IT304
Lab4
Analysis of HTTP using Wireshark

1 The Basic HTTP GET/response interaction

Lets begin our exploration of HTTP by downloading a very simple HTML file. one that is very short, and contains no embedded objects. Do the following:

1.1 Exercise:

1. Start your web browser.
   Start the Wireshark tool, 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 are only interested in the HTTP protocol here, and don't want to see the clutter of all captured packets).

2. Begin Wireshark packet capture.

3. Enter the following to your browser.
   `http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html`

4. Stop Wireshark packet capture.

1.2 Questions:

By looking at the information in the HTTP GET and response messages, answer the following questions.

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

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

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

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

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

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

2 The HTTP CONDITIONAL GET/response in-teraction

2.1 Exercise:

1. Start your web browser, and make sure your browsers cache is cleared.

2. Start the Wireshark tool.

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.

2.2 Questions:

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?

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

3. 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?

4. 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.

3 Retrieving Long Documents

In our examples thus far, the documents retrieved have been simple and short HTML files. Lets next see what happens when we download a long HTML file. Do the following:

3.1 Exercise:

1. Start up your web browser, and make sure your browsers cache is cleared, as discussed above.
2. Start up the Wireshark packet.
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.

3.2 Questions:

1. How many HTTP GET request messages were sent by your browser?
2. How many data-containing TCP segments were needed to carry the single HTTP response?
3. What is the status code and phrase associated with the response to the HTTP GET request?

4 HTML Documents with Embedded Objects

Now that weve seen how Wireshark displays the captured packet tranfer c for large HTML files, we can look at what happens when your browser downloads a file with embedded objects, i.e., a file that includes other objects (in the example below, image files) that are stored on another server(s). Do the following:

4.1 Exercise:

1. Start up your web browser, and make sure your browsers cache is cleared, as discussed above.
2. Start up the Wireshark packet.
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.

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

4.2 Questions:

1. How many HTTP GET request messages were sent by your browser? To which Internet addresses were these GET requests sent?

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

5 HTTP Authentication

Finally, lets try visiting a web site that is password-protected and examine the sequence of HTTP message exchanged for such a site. The URL `http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html` is password protected. The user-name is "wireshark-students" (without the quotes), and the password is "network" (again, without the quotes). So lets access this secure password-protected site. Do the following:

5.1 Exercise:

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

2. Start up the Wireshark packet.

3. Enter the following URL into your browser.
`http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html`

4. Type the requested user name and password into the pop up box.

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.

5.2 Questions:

1. What is the servers response (status code and phrase) in response to the initial HTTP GET message from your browser?

2. When your browsers sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?