Report for Project #3: HTTP server with authentication

1. Team Members:

Yuchen Wei (yw780) and Tiange Zhang (tz196)

1. Collaboration:

We use GitHub to collaborate between the two of us, updating and communicating the parts we have completed and the changes and improvements we have made to the code. The primary reference materials we used in the process were from the lectures we received, the previous Python programming courses we participated in, and our experience in learning Python independently. We completed sever.py and report together offline.

1. Is there any portion of your code that does not work as required in the description above? Please explain.

After comparing the results of our tests with the samples included in the project and the description, we believe that our code fulfills all the requirements of the description and can be reflected by the tests.

1. Did you encounter any difficulties? If so, explain.

The overall process of completing sever.py was relatively smooth; however, during testing, we found that entering both the correct token of an existing cookie and a pair of incorrect usernames or passwords via curl resulted in the web page not being able to perform the success popup. If we only do web tests, it could not be reflected. After we talked to TA Parvathi, she suggested us while checking the token entered by curl matches with the token in the existing cookie and then generate a flag, and if the cookie is successfully read, the username and password will not be checked, and if not, the check will continue.

1. Describe two observations or facts you learned about HTTP and cookies in the process of working on this project. Please be specific and technical in your response.

About the concept of cookie storage in HTTP, at first, we all thought that a cookie needs to save both username and password, and if a cookie is called during login, we assumed that the username and password saved in the cookie would be extracted and automatically help the user enter them to pass the authentication. However, the actual cookie storage simplifies many steps, as the cookie requires only an identifier and the username of the last successful login to record the user login information and save the user's login time. This is why HTTP uses cookies extensively, as they are simple to use and significantly reduce the amount of data that needs to be stored.

In our tests, we found out how to check and clear the cookies stored in the browser for usernames and passwords that have been successfully entered. In browsers like Chrome, they have a built-in feature to find and delete history and also to clear cookies. So we can either repeatedly test the results by clearing all saved cookies directly or by looking at the token of the cookie that has been stored in the generated cookie.txt file to find the exact cookie and remove it.