Homework 5: HTTPS

There are no learning objectives for this assignment.

## Application Objective 1: HTTPS and Nginx

Setup your app to listen for HTTPS requests on port 443 using a self-signed certificate and nginx running in a separate docker container. The app must function properly. More specifically:

* If you receive an HTTP request over port 80, it should be redirected to HTTPS over port 443
* The app should be set up to use WebSockets for the chat feature. The WebSockets should be encrypted using WSS
* Add the secure directive to your auth token cookies

### Testing Procedure

1. Start your server with docker compose up
2. Open a browser and, with the network tab open, navigate to http://localhost
   1. Accept any warning about a self-signed certificate
   2. Verify that the home page loads
   3. Check the network tab to verify that you received a redirect to https://localhost and that the page loaded properly over HTTPS
3. Verify that WebSocket connection was made using the WSS protocol
4. Register and login
5. Send a message in chat and verify that the feature still works (Displays your username) over WSS
6. Check the auth token cookie and verify that the secure directive was set

## Application Objective 2: Serving Static Files with Nginx

Modify your Nginx configuration to host all of your static files directly through Nginx. This means that Nginx will respond to requests for static files without forwarding the request to your server. The static files to be hosted are anything that starts with "/public/" as its path. More specifically, this includes:

* Your CSS file(s)
* Your JavaScript files
* Your favicon
* All images from the handout code
* All images uploaded by users (This is the most difficult part of this objective)

Note: Your HTML will still be hosted by your server. Since it is rendered from a template, it is not static content.

Note: We will not test with video uploads, but you should ensure your image upload feature works alongside your WebSocket chat for this objective.

Hint: As a recommended approach to setup user uploaded images, look into volumes as a way to share a local directory between containers.

### Testing Procedure

1. Start your server with docker compose up
2. Open a browser and, with the network tab open, navigate to https://localhost
3. Verify in the network tab that the static files were served by Nginx, while the non-static content was served by the server
   1. Check the code to verify if needed
4. Register and login
5. FInd the file upload form next to the chat box and upload an image
   1. Verify that the image appears in chat with your username
   2. Verify that the image was hosted by Nginx

## Submission

Submit all files for your server to Autolab in a .**zip** file (A .rar or .tar file is not a .zip file!). Be sure to include:

* A Dockerfile in the root directory
* A docker-compose file in the root directory that exposes your app on ports 80 and 443
* All of the static files you need to serve (HTML/CSS/JavaScript/images)

| It is **strongly** recommended that you download and test your submission after submitting. To do this, download your zip file into a new directory, unzip your zip file, enter the directory where the files were unzipped, run docker compose up, then navigate to localhost:8080 in your browser. This simulates exactly what the TAs will do during grading.  If you have any Docker or docker compose issues during grading, your grade for each objective may be limited to a 1/3. |
| --- |

## Grading

Each objective will be scored on a 0-3 scale as follows:

| 3 (Complete) | Clearly correct. Following the testing procedure results in all expected behavior |
| --- | --- |
| 2 (Complete) | Mostly correct, but with some minor issues. Following the testing procedure does not give the exact expected results |
| 1 (Incomplete) | Clearly incorrect, but an honest attempt was made to complete the objective. Following the testing procedure gives completely incorrect results or no results at all. This includes issues running Docker or docker compose even if the code for the objective is correct |
| 0 (Incomplete) | No attempt to complete the objective or violation of the assignment (Ex. Using an HTTP library) -or- a security risk was found while testing the objective |

Note that for your final grade there is no difference between a 2 and 3, or a 0 and a 1. The numeric score is meant to give you more feedback on your work.

| 3 | Objective Complete |
| --- | --- |
| 2 | Objective Complete |
| 1 | Objective Not Complete |
| 0 | Objective Not Complete |