Super HTTP - DoS

Instructions

1. Use the same HTTP server from the recap topic (`httpserver.py`) but rename it to `superhttp.py`. You may either use the attachment given with the exercise, or your updated server you submitted - it doesn't matter; only make sure the server is set to listen on port 8005.
2. Create a text file `request.txt` with the contents as this HTTP request:

```
GET / HTTP/1.1


```

Note: Ensure there are two empty lines immediately following the `GET / HTTP/1.1` line, resulting in a total of three lines in the file.

3. Run the server and use netcat to send the HTTP request.

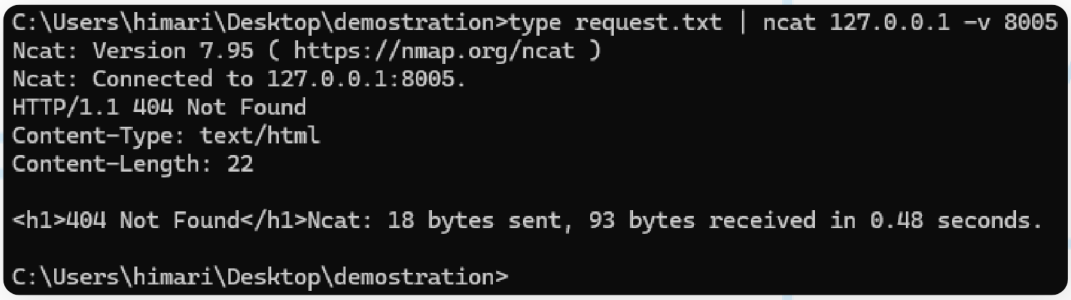
Command (Windows):

```
type request.txt | nc 127.0.0.1 8005`
```

Command (MacOS/chromeOS)

```
cat request.txt | nc 127.0.0.1 8005`
```

You should get the HTTP response and the connection should close (even a 404 result is fine).

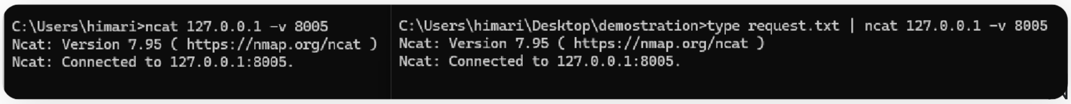


```
C:\Users\himari\Desktop\demostration>type request.txt | ncat 127.0.0.1 -v 8005
Ncat: Version 7.95 ( https://nmap.org/ncat )
Ncat: Connected to 127.0.0.1:8005.
HTTP/1.1 404 Not Found
Content-Type: text/html
Content-Length: 22

<h1>404 Not Found</h1>Ncat: 18 bytes sent, 93 bytes received in 0.48 seconds.

C:\Users\himari\Desktop\demostration>
```

4. Open two terminals simultaneously. Connect with netcat to the HTTP server (`nc 127.0.0.1 8005`) in one terminal, but don't send the request - keep the connection hanging. Now from the second terminal, send the `request.txt` file the same way as before with netcat - there should be no response, since the server is stuck on the first connection! Once you kill the first connection (with `Ctrl+C`), you should get the response.



```
C:\Users\himari>ncat 127.0.0.1 -v 8005
Ncat: Version 7.95 ( https://nmap.org/ncat )
Ncat: Connected to 127.0.0.1:8005.
```

```
C:\Users\himari\Desktop\demostration>type request.txt | ncat 127.0.0.1 -v 8005
Ncat: Version 7.95 ( https://nmap.org/ncat )
Ncat: Connected to 127.0.0.1:8005.
```

ote: The terminal on the left was started before the terminal on the right.

5. Fix this issue by adding a 10 second timeout on reading from the client socket. This means the server will be "stuck" for at most 10 seconds.

To submit

Submit file `superhttp.py`.

Sentinel