Lab Exercise 3: HTTP & DNS

Exercise 3: Using Wireshark to understand basic HTTP request/response messages

Question 1: What is the status code and phrase returned from the server to the client browser?

Solution 1 :

Status Code : HTTP/1.1 200
Phrase : OK

Question 2: When was the HTML file that the browser is retrieving last modified at the server? Does the response also contain a DATE header? How are these two fields different?

Solution 2 :

Date : Tuesday ,23 September 2003 ,05:29:50
Last Modified : Tuesday ,23 September 2003 ,05:29:00

Date indicates the time and date when the HTTP response was created and sent by the server. It is the time when the server retrieves the object from its file system, inserts the object into the response message, and sends the response message.

The Last-Modified header line indicates the time and date when the object was created or last modified. It is critical for object caching, both in the local client and in network cache servers.

Question 3: Is the connection established between the browser and the server persistent or non-persistent? How can you infer this?

Solution 3 :

Both the client and server are configured for using persistent HTTP connections. Looking at the Connection: header in both the request and response. Connection: keep-alive, which means it is using single TCP connection to send and receive multiple HTTP requests/responses.

<u>Question 4</u>: How many bytes of content are being returned to the browser?

<u>Solution 4</u> :

73 bytes of HTML data is being returned to the browser.

<u>Question 5</u>: What is the data contained inside the HTTP response packet?

<u>Solution 5</u> :

The data in the HTTP response is,

<html>
Congratulations. You've downloaded the file lab2-1.html!
<html>

Exercise 4: Using Wireshark to understand the HTTP CONDITIONAL GET/response interaction

Question 1: Inspect the contents of the first HTTP GET request from the browser to the server. Do you see an "IF-MODIFIED-SINCE" line in the HTTP GET?

Solution 1 :

No, there is no "IF-MODIFIED-SINCE" line in the HTTP GET.

Question 2: Does the response indicate the last time that the requested file was modified?

Solution 2 :

Yes, the response does indicate the last time(for the second one) that the requested file was modified Tuesday, 23 Sep 2003 05:35:50 GMT.

Question 3: Now inspect the contents of the second HTTP GET request from the browser to the server. Do you see an "IF-MODIFIED-SINCE:" and "IF-NONE-MATCH" lines in the HTTP GET? If so, what information is contained in these header lines?

Solution 3 :

Yes, there is "IF-MODIFIED-SINCE:" and "IF-NONE-MATCH" lines in the HTTP GET.
It shows the value Tuesday, 23 Sep 2003 05:35:00 GMT.
If-None-Match: "1bfef-173-8f4ae900"

Question 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.

Solution 4 :

Status Code : 304
Phrase = "Not Modified"

The server did not return the body contents of the file. This is due to the fact that the GET request is only conditional. The server will only send back the request resource if only it has been last modified after a given date. If the request has not been modified since, the response will be “304 status” without any body. This will save resources as the party making the GET request will not have to download the page again since nothing has been changed.

Question 5: What is the value of the Etag field in the 2nd response message and how it is used? Has this value changed since the 1st response message was received?

Solution 5 :

ETag (first response): “1bfef-173-8f4ae900”

ETag (second response): "1bfef-173-8f4ae900"

The value of Etag for the first and second response is the same.

The Etag HTTP response header is an identifier for a specific version of a resource. It allows caches to be more efficient, and saves bandwidth, as a web server does not need to send a full response if the content has not changed. On the other side, if the content has changed, Etags are useful to help prevent simultaneous updates of a resource from overwriting each other.If the resource at a given URL changes, a new Etag value must be generated.