallel? Explain.

A17: 我的浏览器是通过一个网站串行下载了两个图像；因为两个 GET 请求有时间先后顺序，前一个 GET 得到响应后才发出第二个 GET，且目标地址均一样，所以判断是串行下载了两个图像；

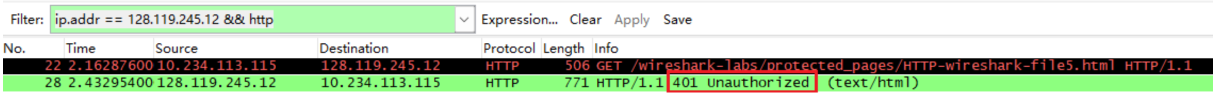


```
Filter: ip.addr == 128.119.245.12 && http                Expression... Clear Apply Save
No.   Time        Source          Destination     Protocol Length Info
   15 2.73829200 10.234.113.115  128.119.245.12  HTTP      490 GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
   17 2.98742200 128.119.245.12  10.234.113.115  HTTP     1355 HTTP/1.1 200 OK  (text/html)
   61 3.25174800 10.234.113.115  128.119.245.12  HTTP      447 GET /favicon.ico HTTP/1.1
   87 3.49888500 128.119.245.12  10.234.113.115  HTTP      538 HTTP/1.1 404 Not Found  (text/html)
  122 3.76006800 10.234.113.115  128.119.245.12  HTTP      447 GET /pearson.png HTTP/1.1
  136 4.01884500 128.119.245.12  10.234.113.115  HTTP      746 HTTP/1.1 200 OK  (PNG)
```

5. HTTP Authentication

Q18: What is the server’s response (status code and phrase) in response to the initial HTTP GET message from your browser?

A18: 最初的状态码及短语是”401 Unauthorized”;

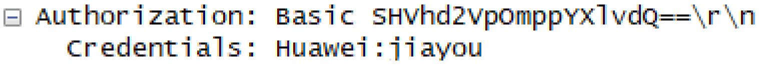


Filter: ip.addr == 128.119.245.12 && http
Expression... Clear Apply Save

```
No.  Time        Source          Destination     Protocol Length Info
  22 2.16287600  10.234.113.115  128.119.245.12  HTTP      506 GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1
  28 2.43295400  128.119.245.12  10.234.113.115  HTTP      771 HTTP/1.1 401 Unauthorized  (text/html)
```

Q19: When your browser’s sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?

A19: 新增了输入的 Username 和 Password;



```
⊟ Authorization: Basic SHVhd2VpOmppYXlvdQ==\r\n
    Credentials: Huawei:jiayou
```

六、结果及分析

1. The Basic HTTP GET/response interaction

在 Windows11 系统上，利用 Wireshark 软件抓包并分析 HTTP 协议，可通过实操熟悉了本的请求/响应交互，并结合课上学习的内容，可以进一步熟悉 HTTP 报文格式，比如我们可以通过报文获得客户端 IP 地址、服务器 IP 地址、最后修改时间、报文段和内容长度等信息；

2. The HTTP CONDITIONAL GET/response interaction

本地浏览器请求过一次服务器上的资源后，会将其缓存在本地，在一定时间周期内，当它再次请求已缓存的资源时，会验证服务器端的内容和本地缓存相比是否有修改，若没有，则服务器返回”304 Not Modified”，那么浏览器将会直接从本地缓存获取相应的内容；

3. Retrieving Long Documents

当客户端请求的内容较大时，服务器的响应会被分成多个 TCP 数据段单独进行传输，在接收完成后进行合并，得到最终的结果；

4. HTML Documents with Embedded Objects

若网页内容中含有多张图片时，那么客户端会多次请求，可能采取并行下载或串行下载这两种不同的方式，可以通过 Wireshark 软件具体分析；

5. HTTP Authentication

当网页需要验证用户权限时，用户第一次请求后，服务器不会直接返回内容，而是先返回” 401 Unauthorized”，待用户输入 Username 和 Password 进行验证成功后，服务器才会返回后续内容。

总的来讲，本次实验成功完成，熟悉了 HTTP 协议的相关内容。

七、上机实验总结

通过本次实验，我学习了如何在 Windows 系统上利用 Wireshark 软件对 http 协议进行抓包分析，实现了如下目标：

- 通过实操熟悉了基本的请求/响应交互;
- 结合课上学习的内容进一步熟悉了 HTTP 报文格式，可以通过报文获得客户端 IP 地址、服务器 IP 地址、最后修改时间、报文段和内容长度等信息;
- 请求大型 HTML 文件时，内容会被分成多个 TCP 段进行传输；
- 请求带有嵌入对象的 HTML 文件，如有多个图片资源时，有并行下载和串行下载两种模式，可以通过 Wireshark 软件分析出具体方式；
- 体验了 HTTP 身份验证和安全性，当网页需要验证用户权限时，响应报文会返回”401 Unauthorized”的状态码及短语，只有当用户通过验证之后，服务器才会返回后续内容。