XSS Laboratories

Cosc424, 2021

Weeks starting the 16th and 23rd of August

Work for assessment must be submitted before 9am Monday 30th of August.

## Laboratories

In the XSS labs you are to complete exercises that explore persistent and reflected cross-site scripting. In doing so, you will ensure that all tests provided in `tgt_server.test_xss.py` will pass and that you can demonstrate persistent and reflected cross-site scripting interactions with the implemented web application. Note that additional unit tests will be used for the purpose of determining a final mark.

These labs involve modifying a Python 3.9 web application that interacts with an in-memory database. The web application is implemented using the FastAPI web application framework and the jinja template engine. Minimal Javascript coding is also required.

> This web application is implemented for the purpose of these labs only.

Start the web application from the command line as shown below:

```
(venv) cd tgt_server
(venv) PYTHONPATH=../ python3 ./__main__.py
INFO:     Started server process [4686]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:5000 (Press CTRL+C to quit)
```

Using a browser, visit the Home page for the web application at `http://0.0.0.0:5000/user-profiles`. Clicking on a name will provide more information about that person.

For a technical description of the web application, visit the web page at `http://0.0.0.0:5000/docs` to see the Swagger specification of the REST API provided. Take a careful look around and check out all options, including the buttons labelled 'Try it out'.

> *Note that the above may only work after you have completed one or more exercises described below.*

To run the unit tests using pytest:

```
(venv) cd tgt_server
(venv) PYTHONPATH=./ pytest test_xss.py
```

Note the different PYTHONPATH used to run the unit tests.

### Submission

All changes you are to make to the code will be within the `tgt_server.exercises` package: no other file is to be changed or will be inspected during marking.

To submit your work, compress the `tgt_server` folder and upload via UC Learn.

### Marking

A total of 20 Marks will be awarded for the answers to the exercises provided below and contribute to 10% of the final grade. Marks are earned by successfully passing all unit tests and for coding style. Coding style will be judged upon: following Python best practices; appropriate comments; appropriate exception handling; and so on.

## Exercises

In lectures a number of cross-scripting techniques have been described: of these two will be explored in practice during these exercises. First, however, you complete an initial exercise that involves performing tasks that will result in a running web application.

In all cases, the code provided has comments that explain what functions do: feel free to add your own unit tests if that will be helpful.

The web application provides a static webpage as the target for cross-site scripting. See `tgt_server.exercises.ex1.malicious_url` for the url.

### Exercise 1 [8 marks]: Web Application Implementation

**[2 Marks] The Cross-site Script** The variable `tgt_server.exercises.ex1.malicious_xss` is to contain a very simple script: an assignment statement that sets the `document.location` property to be the `malicious_url`. The malicious url is provided in the same module `tgt_server.exercises.ex1`.

**[6 marks] Complete methods for updating and searching the in-memory data** An in-memory data set is provided in `tgt_server.data`. Take a careful look at the organisation of the values provided because you are to implement the two methods `update_user_profile` and `search_profiles` in the module `tgt_server.exercises.ex1.py`.

Successful implementation means that you will pass more unit tests and additional functionality will be observed in the web application.

### Exercise 2 [6 marks]: Using the REST API provided by the Web Application

Demonstrate your understanding of the REST API by implementing the methods `get_profile_id()` and `update_profile()` in `tgt_server.exercises.ex2.py`

Your implementation interacts with the REST API using the testclient passed in as a parameter to these methods.

**Exercise 3 [6 Marks]: XSS Attacks**

Demonstrate your understanding of two forms of XSS: persistent and reflect.

**[3 Marks] Reflection XSS** Implement `reflect_xss` in the module `tgt_server.exercises.ex3.py`. Successful implementation means that you will pass more unit tests.

**[3 Marks] Persistent XSS** Implement `persistent_xss` in the module `tgt_server.exercises.ex3.py`. Successful implementation means that you will pass more unit tests.