Click here to complete the Project 1 peer eval form.

SY306 Team Project 2

Message Board Defense

0. Team Assignments

This project will do done using the same teams as for project 1.

1. Backup & Set-up

1. For backup purposes, create a zip file containing everything from your `/public_html/project01` folder.

2. If you have not yet submitted your `project01` zip file to `submit.cs.usna.edu` (one submission per team required), do that now.

3. Create a new folder inside the team lead's `public_html` called `project02` and store your work in that directory.

4. To start off, **copy** - do not move - your files from `project01` to `project02`.

    Tip: an *easy* way to copy your `project01` files to `project02` is to run the following command from the team lead's `~/public_html$` directory:

    ```
    cp -p -r project01 project02
    ```

5. Save the current state of your database, so you can later restore it if needed.

    a. From the command line (as the team lead):

        i. ssh into `midn.cyber.usna.edu`

        ii. in the shell on `MIDN`, execute the following commands:

    ```
    cd /public_html/project02
    ```

    ```
    mysqldump -u m21xxxx -p -h csmidn m21xxxx > project01_backu
    ```

    b. Check that the file `project01_backup.sql` was created in your `project02` folder

2. Security & Defense

1. Make sure all your files for the project are copied to your `project02` folder. All the modification for this project will be done in this folder. Do **NOT** modify files in the `project01` folder.

2. Make all necessary modifications to ensure your project still works after being copied to `project02`. Double check all links between pages.

3. In your `project02` folder, implement defensive measures to prevent the attacks that your colleagues might come up with for your website. Do NOT modify the tables in your database. If needed, create new tables with a different name (ex Users_secure, Messages_secure).

4. **At a minimum, you must:**

    a. Limit input size for the post to some reasonable length (in Python)

    b. Do not store passwords in plain text (see examples in classes 20 on how to use a hash function in Python and MySQL)

    c. Duplicate all/any JS checks in Python

    d. Mitigate the risk of HTML and JavaScript injections by escaping < and >

    e. Use parametrized queries (similar to the class 20 examples) to prevent SQL injections

    f. Implement the "secret /CSRF" token approach or captcha to prevent Cross-site Request Forgery Attacks

    g. Implement secure session management (client & server cookie/session validation) to prevent session hijacking/riding.

    h. Use HTTP digest authentication to secure access to an admin only portion of your website. For example, maybe there is a dashboard that is only accessible to a user with admin privileges. **NOTE:** HTTP authentication will be taught during class 26, but is open now for review.

5. You can implement any other security measures (including security specific libraries) that you think will improve the security of your application.

6. Document your changes and security measures implemented in a file `sy306_project2SecurityImplementation_teamX.docx`. Be sure to include the username and password necessary to access part `4.h` above. If your team had any security defense implemented for Project1 project, describe that in the document as well.

3. Deliverables

1. All of your files should be in a folder called "`project02`" (without the quotes) on the team lead's `public_html`. **Your instructor will assume that your web pages are viewable at http://midn.cyber.usna.edu/~m21xxxx/project02/index.html** or index.py or whatever entry page for you application is, where m21xxxx is your team lead's alpha number.

2. In the team lead's default.html page, under the heading **Project02** create a link to the team's project02 entry page as well as a link to the `sy306_project2SecurityImplementation_teamX.docx` document.

3. Your files in the project02 folder should have all the original functionality of project01 with the security improvements of part 2 (above) properly implemented.

4. **Project 02 Electronic submission: Due 03:00 on Wednesday April 24th**

   a. The file `sy306_project2SecurityImplementation_teamX.docx` documenting your changes and describing the security measures implemented, via the online system: `submit.cs.usna.edu`.

   b. All the Project02 message board files via the online system: `submit.cs.usna.edu`.

4. Extra Credit

For **five points** added to your **final SY306 project grade** do the following:

1. **Requirements:** Implement the message board chat functionality (post & view messages) of Project02 with AJAX and JSON:

   a. When posting a new message, a user's message board will be dynamically updated with the new message without the site having to be reloaded.

   b. When not posting, a user's message board will be dynamically updated with new messages (*from other users*) without the site having to be reloaded.

2. Deliverables

   a. Copy your files from `Project02` into a new folder called `Project02_EXTRA` and perform all your extra credit work in that folder.

   b. The submission deadline for this extra credit is **23:59 on Monday April 29th.**

   c. In the team lead's default.html page, under the heading **Project02_EXTRA** create a link to the team's `project02_EXTRA` entry page.

   d. Submit a zip file with all the Project02_EXTRA message board files via the online system: `submit.cs.usna.edu`.

NOTE 1: While both Dr. Richards and LCDR Leavitt will be glad to broadly discuss AJAX & JSON concepts, they will not answer code specific questions for the Extra Credit assignment.

NOTE 2: Saving a backup copy of your working lab is recommended before starting on this.

NOTE 3: Review class 25, JSON and AJAX with Python for additional guidance.