Project Part 4

Objective 1: Deployment and Encryption

Deploy your app on a publicly available server with a domain name and HTTPS with a valid certificate. You may use any means available to accomplish this, though it is recommended that you take advantage of the GitHub Student Developer Pack.

When deployed, add a link to your app in the readme of your repository so we can find your deployment.

Note the following implications of this objective:

- You will use the WSS protocol for your WebSocket connection (If you finished WebSockets in part 3)
- Your certificate must be valid. It's recommended that you use CertBot to acquire a free certificate
- Any HTTP requests must be redirected to use HTTPS. Do not let users access your app with unencrypted requests

Free Clause: You are not required to spend any money to take this course. If your team is in a situation where you need to spend money to deploy your app (eg. you all already used your student developer pack credit on other projects), please let me (Jesse) know and I'll work with you to ensure you are not required to spend money on the course requirements. You pay enough in tuition. You do not need to pay more to take a class.

Testing Procedure

1. Find the app url in the project repo and navigate to the public deployment
2. Ensure the page loads using an HTTPS connection and no security warnings appear
3. Navigate to the same app using HTTP and verify that you are redirected to the app using HTTPS
4. ~~Verify that a WebSocket connection is made using WSS and that WebSockets are functional~~ (Part 3 functionality not required to earn full credit in part 4)

Objective 2: Email Verification

Add email verification. Email should be indicated as "not verified". Send the user a unique url to their email. When clicked, their email is then verified.

Do not use your personal email account! You should create a new email account for your project.

Note: You are expected to use a Gmail account for this objective, but if you find another way to send emails from your server you are welcome to complete the objective any way you'd like.

Security: Your email credentials must never be pushed to your repo. This is critical. Your repo is public so pushing your credentials means everyone on the planet will have access to your email account

Testing Procedure

1. Navigate to the public deployment
2. Create an account with a valid email address (TAs: You may want to create a new email account for testing)
3. Verify that somewhere on the page your email address is indicated as "not verified"
    a. If there is a clear way to access a profile page (eg. clicking on your profile image), navigate to it for this check
4. If there is a button to press to verify your email, click it. If this button exists, it should be close to the "not verified" text (The verification email might have been sent automatically on account creation)
5. Go to your email and check for a verification email sent by the deployed app
    a. Make sure there are no security warning for the email
6. Look for a verification url in this email
    a. Verify that the url contains a long random string with enough entropy that it cannot be guessed by an attacker (At least 80 bits of entropy)
    b. Click the link
7. Refresh the page on the app and ensure that it now indicates that your email is verified
8. **Security:** Look through the repo and ensure that no email credentials have been pushed

Objective 3: DoS Protection (IP rate limiting)

Add very basic DoS protection to your app based on rate limiting IP addresses. Your IP protection should:

- Block requests from an IP address if it has made more than 50 requests within a 10 second period
    - Every single request from an IP should count towards this limit (eg. Just loading your homepage requires ~5 requests that all count toward the limit)
- While an IP is blocked, respond to all requests from the IP with a 429 "Too Many Requests" response with a message explaining the issue to the user
- When an IP becomes blocked, it should remain blocked for 30 seconds
- After 30 seconds pass, requests from the IP should be handled as usual unless they hit your rate limit and become blocked again

Testing Procedure

1. Navigate to the public deployment
2. Count the number of requests made to load the page
3. Refresh slowly enough times to reach 70 requests, but not fast enough to trigger the rate limiter (<50 requests per 10 seconds)
    a. Ensure all requests are handled as expected
4. Wait ~10 seconds
5. Refresh quickly such that >50 requests are sent in at 10 second window
    a. Ensure that after the 50th request, you start seeing 429 responses and the page no longer loads
6. Wait ~20 seconds
7. Refresh the page and ensure that you are still blocked
8. Wait another ~10 seconds
9. Refresh the page and ensure that it loads
10. [Hit the rate limit again on the same device] Refresh quickly such that >50 requests are sent in at 10 second window
    a. Ensure that after the 50th request, you start seeing 429 responses and the page no longer loads
11. Within 30 seconds, on a second device with a different public IP address, navigate to the public deployment and ensure that the page loads
    a. Make sure the other device has a different **public** IP address. If you are at a residence, all your devices probably have the same public IP address. To ensure you have a different IP, you can use your phone on mobile data, or turn on a VPN

Submission

Add your public link to your deployment to the README in your repo. We will use this public deployment for grading purposes.

Objective 3 will be graded by cloning your repo and running docker compose. Ensure that your repo contains everything needed for objective 3 before the deadline.

Team Scoring

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

3	Clearly correct. Following the testing procedure results in all expected behavior
2	Mostly correct, but with some minor issues. Following the testing procedure does not give the exact expected results

1	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	No attempt to complete the objective or violation of the assignment (Ex. Not using a framework) -or- a security risk was found while testing the objective

3	2 Application Objectives
2	1 Application Objective
1	0 Application Objectives
0	0 Application Objectives

Please note that there is only one chance to earn these application objectives. There will not be a second deadline for project parts 2-4.

Individual Grading

The grading above will be used to determine your team score which is based on the functionality of your project. Your actual grade may be adjusted based on your individual contributions to the project. These decisions will be made on a case-by-case basis at the discretion of the course staff. Factors used to determine these adjustments include:

- Your meeting form submissions: team meeting and eval form
    - You must fill out this form after every meeting. Failure to do so is an easy way to earn a negative individual grade adjustment
    - The quality of your submissions will be taken into account as well (eg. Saying "I'll do stuff" before the next meeting is a low quality submission)
    - You may submit more meeting forms than are required even if there was no meeting (eg. If you want to adjust your evals after a deadline without waiting for the next meeting)
- Your evaluations from the meeting form
    - If you teammates rated you poorly/excellently, your grade may be adjusted down/up respectively
- Your commits in the team repo
    - Your commits may be check to see if you did in fact complete the work you mentioned in the meeting form, as well as compare the amount of work you completed to that of your teammates
    - You **MUST** commit your own code! It is not acceptable for your teammate to commit your code for you. You should have a clear separation between your

tasks and commit the code for your task. If a commit is not in your name, you effectively did not write that code. If a teammate is making this difficult, let the course staff know in the meeting form