**Simple Forum Application**

Michael Clark

CS9163: Spring 2013

This application is a very simple forum that allows all users to view threads and associated posts, while only allowing authenticated users to create new threads and post replies. Authenticated users are also able to upload certain types of files as part of the post. All users are allowed to view and download these files.

The application is written in Python and uses the Web Service Gateway Interface (WSGI) instead of using a more commonly used higher-level framework such as Django (which itself implements WSGI).

Code is also available on github at: <https://github.com/fredtheranger/forum>

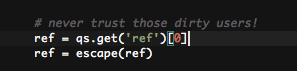
**The application’s main components consist of the following:**

* Forum.py – The application’s main class. This module can be run as the main program and utilizes wsgrief.simple\_server to implement a simple HTTP server instead of requiring the application to be deployed in an actual HTTP server such as the Apache httpd server.
* Forum.models – Modules representing the main pieces of the application (thread, posts, users, and files).
* Forum.dao – A basic module to allow access to the forum’s database, including inserting and querying the tables representing the above models.
* Tests – Unit tests for the DAO and models.

**Running the application:**

1. Update paths as needed in the forum.ini file.
2. Create the database and associated schema by running the setup\_create\_db.py script in the root of the forum directory. The script will allow create a default admin user.
3. Open <http://localhost:8080/> in a browser.
4. In order to create threads or posts you will need to login (default user is admin and password is admin).

**Potential vulnerabilities and mitigations:**

* **XSS** – User input is escaped using the Python CGI.escape function. This function is a simple way to prevent basic XSS attacks. From the Python manual, the function will “convert the characters '&', '<' and '>' in string s to HTML-safe sequences.”  
    
  
* **SQL Injection** – In order to protect against SQL injection, all database SQL uses parameter substitution.   
    
  
* **CSRF** – In order to prevent cross script request forgery (CSRF) attacks, a token is added as a hidden field to forms. The token is associated with a session and has a brief lifespan. Upon form submission the token submitted with the form is checked against the token saved in the session table in the database.   
    
  
* **Directory traversal** – Directory traversal attacks are mitigated while allowing users to upload and download files by hard-coding a base path for where the uploaded files are stored and only saving the basename of the file uploaded. The download URLs do not reference the filename but instead an integer rowed of the file reference stored in the database. If an attacker crafts a malicious URL in an attempt to exploit the download feature by using something other than a valid integer, then the application will respond with a page not found error.

**Development practices**

Despite being a small project completed in a relatively short timeframe (a couple hours a day for a couple weeks versus hundreds of hours over the course of months), this project attempted to follow good development practices, including the following:

* **Source control repository** – The project used github. Github was chosen mainly on the recommendation of the class lecture as well as to take the opportunity to learn a new tool (previously, I had used Subversion almost exclusively). As stated above the repository for this project can be found at <https://github.com/fredtheranger/forum>
* **Test-driven development** – For each of the applications main models as well as the data access layer, unit tests where written prior to implementing the actual functionality. This provided verification that the functionality worked as expected as well as allowing for regression testing. Technically these tests fail the criteria for pure unit tests because of the way the models are dependent upon one another and a lack of a library to mock objects in the standard Python libraries (there are external libraries that could be used but outside of the scope of this project).