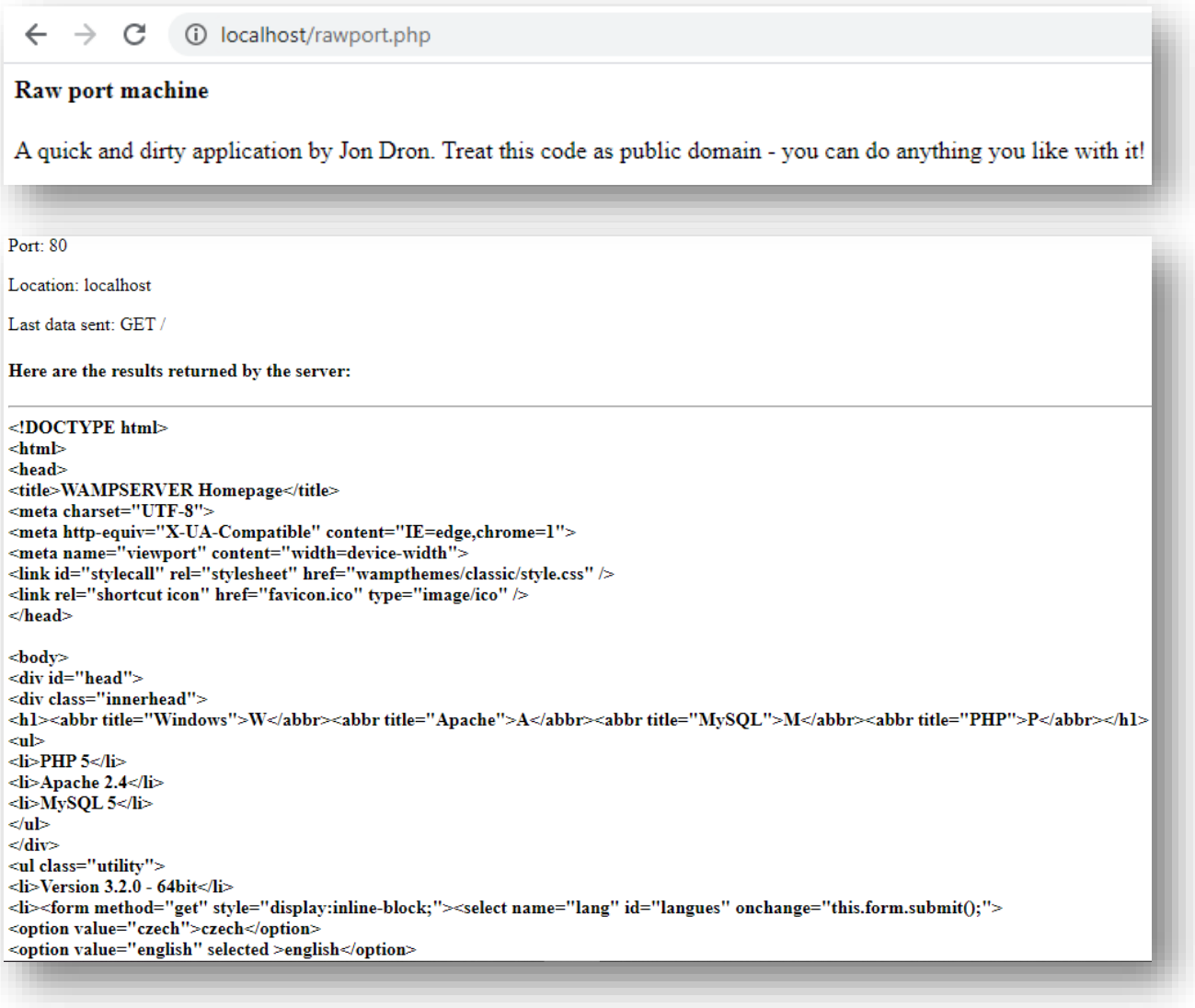
1. Using HTTP 0.9 – the simplest possible connection

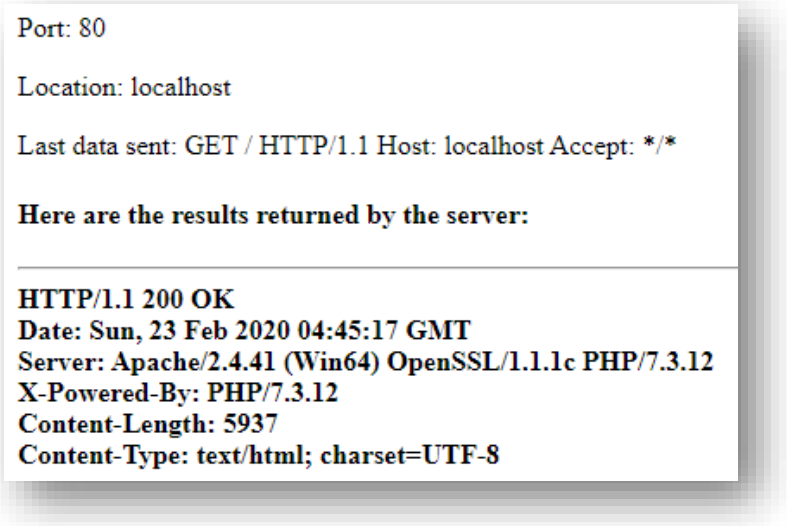
I used the default unencrypted port “localhost”, which points to the “c:/wamp64/www” directory where the “rawport.php” file is located.



The result returned was basically the source code of the “index.php” file located in my root directory which http://localhost points to and displays the WAMP server configuration. This is simply because the HTTP 0.9 that we are connecting to is a simple client-server protocol that has no HTTP headers. Hence, there are no status or error codes and only hypertext (HTML) can be returned by the server to the client.

If I were a web browser, I would include metadata about requests from the client and response from the server. Nevertheless, I would not be able to display pictures because HTTP 0.9’s lack of headers means that pictures cannot be transmitted. The connection is terminated immediately after the HTML response is sent, so there is no way to avoid getting the whole file again as a new connection has to be established. Also, there is no cache support in HTTP 0.9.

2. Using HTTP 1.1

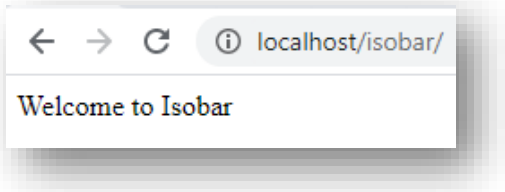


In HTTP 0.9, a connection needs to be opened and then closed for each HTTP GET request-response. HTTP 1.1 gives us the advantage of reusability by allowing us to reuse the same open connection for multiple request-response cycles. We press return twice because HTTP 1.1 keeps the connection open to allow extra commands to be sent, each of which is separated by a single carriage return. A blank line on its own signifies that the connection is completed as said in the Unit 6 self-test quiz.

The difference noticed in the output is the HTTP headers that are displayed. The Content-Type header provides the ability to transmit files other than plain HTML files. A status code is also included allowing the browser to understand the success or failure of the request and to adapt its behavior in consequence.

3. Redirection and default document

A new directory called “isobar” was created in the root directory with an “index.php” file placed in it. After entering http://localhost/isobar into the browser, the browser automatically adds a trailing slash (/).



However, after inputting the necessary information in the raw port machine, we get a 301 error telling us that the document has moved to http://localhost/isobar/.

Port: 80

Location: localhost

Last data sent: GET /isobar HTTP/1.1 Host: localhost Accept: */*

Here are the results returned by the server:

HTTP/1.1 301 Moved Permanently
Date: Sun, 23 Feb 2020 05:56:40 GMT
Server: Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.3.12
Location: http://localhost/isobar/
Content-Length: 332
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
** The document has moved here. **
<hr>
<address>Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.3.12 Server at localhost Port 80</address>
</body></html>

So, by adding a trailing slash, we get:

Port: 80

Location: localhost

Last data sent: GET /isobar/ HTTP/1.1 Host: localhost Accept: */*

Here are the results returned by the server:

HTTP/1.1 200 OK
Date: Sun, 23 Feb 2020 06:11:49 GMT
Server: Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.3.12
X-Powered-By: PHP/7.3.12
Content-Length: 126
Content-Type: text/html; charset=UTF-8

<!DOCTYPE HTML>
<html lan="en">
<head>

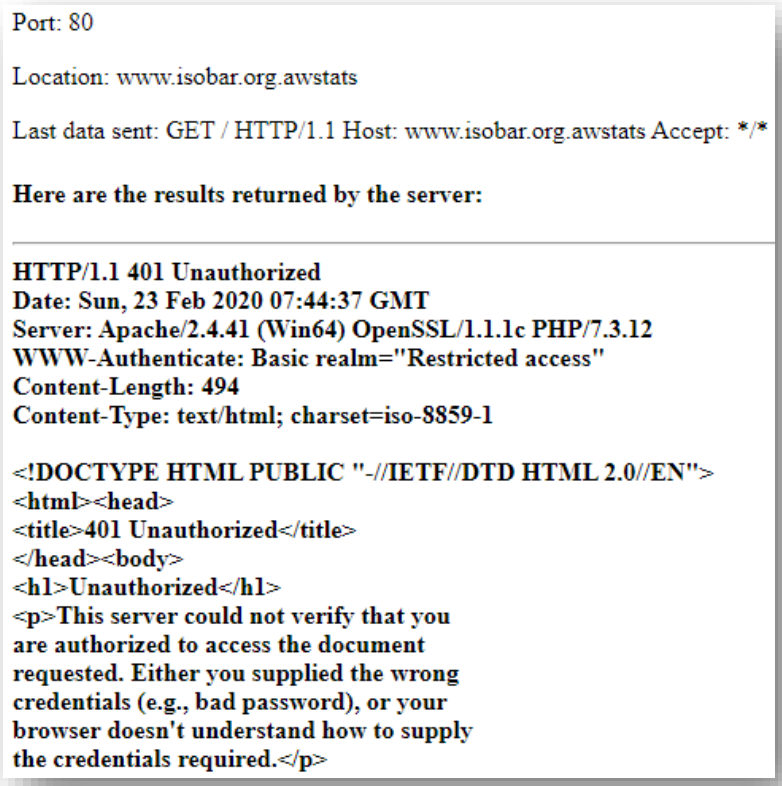
</head>

<body>
Welcome to Isobar </body>

</html>

4. Security

In TME5, I set a virtual host that pointed directly to the authenticated directory. So, the only way task 5 works is by using "/ HTTP/1.1" in the location field instead of "/protected /HTTP/1.1" I tried connecting to http://www.isobar.org.awstats/ but got a "401 Unauthorized" error because the directory is authenticated. After including the encoded "username:password" combo as part of the GET request, we get a success.



Port: 80

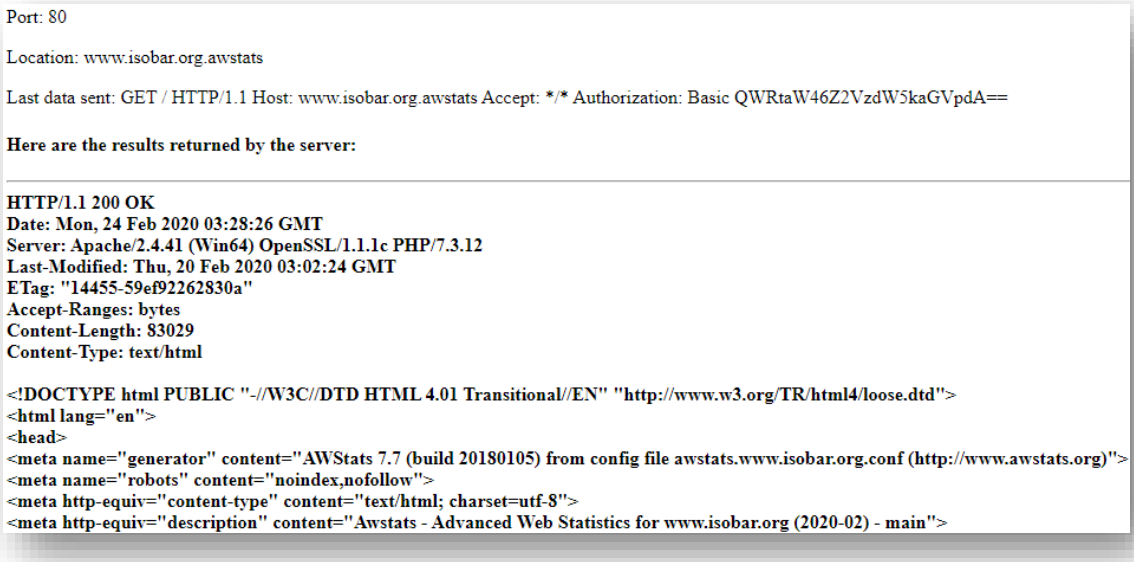
Location: www.isobar.org.awstats

Last data sent: GET / HTTP/1.1 Host: www.isobar.org.awstats Accept: */*

Here are the results returned by the server:

HTTP/1.1 401 Unauthorized
Date: Sun, 23 Feb 2020 07:44:37 GMT
Server: Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.3.12
WWW-Authenticate: Basic realm="Restricted access"
Content-Length: 494
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>401 Unauthorized</title>
</head><body>
<h1>Unauthorized</h1>
** This server could not verify that you**
are authorized to access the document
requested. Either you supplied the wrong
credentials (e.g., bad password), or your
browser doesn't understand how to supply
**the credentials required. **



Port: 80

Location: www.isobar.org.awstats

Last data sent: GET / HTTP/1.1 Host: www.isobar.org.awstats Accept: */* Authorization: Basic QWRtaW46Z2VzdW5kaGVpdA==

Here are the results returned by the server:

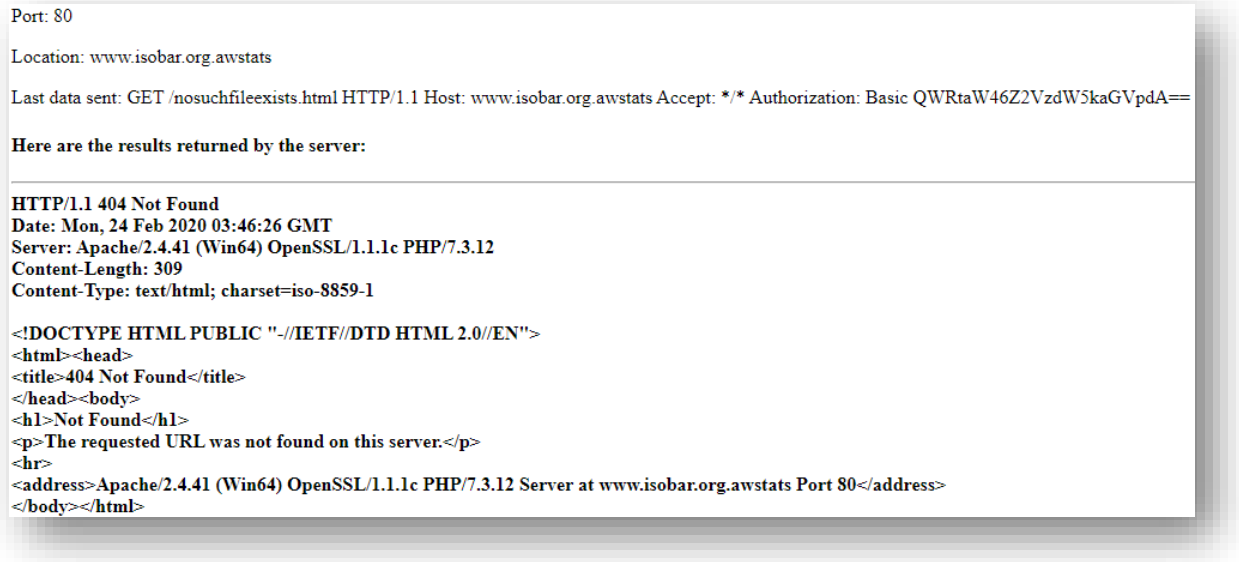
HTTP/1.1 200 OK
Date: Mon, 24 Feb 2020 03:28:26 GMT
Server: Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.3.12
Last-Modified: Thu, 20 Feb 2020 03:02:24 GMT
ETag: "14455-59ef92262830a"
Accept-Ranges: bytes
Content-Length: 83029
Content-Type: text/html

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html lang="en">
<head>
<meta name="generator" content="AWStats 7.7 (build 20180105) from config file awstats.www.isobar.org.conf (http://www.awstats.org)">
<meta name="robots" content="noindex,nofollow">
<meta http-equiv="content-type" content="text/html; charset=utf-8">
<meta http-equiv="description" content="Awstats - Advanced Web Statistics for www.isobar.org (2020-02) - main">

The reason why HTTP requires the username and password to be encoded is to escape special characters by encoding them in base 64, making them HTTP compatible.

5. Non-existent Pages

I included the “nosuchfileexist.html” text to the GET request with no authorization and got a “401 Unauthorized” error. However, after adding the encoded “username:password” combo, I got the expected “404 Not Found” error.



Port: 80

Location: www.isobar.org.awstats

Last data sent: GET /nosuchfileexists.html HTTP/1.1 Host: www.isobar.org.awstats Accept: */* Authorization: Basic QWRtaW46Z2VzdW5kaGVpdA==

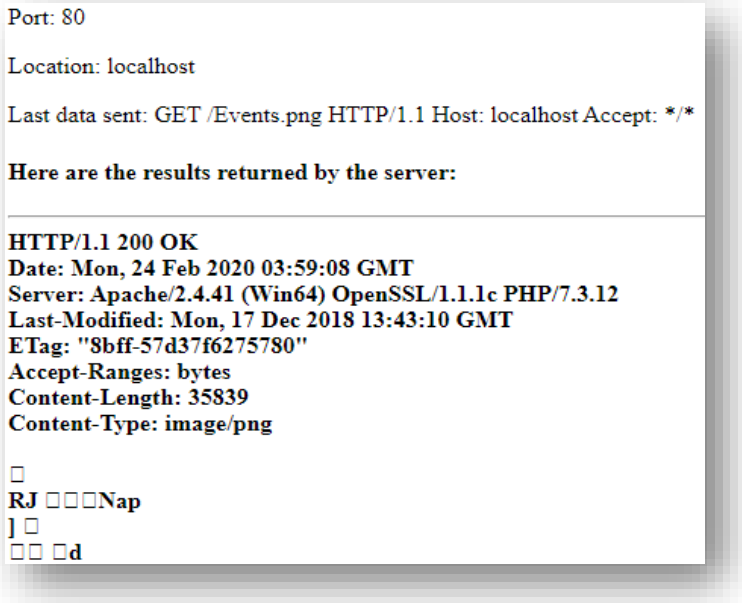
Here are the results returned by the server:

HTTP/1.1 404 Not Found
Date: Mon, 24 Feb 2020 03:46:26 GMT
Server: Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.3.12
Content-Length: 309
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
** The requested URL was not found on this server. **
<hr>
<address>Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.3.12 Server at www.isobar.org.awstats Port 80</address>
</body></html>

6. Getting pictures

I switched my host back to “localhost” and placed an “Events.png” image in the root directory. I attempted to access it through the raw port machine, and I got a “200 OK” success.



Port: 80

Location: localhost

Last data sent: GET /Events.png HTTP/1.1 Host: localhost Accept: */*

Here are the results returned by the server:

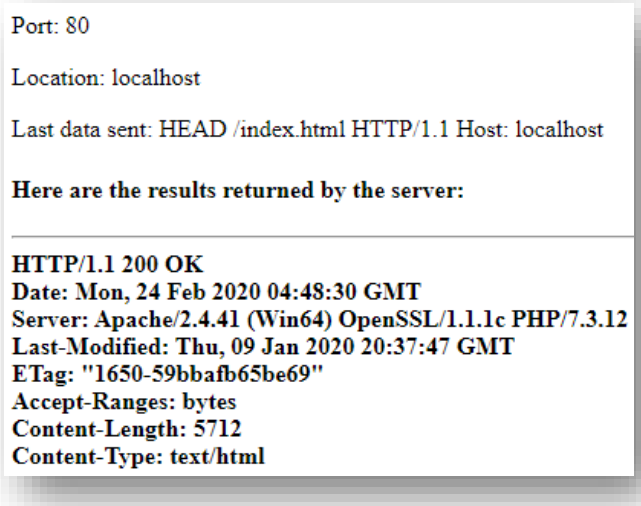
HTTP/1.1 200 OK
Date: Mon, 24 Feb 2020 03:59:08 GMT
Server: Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.3.12
Last-Modified: Mon, 17 Dec 2018 13:43:10 GMT
ETag: "8bff-57d37f6275780"
Accept-Ranges: bytes
Content-Length: 35839
Content-Type: image/png

□
RJ □□□Nap
] □
□□ □d

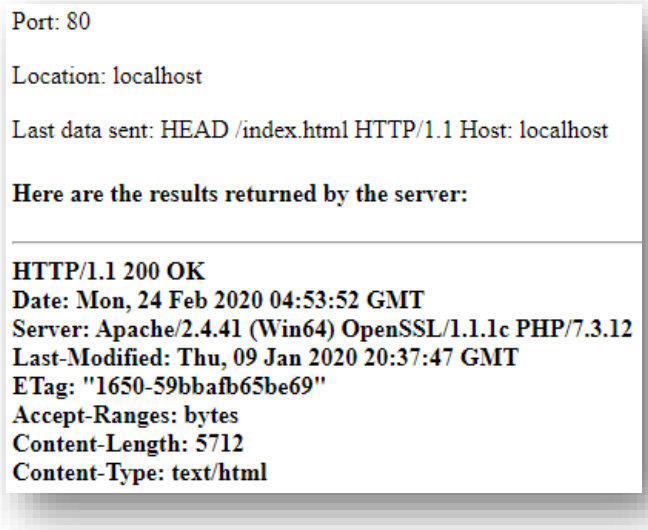
I changed the “Accept” from “*/*” to “text/*” and I got the same result, which means that there is something wrong with my web server's configuration. I went to the “C:\wamp64\bin\apache\apache2.4.41\conf” directory to include a “text/*” line in the “mime.types”. I restarted my server and ran the raw port machine and it still didn’t work. As said in the hint-and-tips, a 406 response is expected under normal conditions but apache’s unconfigurable “mod_negotiation” module is responsible for bypassing the “Accept” instruction.

7. Caching

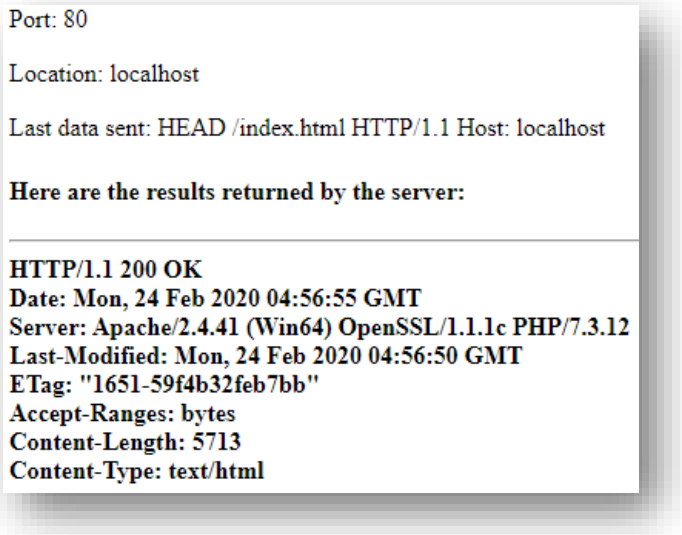
Using the HEAD command for caching, we get an E-TAG in our result.



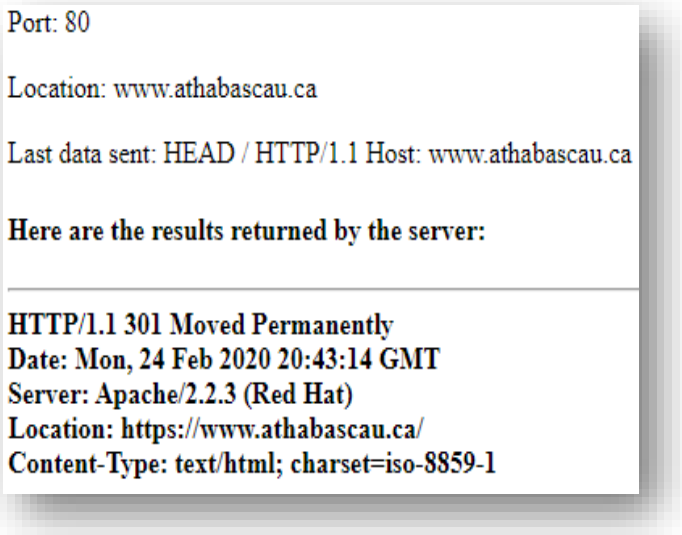
After re-submitting the command, we get the same E-tag with a new timestamp.



However, after editing the file by just adding “space”, we get a new E-tag.

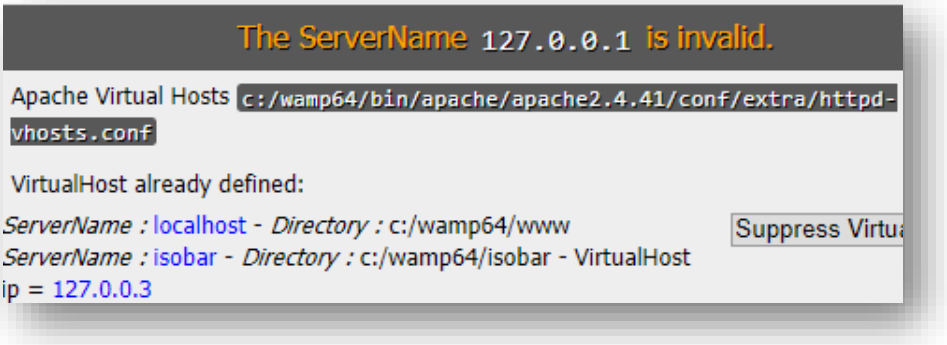


The name and the version of the webserver for www.athabasca.ca is Apache/2.2.3 (Red Hat) as shown below:



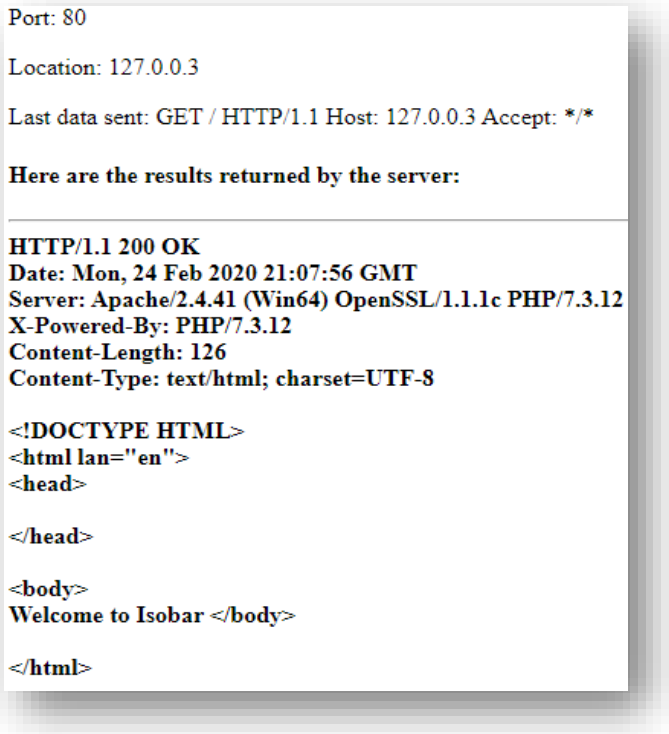
8. What is this “host” thing?

In TME5, my WAMP server did not allow me to create a new virtual host named “127.0.0.1”.

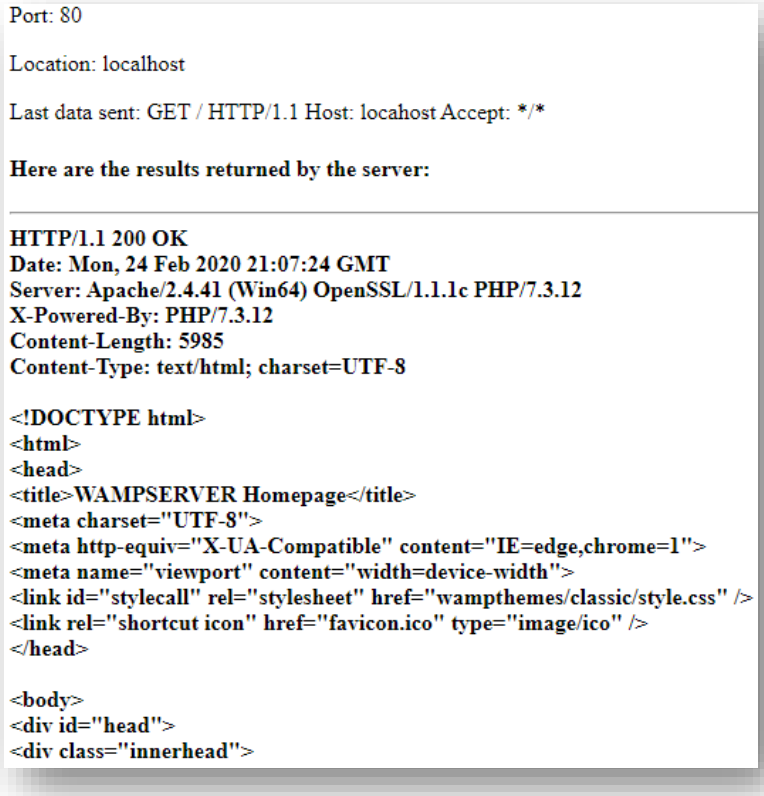


In TME6, I realized that on default, “127.0.0.1” not only refers to the same server but also the same root directory. So, I created a new virtual host “127.0.0.3” that had the root directory “C://wamp64/isobar”.

127.0.0.3 produced:

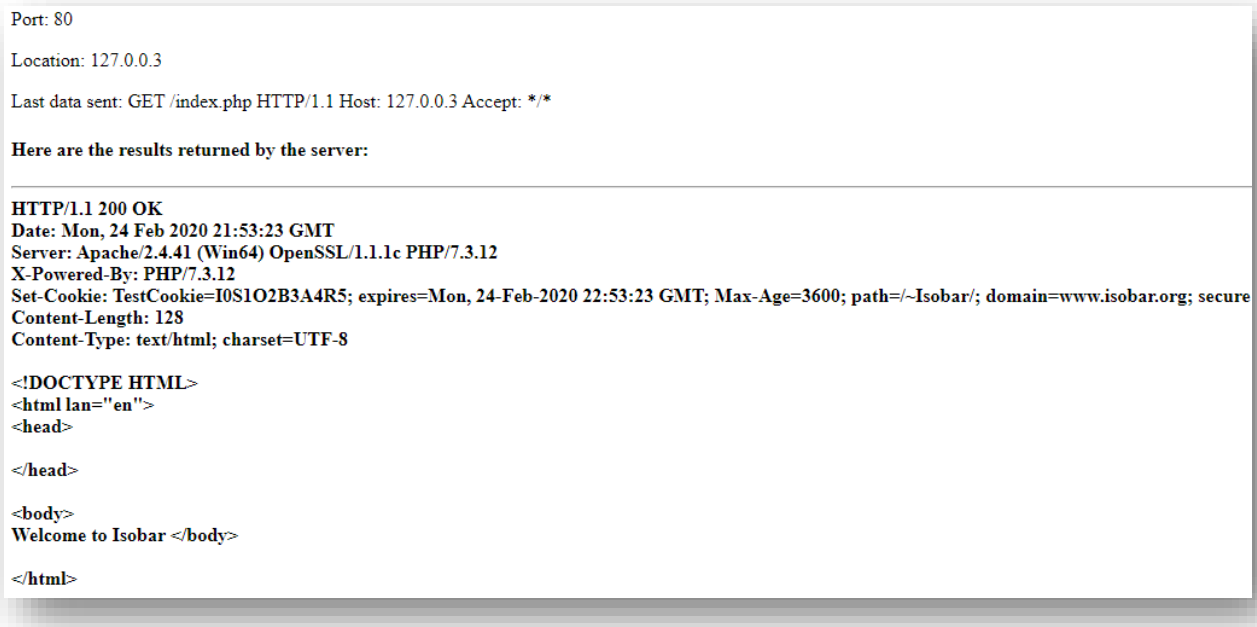


Localhost produced:



9. Cookies and a bit more redirection

I set a cookie in my “index.php” file located in the “C://wamp64/isobar” directory and the cookie was displayed in the HTTP header.



References

Gamage, T. A. (2017). *Evolution of HTTP — HTTP/0.9, HTTP/1.0, HTTP/1.1, Keep-Alive, Upgrade, and HTTPS*. Retrieved from Medium: https://medium.com/platform-engineer/evolution-of-http-69cfe6531ba0

MDN Contributors. (n.d.). *Evolution of HTTP*. Retrieved from MDN Web Docs: Mozilla: https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/Evolution_of_HTTP

PHP Manual. (n.d.). *Setcookie: Example #1 setcookie() send example*. Retrieved from PHP Documentation: https://www.php.net/manual/en/function.setcookie.php