

## CSE 4344 Project 1 Webserver Project Writeup

### About:

This program is written in python 3 and developed/tested in a Linux Environment. The web server has been tested on Firefox and Google Chrome.

### To run:

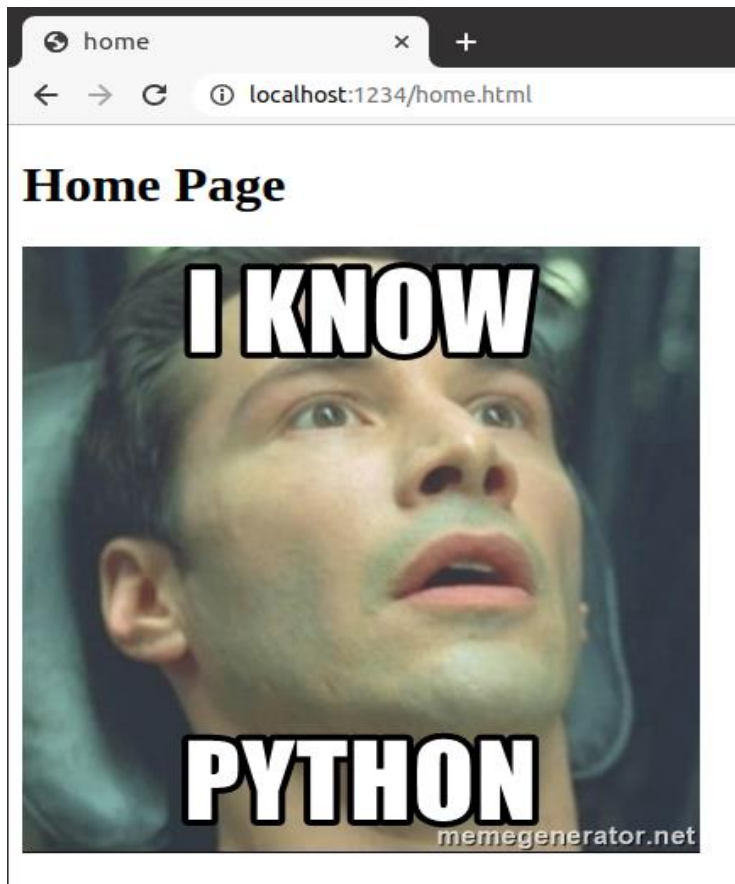
There is no need to pip/install packages to run this program. It runs entirely on the default modules. Open a terminal and run the following command:

**# python3 start\_server.py**

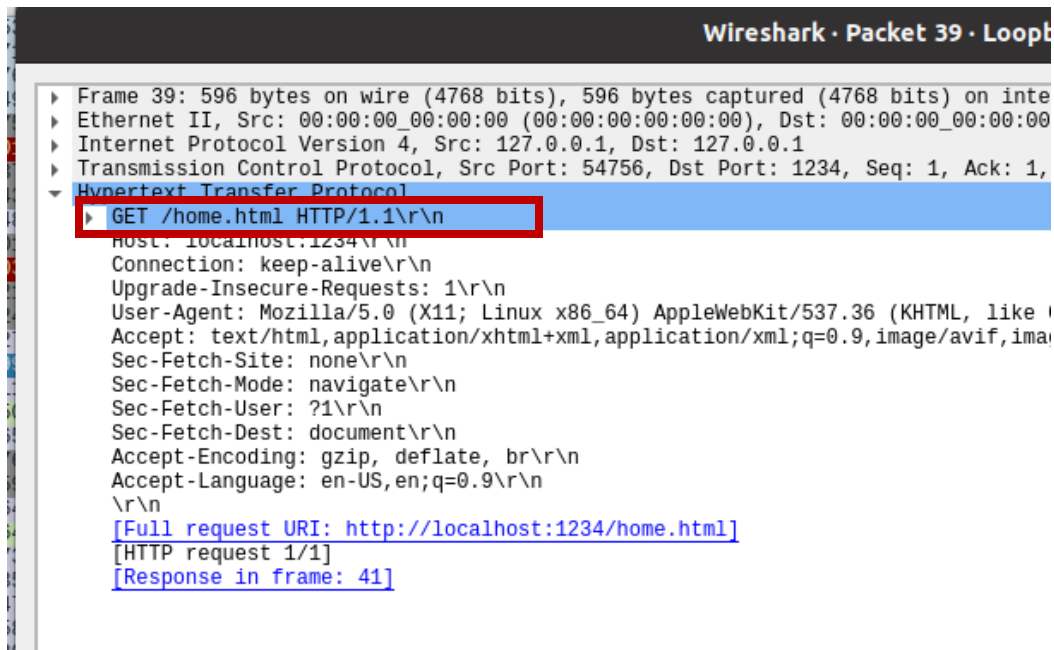
Use ctrl+c in the terminal to stop the server.

### 200 Ok

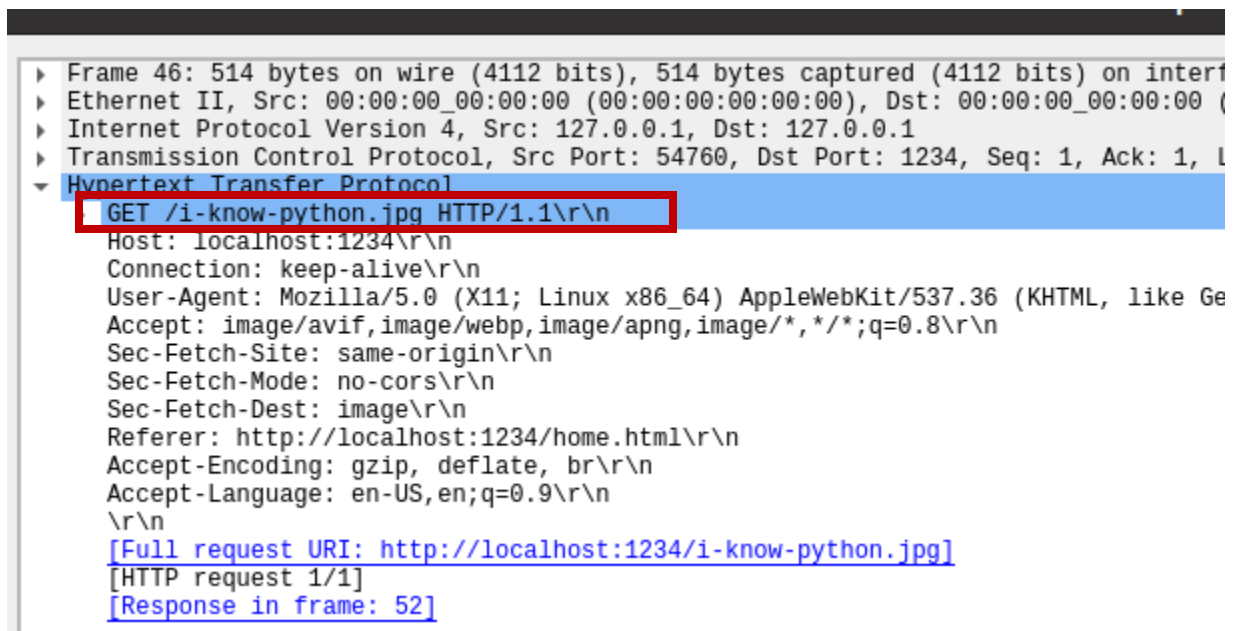
To demonstrate the 200 response go start the server using the command above and go to the url in the web browser: <http://localhost:1234/home.html>



Home.html get request from the web browser:



i-know-python.jpg get request from the browser:



## Server response for home.html:

Wireshark · Packet 41 · Loopback: lo

▶ Frame 41: 549 bytes on wire (4392 bits), 549 bytes captured (4392 bits) on interface lo  
▶ Ethernet II, Src: 00:00:00\_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00\_00:00:00 (00:00:00:00:00:00)  
▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1  
▶ Transmission Control Protocol, Src Port: 1234, Dst Port: 54756, Seq: 1, Ack: 531, Len:

▼ Hypertext Transfer Protocol

▶ HTTP/1.1 200 OK\n

▶ HTTP/1.1 200 OK\n

Date: Fri, 02 Oct 2020 20:30:00 GMT\n

Server: Michaels-Python-Server (custom)\n

Connection: Closed\n

▶ Content-Length: 266\n

Last-Modified: Wed, 30 Sep 2020 01:01:52 GMT\n

Content-Type: text/html\n

\n

[HTTP response 1/1]  
[Time since request: 0.000851648 seconds]  
[\[Request in frame: 39\]](#)  
[Request URI: http://localhost:1234/home.html]  
File Data: 266 bytes

▼ Line-based text data: text/html (11 lines)

```
<!DOCTYPE html>\n<html>\n  <head>\n    \n    \t<link rel="icon" type="image/x-icon" href="python-icon.ico">\n    <title>home</title>\n  </head>\n  <body>\n    <h1>Home Page</h1>\n    \n  </body>\n</html>
```

## Server response for i-know-python.jpg:

Wireshark · Packet 52 · Loopback: lo

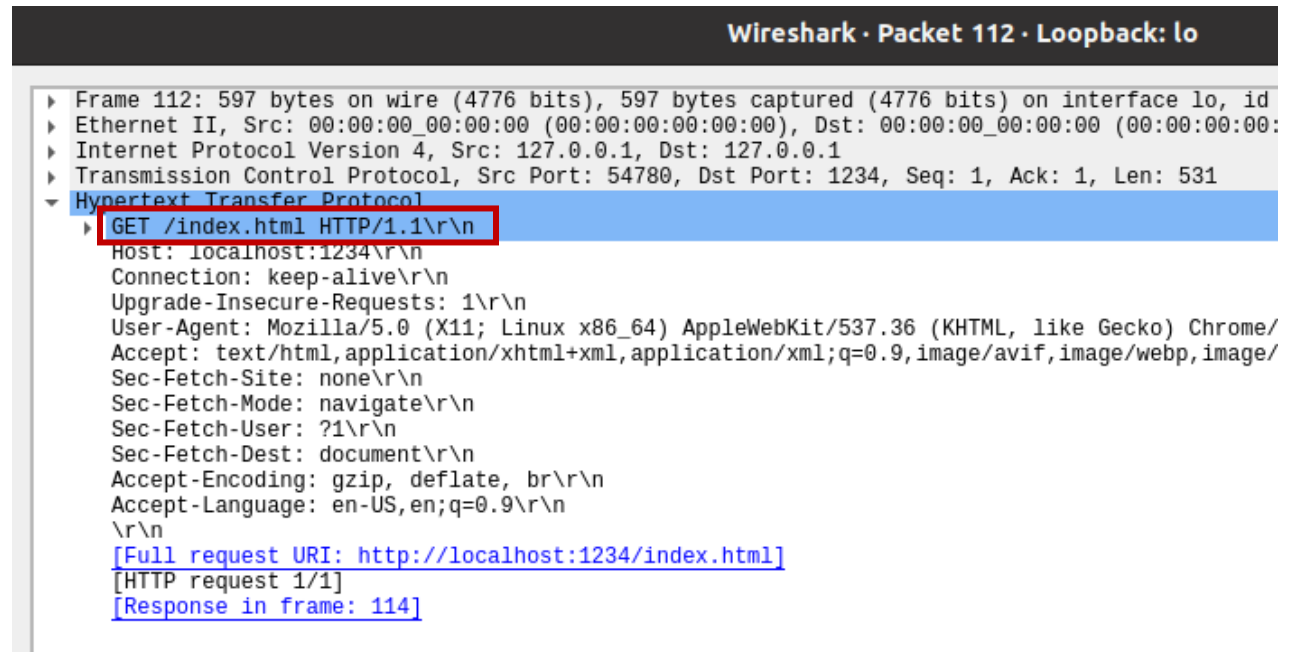
- ▶ Frame 52: 25418 bytes on wire (203344 bits), 25418 bytes captured (203344 bits) on interface lo, id 0
- ▶ Ethernet II, Src: 00:00:00\_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00\_00:00:00 (00:00:00:00:00:00)
- ▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
- ▶ Transmission Control Protocol, Src Port: 1234, Dst Port: 54760, Seq: 65537, Ack: 449, Len: 25352
- ▶ [3 Reassembled TCP Segments (90888 bytes): #48(32768), #50(32768), #52(25352)]
- ▼ Hypertext Transfer Protocol
  - ▶ HTTP/1.1 200 OK
  - ▶ HTTP/1.1 200 OK
  - Date: Fri, 02 Oct 2020 20:30:00 GMT
  - Server: Michaels-Python-Server (custom)
  - Connection: Closed
  - Content-Length: 90669
  - Last-Modified: Wed, 20 Sep 2020 00:25:18 GMT
  - Content-Type: image/jpg
  - sh
  - [HTTP response 1/1]
  - [Time since request: 0.008381564 seconds]
  - [Request in frame: 461]
  - [Request URI: http://localhost:1234/i-know-python.jpg]
  - File Data: 90669 bytes
- ▼ JPEG File Interchange Format
  - Marker: Start of Image (0xFFD8)
  - ▶ Marker segment: Reserved for application segments - 0 (0xFFE0)
  - ▶ Marker segment: Define quantization table(s) (0xFFDB)
  - ▶ Marker segment: Define quantization table(s) (0xFFDB)
  - ▶ Start of Frame header: Start of Frame (non-differential, Huffman coding) - Baseline DCT (0xFFC0)
  - ▶ Marker segment: Define Huffman table(s) (0xFFC4)
  - ▶ Marker segment: Define Huffman table(s) (0xFFC4)
  - ▶ Marker segment: Define Huffman table(s) (0xFFC4)
  - ▶ Marker segment: Define Huffman table(s) (0xFFC4)
  - ▶ Start of Segment header: Start of Scan (0xFFDA)
  - Entropy-coded segment (dissection is not yet implemented): fc2ebe433191b90707a73d4f7fc735c3dcda980ef7c92ecc...
  - Marker: End of Image (0xFFD9)

## 301 Moved Permanently

To demonstrate a 301 response go to the browser and type: `http://localhost:1234/index.html`

This will redirect to the `home.html` page as shown previously. The screenshots are displayed below.

**index.html get request from the web browser:**



The image shows a Wireshark packet capture window titled "Wireshark · Packet 112 · Loopback: lo". The packet list on the left shows "Frame 112: 597 bytes on wire (4776 bits), 597 bytes captured (4776 bits) on interface lo, id 1". The packet details pane shows the following layers:

- Ethernet II, Src: 00:00:00\_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00\_00:00:00 (00:00:00:00:00:00)
- Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
- Transmission Control Protocol, Src Port: 54780, Dst Port: 1234, Seq: 1, Ack: 1, Len: 531
- Hypertext Transfer Protocol

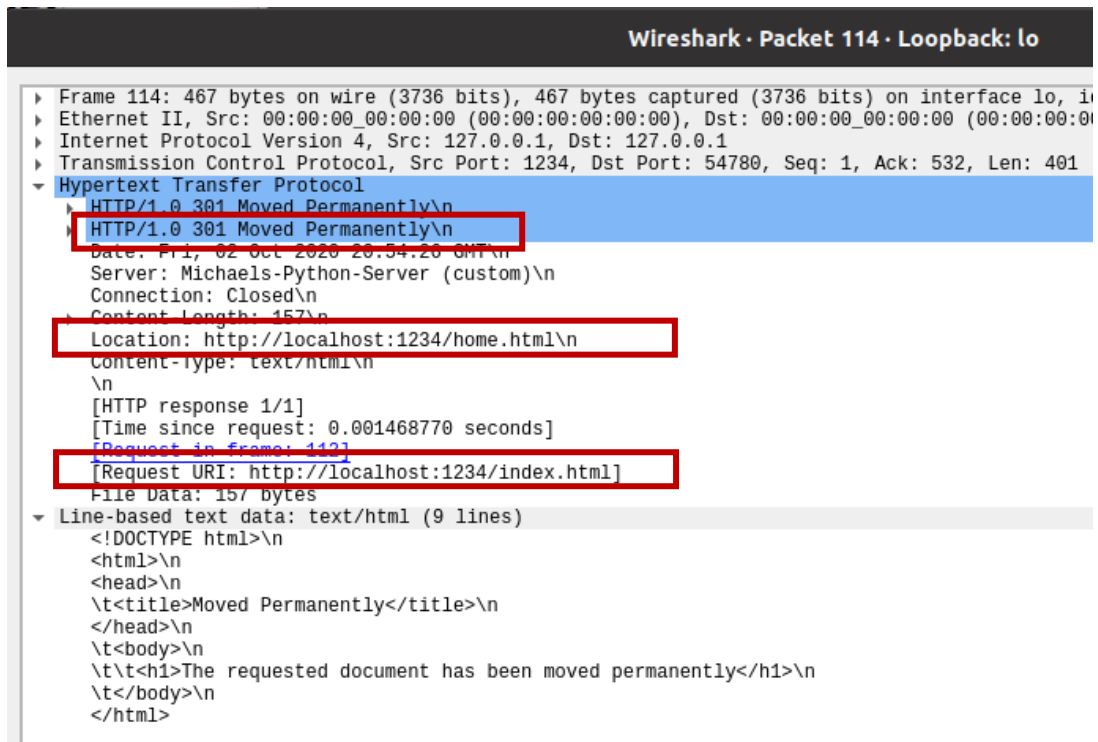
The Hypertext Transfer Protocol layer is expanded, showing the following fields:

- GET /index.html HTTP/1.1\r\n
- Host: localhost:1234\r\n
- Connection: keep-alive\r\n
- Upgrade-Insecure-Requests: 1\r\n
- User-Agent: Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/
- Sec-Fetch-Site: none\r\n
- Sec-Fetch-Mode: navigate\r\n
- Sec-Fetch-User: ?1\r\n
- Sec-Fetch-Dest: document\r\n
- Accept-Encoding: gzip, deflate, br\r\n
- Accept-Language: en-US,en;q=0.9\r\n
- \r\n

Below the fields, there are three links:

- [Full request URI: <http://localhost:1234/index.html>]
- [HTTP request 1/1]
- [Response in frame: 114]

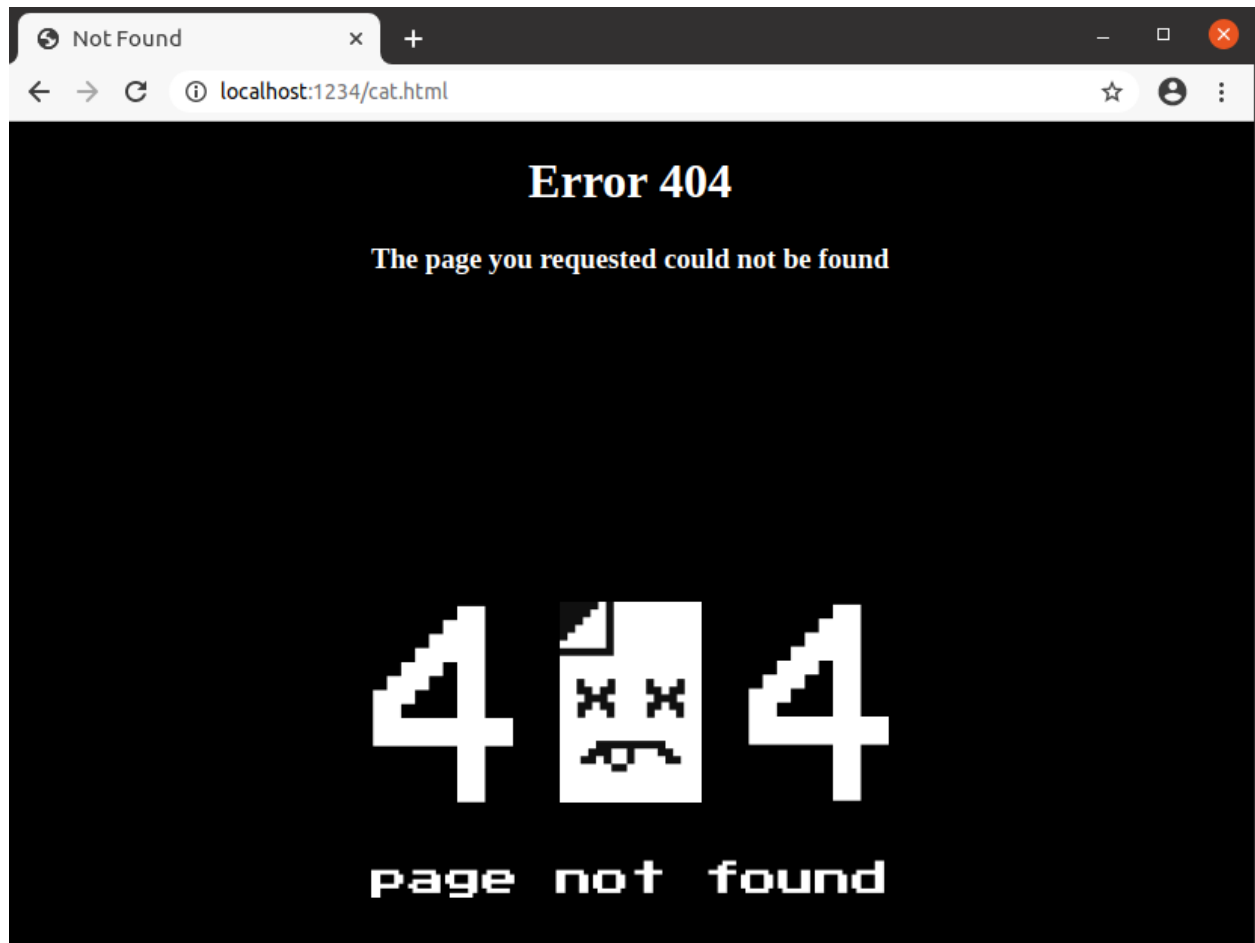
### 301 response from the server:



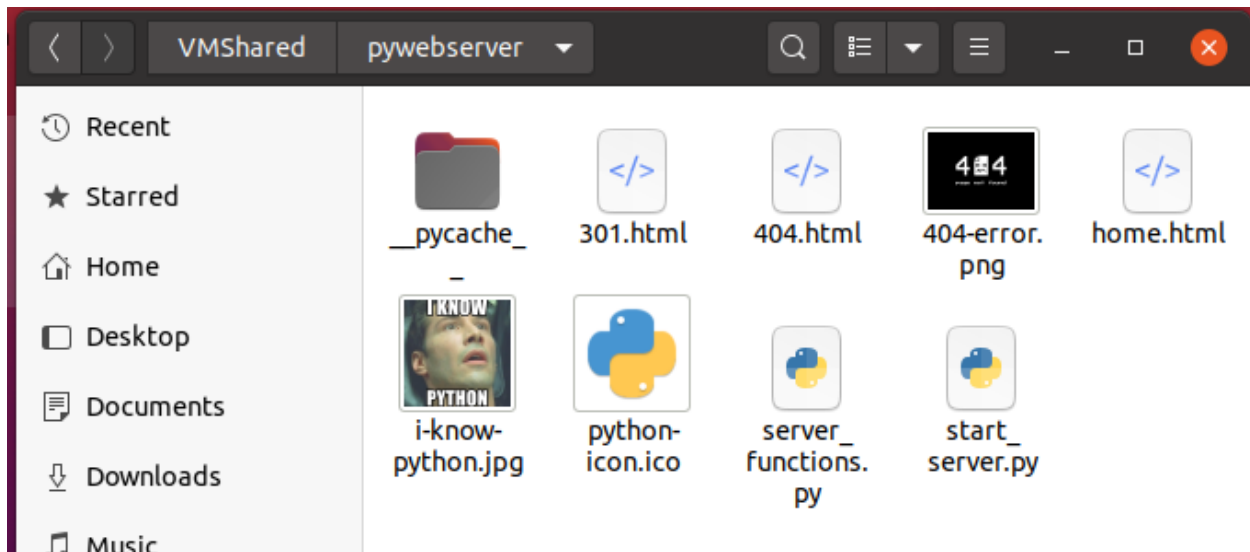
This redirects back to home.html as shown before.

## 404 Page Not Found

For the 404 response, I created an html file called 404.html. This file displays some text and an image to notify the user that the page they requested could not be found:

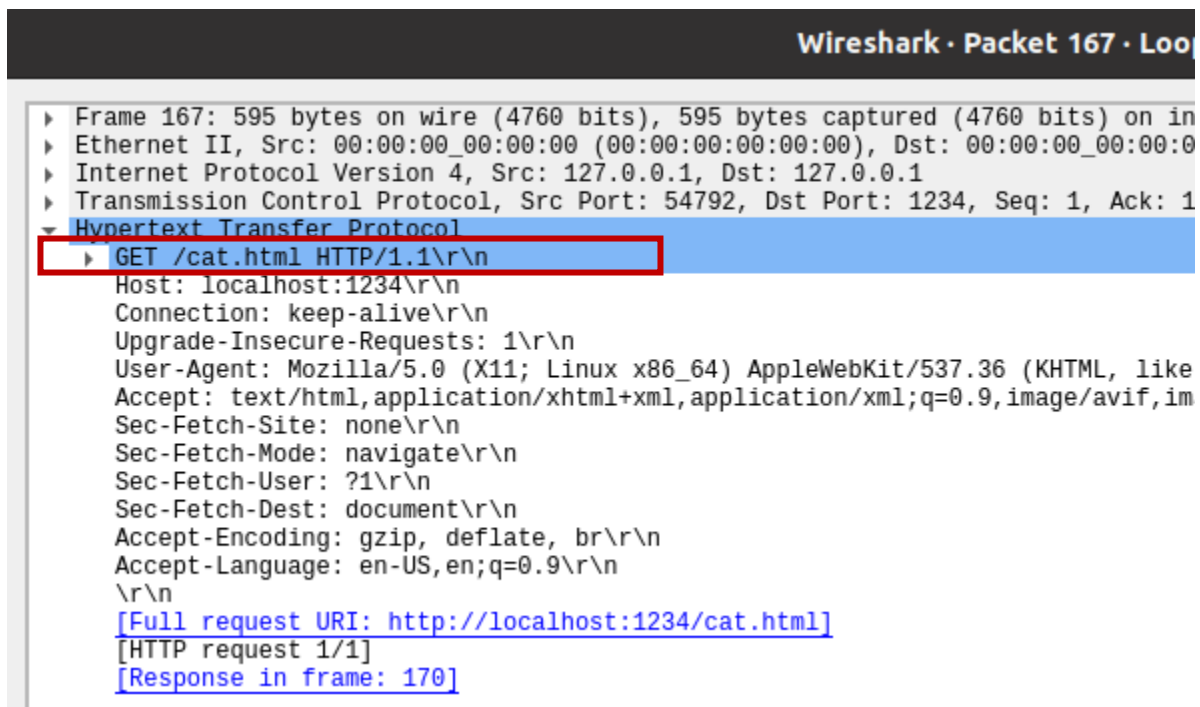


To demonstrate this, here are the current contents of the directory:



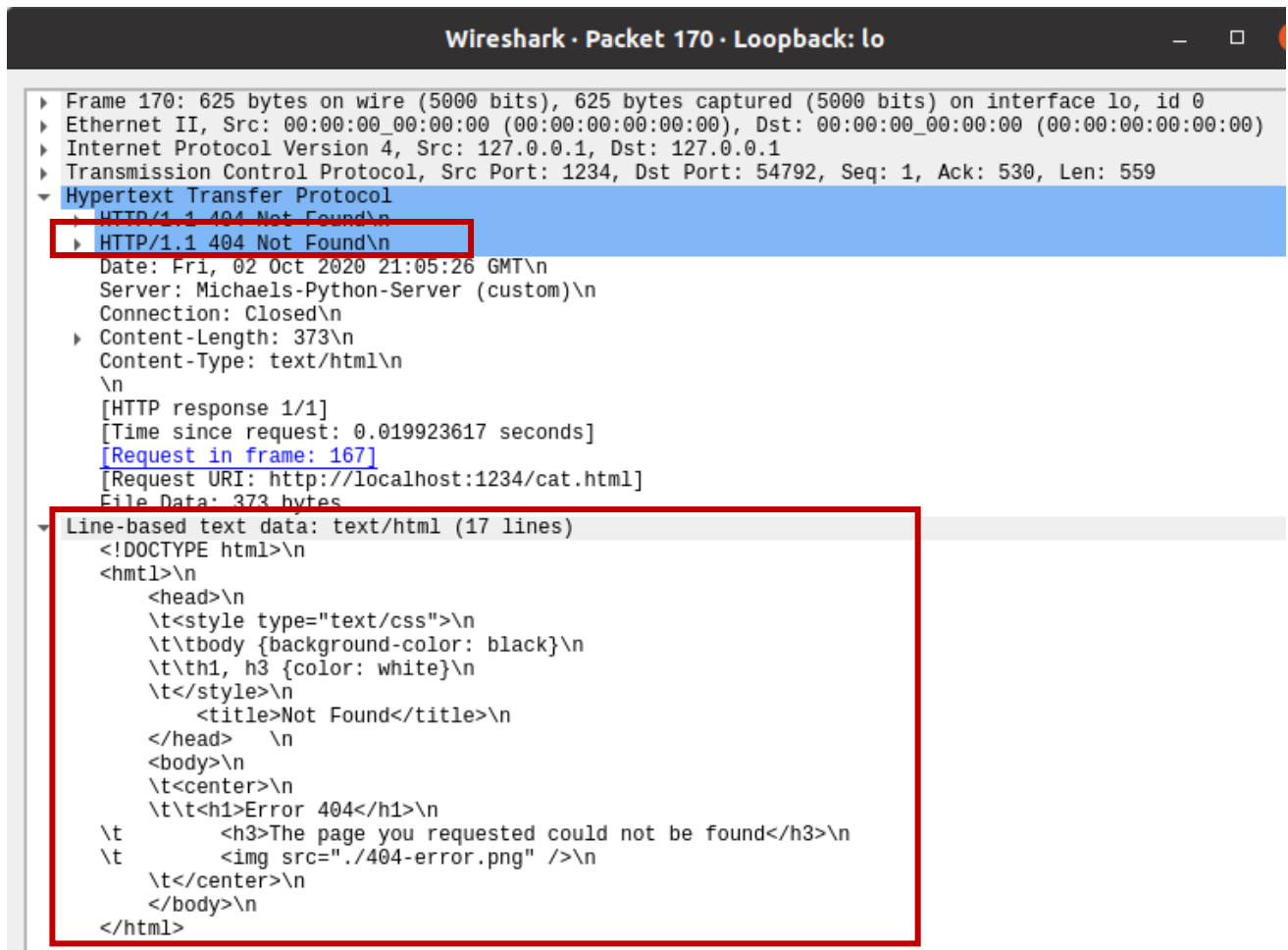
The webserver is running from this directory and I will request an html file in the browser called “cat.html”. Since this file does not exist it should generate a 404 error.

**cat.html get request from the web browser:**

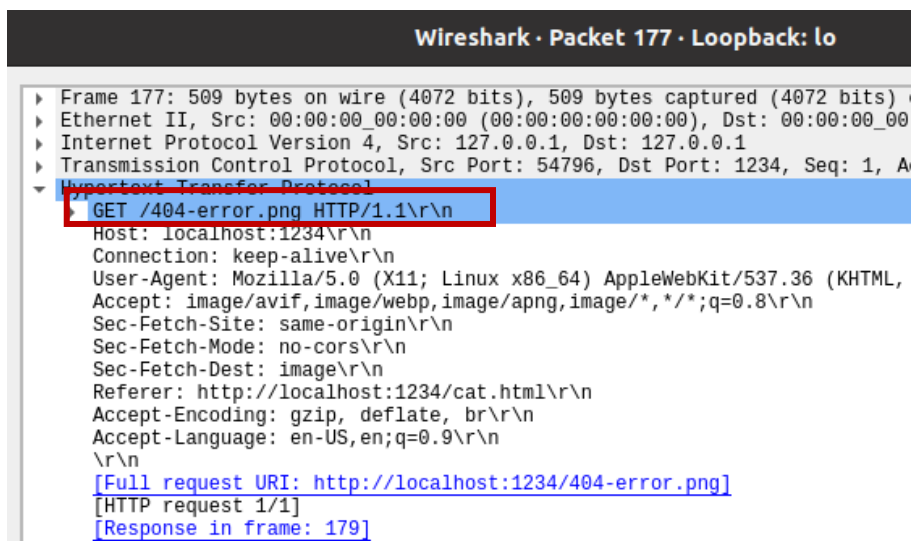




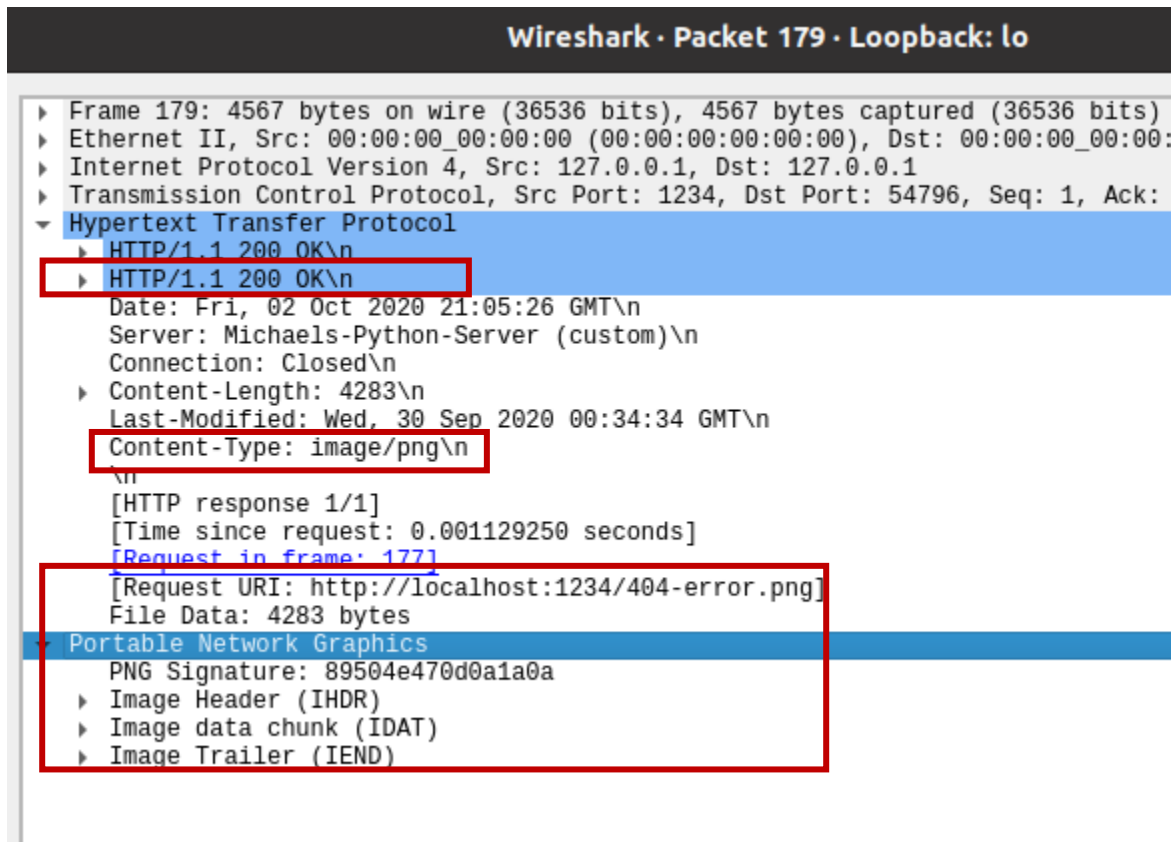
#### 404 response from the server:



#### 404-error.png get request from the web browser:



404-error.png response from server:



### Known Bugs

The program runs fine and as far as I can tell there are no known bugs that crash the server by navigating to different pages. However, for some reason, some of the threads will run into an `IndexError` trying to read from the tokenized list of strings that are received when the server receives a request from the browser (this list is used to obtain the file requested). The server will receive an `GET` request, but there will be no file requested from the browser in the response. This is a minor bug and will NOT crash the program.

### References

<https://realpython.com/python-sockets/>

<https://realpython.com/intro-to-python-threading/>

<https://docs.python.org/3/library/datetime.html>