School of Computer Science and Cybersecurity

CUC

Lab Report #

|  |  |
| --- | --- |
| Lab Name | Wireshark Lab8:HTTP |
| Course Name | Computer Networks |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | 赵婧宇 | Student ID | 201711123028 |

|  |  |
| --- | --- |
| Partners |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Date | 2019.06.20 | Lab Location | #48 |

|  |
| --- |
| Notes： |

**Section I Introduction**

Prepares the reader to understand the whole experiment.

|  |  |
| --- | --- |
| **Must Have:**  1. Clearly stated purpose of the experiment  2. Important background and/or theory | **May include:**  1. Description of specialized equipment  2. Justification of experiment's importance |

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, and HTTP authentication and security. Before beginning these labs, you might want to review the text.

**Section II Methods & Materials**

Can be lists or even "refer to lab manual" where appropriate.

**Section III Procedure & Results**

Describes ACTUAL process, especially changes from planned method.

|  |
| --- |
| * **number** and **title** tables and graphs correctly and clearly * draw attention to key points in tables or graphs with a sentence * provide sample calculation only * state key result in sentence form |

**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://eteaching.cuc.edu.cn/computernetworks/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**

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

* Start up the Wireshark packet sniffer

• Enter the following URL into your browser

<http://eteaching.cuc.edu.cn/computernetworks/Labs/HTTP-Wireshark-file2.html>

Your browser should display a very simple five-line HTML file.

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

• 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.

**3. Retrieving Long Documents**

• Start up your web browser, and make sure your browser’s cache is cleared, as discussed

above.

• Start up the Wireshark packet sniffer

• Enter the following URL into your browser

<http://eteaching.cuc.edu.cn/computernetworks/Labs/HTTP-Wireshark-file3.html>

Your browser should display the rather lengthy US Bill of Rights.

• Stop Wireshark packet capture, and enter “http” in the display-filter-specification window, so

that only captured HTTP messages will be displayed.

**4. HTML Documents with Embedded Objects**

• Start up your web browser, and make sure your browser’s cache is cleared, as discussed

above.

• Start up the Wireshark packet sniffer

• Enter the following URL into your browser

<http://eteaching.cuc.edu.cn/computernetworks/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. Your browser will have to retrieve these images from the indicated web sites. The upper image is retrieved from the web site of tsinghua.edu.cn. The lower image is stored at the local server.

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

**Section IV Discussion**

Answer the questions in the section [what to hand in] of the lab guide, includes two aspects:

|  |
| --- |
| 1. Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?   Answer: Yes, HTTP 1.1       1. What languages (if any) does your browser indicate that it can accept to the server?      1. What is the IP address of your computer? Of the eteaching.cuc.edu.cn server?      1. What is the status code returned from the server to your browser?     Answer: the status code is 200. It means succeed.   1. When was the HTML file that you are retrieving last modified at the server?      1. How many bytes of content are being returned to your browser?   Answer: 141 bytes     1. 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.   Answer: The HTTP request packet also contains Host field, connection field, Accept field, user-agent field, accept-encoding field and so on. There are also server fields, connection fields and so on in the HTTP response message.       1. 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?   Answer: No. This is the first time the browser requests the page.   1. Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?   Answer: The file content is returned because the return status code is 200. And returns the HTML file.     1. 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?   Answer: Yes. There's this row of fields. Information is the last time the file was requested.     1. 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. Answer: the status code is 304, phrase is Not Modified. The file content is not explicitly returned because the page content has not been modified since the last visit, and there is a cache locally of the last visit.      1. 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?   Answer: One HTTP GET request messages.   1. Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request?      1. What is the status code and phrase in the response?      1. How many data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights?   Answer: Four TCP segments.     1. How many HTTP GET request messages did your browser send? To which Internet addresses were these GET requests sent?   Answer: Three. The destination addresses are  202.205.18.2 202.205.28.2  124.17.26.243     1. Can you tell whether your browser downloaded the two images serially, or whether they were downloaded from the two web sites in parallel? Explain.   Answer: No. From time we can know it was downloaded one by one. |
|  |

**Section V Conclusion**

States what is known as a result of the experiment.

|  |  |
| --- | --- |
| **Must do:**  1. State what's known  2. Justify that statement | **May do:**  1. State significance of findings  2. Suggest further research |

Through experiments, I learned about basic GET/response interactions, HTTP message formats, retrieving large HTML files, and retrieving HTML files with embedded objects.