中国科学技术大学计算机学院

计算机网络实验报告

实验二

利用 Wireshark 观察 http 报文

学　　号：PB17111568
姓　　名：郭雨轩
专　　业：计算机科学与技术
指导老师：张信明

中国科学技术大学计算机学院
2019 年 10 月 20 日

一、 实验目的

1、了解HTTP 协议的层次结构；
2、熟悉HTTP 协议中用于消息传输的数据结构；
3、掌握 HTTP 协议数据传输的原理。

二、 实验原理

Wireshark（前称 Ethereal）是一个网络封包分析软件。网络封包分析软件的功能是抓取网络封包,并尽可能显示出最为详细的网络封包资料。Wireshark 使用 WinPCAP 作为接口，直接与网卡进行数据报文交换，监听共享网络上传送的数据包，并不能对其进行修改或者控制。

本实验使用 Wireshark 抓取 Chrome 浏览器的在访问网页时发送和接收的数据包，对其进行分析。

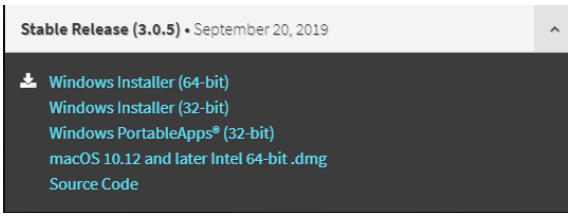
三、 实验环境

软件：Windows10-64bit，Wireshark，Chrome.
硬件：Intel Corei5-7300HQ，NVIDIA GTX1050Ti，16GiB RAM.

四、 实验过程

1、 Wireshark 安装

1） 访问 wireshark.org 得到了 wireshark 安装包

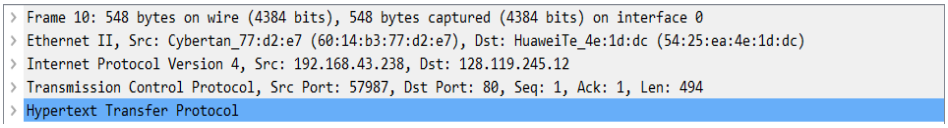


2） 双击打开 wireshark 安装包即可完成安装

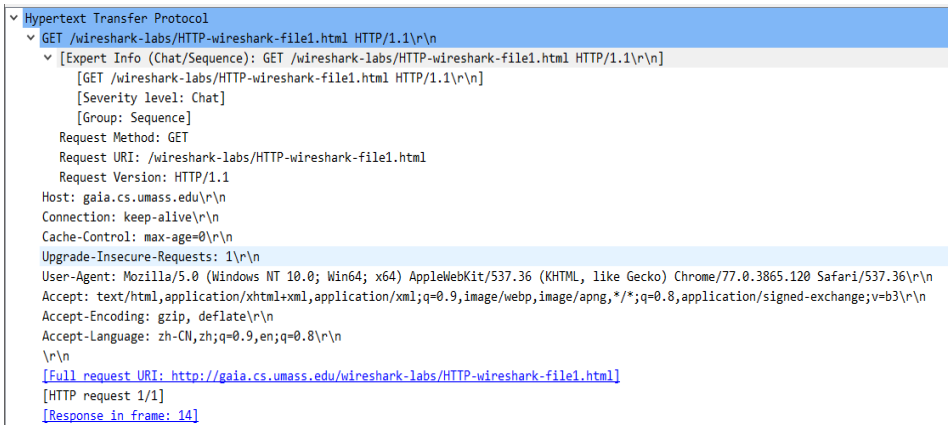
2、 实验过程

1） 实验一

1. 用请求的报文为例，可以看到请求的报文被分为 5 层，

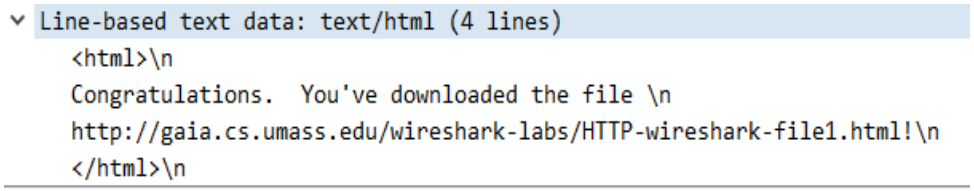


从下到上依次对应：应用层、传输层、网络层、链路层和物理层。



图中可以非常清楚的看到该报文格式与课本中给出的 HTTP 请求报文格式完全一致。

2. 在 HTTP 返回报文中，可以找到内容如图：



3. 我使用的 HTTP 版本为：

```
Request Version: HTTP/1.1
```

服务器使用的 HTTP 版本为：

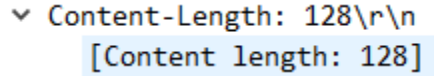
```
Response Version: HTTP/1.1
```

4. 我的 IP 地址为 192.168.43.238，服务器的 IP 地址为 128.119.245.12

5. 上一次更改的时间为：

```
Last-Modified: Sat, 19 Oct 2019 05:59:03 GMT\r\n
```

6. 长度为（单位 Byte）：



2）实验二

1. 找到的截图如下：

```
If-Modified-Since: Sun, 20 Oct 2019 05:59:02 GMT\r\n
```

2. 第一次显式的返回了文件的内容，返回的报文中有：

```
Line-based text data: text/html (10 lines)
    \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
```

第二次没有返回，因为没有上面的条目。

原因：因为第一次请求得到的文件的上一次改动时间没有发生变化，所以当用户在多次请求这个文件的时候，HTTP 的 GET 请求中会包含 IF-MODIFIED-SINCE 这个域，服务器若发现两个时间吻合则不会重新发送一次这个文件。

3）实验三

1. 发送了 1 个 GET。
2. 需要 4 个 TCP 信息段：

```
[4 Reassembled TCP Segments (4861 bytes): #164(1360), #165(1360), #167(1360), #168(781)]
  [Frame: 164, payload: 0-1359 (1360 bytes)]
  [Frame: 165, payload: 1360-2719 (1360 bytes)]
  [Frame: 167, payload: 2720-4079 (1360 bytes)]
  [Frame: 168, payload: 4080-4860 (781 bytes)]
  [Segment count: 4]
  [Reassembled TCP length: 4861]
  [Reassembled TCP Data: 485454502f312e3120323030204f4b0d0a446174653a204d…]
```

3. 如图：

```
HTTP/1.1 200 OK\r\n
  [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      [HTTP/1.1 200 OK\r\n]
      [Severity level: Chat]
      [Group: Sequence]
  Response Version: HTTP/1.1
  Status Code: 200
  [Status Code Description: OK]
  Response Phrase: OK
```

4. 报文中的数据被分为 4 个连续的 TCP 信息段传输：

147	2.774907	192.168.43.238	128.119.245.12	HTTP	522 GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
164	3.058876	128.119.245.12	192.168.43.238	HTTP	1414 HTTP/1.1 200 OK (text/html)
165	3.059090	128.119.245.12	192.168.43.238	HTTP	1414 Continuation
167	3.059283	128.119.245.12	192.168.43.238	HTTP	1414 Continuation
168	3.059489	128.119.245.12	192.168.43.238	HTTP	835 Continuation

4）实验四

1. 发送了 3 个请求信息，分别发送到了：

[Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html]

[Full request URI: http://gaia.cs.umass.edu/pearson.png]

[Full request URI: http://manic.cs.umass.edu/~kurose/cover_5th_ed.jpg]

2. 连续的下载两个图片，两次请求的时间戳不同。

5）实验五

1. 如图：

```
Status Code: 401
[Status Code Description: Unauthorized]
Response Phrase: Unauthorized
```

2. 如图：

```
Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=\r\n
   Credentials: wireshark-students:network
```

6）实验六

1. POST

对于科大邮箱服务，在登录时就会使用 POST 请求，在 POST 向浏览器发送的数据中，有（略去密码部分未截图）：

```
HTML Form URL Encoded: application/x-www-form-urlencoded
▾ Form item: "locale" = "zh_CN"
    Key: locale
    Value: zh_CN
▾ Form item: "uid" = "gyx_20170818"
    Key: uid
    Value: gyx_20170818
▾ Form item: "nodetect" = "false"
    Key: nodetect
    Value: false
▾ Form item: "domain" = "mail.ustc.edu.cn"
    Key: domain
    Value: mail.ustc.edu.cn
```

以上内容，包含了我登录校内邮箱的所有信息。POST 请求将我在浏览器中输入的用户名和密码等信息通过报文的形式传输给了邮箱的服务器。

2. GET

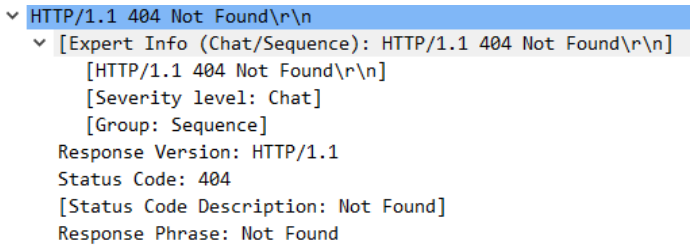
对于 GET 请求，以访问我的实验一搭建的个人主页为例。如图：

```
▾ Hypertext Transfer Protocol
  ▾ GET /~gyx_20170818/vendor/bootstrap/css/bootstrap.min.css HTTP/1.1\r\n
    › [Expert Info (Chat/Sequence): GET /~gyx_20170818/vendor/bootstrap/css/bootstrap.min.css HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /~gyx_20170818/vendor/bootstrap/css/bootstrap.min.css
      Request Version: HTTP/1.1
    Host: home.ustc.edu.cn\r\n
    Connection: keep-alive\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.120 Safari/537.36\r\n
    Accept: text/css,*/*;q=0.1\r\n
    Referer: http://home.ustc.edu.cn/~gyx_20170818/\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: zh-CN,zh;q=0.9,en;q=0.8\r\n
  › Cookie: _ga=GA1.3.1519688172.1560352713\r\n
    If-None-Match: "1abc17e-26074-d7e31ac0"\r\n
    If-Modified-Since: Sun, 29 Sep 2019 15:33:39 GMT\r\n
    \r\n
    [Full request URI: http://home.ustc.edu.cn/~gyx_20170818/vendor/bootstrap/css/bootstrap.min.css]
    [HTTP request 1/1]
    [Response in frame: 3236]
```

可以看到，GET 请求将要向服务器发送的数据会附在 URL 之后，对于中文等字符则先进行 base64 再请求。

7）实验七

故意将实验中给的网址输错一个字符，就会得到出错的网址。在返回的报文中，可以看到：



返回的状态码是 404，要访问的文件不存在。而对于返回的 html 文本，则清楚的显示 NOT FOUND：

```
Line-based text data: text/html (7 lines)
  <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">\n
  <html><head>\n
  <title>404 Not Found</title>\n
  </head><body>\n
  <h1>Not Found</h1>\n
  <p>The requested URL /wiresharklabs/protected_pages/HTTP-wireshark-file5.html was not found on this server.</p>\n
  </body></html>\n
```

五、 实验总结

1） HTTP 功能：Http 协议是一个建立在 TCP/IP 协议上的应用层规范是一个应用层的协议。它定义了与服务器进行交互的几种不同方法，使得应用程序和服务器之间可以进行通信。

2） 实验收获：

a) 我熟悉了 Wireshark 的使用，学会了通过 Wireshark 获得数据包。

b) 通过阅读浏览器发送和接收的 HTTP 报文，并比对书上的 HTTP 报文结构，我对报文的每个条目代表的信息有了深刻的了解。

c) 通过实验六对 GET 和 POST 两种方式的比较和搜集资料，我对这两种方法的差异有了深刻的了解，同时还对 GET 和 POST 的一些常见的误区有了认识。