Antonio Scalfaro

12/06/2023

SDEV 300 / 7381

Lab 8 – Web Application

This week’s lab was to add additional functionality to our web application. I added functionality for updating a password for a user and logging functionality to log all failed login attempts. There is a new route added (‘/update\_password’) that has a form that takes the username, old password, and new password. If the username is not found, the old password does not match the stored password, or the new password is an invalid password, there will be no update made and a flash message will appear below the form. If an update is made, the flash message will display under the form informing the user of the change. This was a fairly easy piece of functionality to add because I had most of the code already available in previous functions and only needed to tweak some lines. I also got some experience using temporary files and the library shutil to help overwrite the existing credentials.txt with the updated credentials. I found that the required check against the common passwords was moot because the passwords are required to be 12 characters long, 1 upper, 1 lower, and 1 special, which is not found on the CommonPassword.txt. In order to test the functionality anyways, I added a common password ‘Apass4nico!!’ which does prevent that password from being used, confirming the functionality works. I really enjoyed seeing this project come together and adding all the layers on week by week and seeing how well written code allows for quick modification of existing code bases. The hardest part of this weeks requirements was the logging. I had to figure out how to prevent the logging of additional actions on the webpage once a successful login was made. It took some time but I figured out how to disable the logging upon successful login. All failed logins are recorded in logger.txt. Logging all the failed logins can help to ensure that abuse is not being perpetrated. If too many failed logins are attempted in a short period of time, this could indicate a malicious attempt to get into the system. Since the IP address is also recorded, it would be more beneficial to correlate failed logins in a certain period of time with any given IP address. This information would provide a good starting point to identify threats to the site. My current password requirements match NIST standards, so there was no tweaking needed there. The Pylint score was 10/10.

**Update Password Page:**

A screenshot of a computer

Description automatically generated

**Logging File for Failed Login:**

A screenshot of a computer

Description automatically generated

**Pylint:**

A screenshot of a computer

Description automatically generated