Building Secure Web Applications with Python and Flask

This course is an introduction to building secure, full-stack web applications with Python and Flask. We'll start with Python and Flask and transition to web application security, where we'll look at different types of security vulnerabilities and best practices to patch up these vulnerabilities in your own apps. Then, we'll go to building your own API and securely authenticating with it, and finish by showing you how you can deploy your web app!

Course Details

- Course: CMSC389F
- Prerequisites: C- or better in CMSC216 and CMSC250
- Credits: 1
- Seats: 30
- Lecture Time: TBD
- Location: TBD
- Semester: Fall 2019
- Textbook: No textbook, all materials are provided and documentation is online
- Course Facilitator(s): Yashas Lokesh, Kenton Wong
- Faculty Advisor: Michael Marsh

Topics Covered

- Python
    - Variables, expressions, operators
    - Iterations, conditionals
    - Functions
        - As first-class objects
        - Decorators

    - "Main" function
    - Built-in functions
- Web Application Security
    - Cross-site scripting (XSS)
    - Cross-site request forgery (CSRF)
    - SQL injections
    - Man-in-the-Middle attacks (MitM)
    - Token & Two-factor authentication
- Flask
    - Routing your web app
    - Templating
    - Creating a REST API
    - Adding extensions for more features
        - WTForms
        - SQLAlchemy
        - Freeze
- SQLite
    - Lightweight SQL database
    - Local data storage
- Python packages
    - Requests
    - Bokeh
    - JSON
    - SQLite
- App Deployment
    - Heroku
    - Python Anywhere
    - *Possibly*: Google App Engine, AWS
- Version Control
    - Git

Schedule

Week	Topic	Assignment
1	Intro to Python	Python practice (P1) assigned
2	Requests, working with JSON	P1 due, GitHub API project (P2) assigned
3	Flask Intro	
4	Intro to Web App Security	P2 due, Basic Flask App (P3) assigned
5	Databases, SQL Injections	
6	User Management	P3 due, User Management project (P4) assigned

Week	Topic	Assignment
7	Cross-site Request Forgery (CSRF)	
8	Cookies, MITM, Security Headers	P4 due
9	Midterm	
10	Creating RESTful APIs	
11	Two-Factor Authentication	Final Project assigned
12	Useful Python packages	
13	THANKSGIVING BREAK	
14	Deploying your app	
15	Presentations	Final Project due

Grading

Grades will be maintained on (ELMS/department grade server/etc). You will be responsible for all material discussed in lecture as well as other standard means of communication (Piazza, email announcements, etc.), including but not limited to deadlines, policies, assignment changes, etc.

Grades will be maintained on the CS Department grades server.

You will be responsible for all material discussed in lecture as well as all other standard means of communication (Piazza, ELMS announcements), including but not limited to deadlines, policies, assignment changes, etc.

Your final course grade will be determined according to the following percentages:

Percentage	Title	Description
50%	Projects	Weekly projects to apply lecture material and make practical applications.
20%	Midterm	Examination
30%	Final Project	Final project to demonstrate mastery of all topics learned and apply knowledge to create a new application from scratch.

Any request for reconsideration of any grading on coursework must be submitted within one week of when it is returned. No requests will be considered afterwards.

Projects

The project is due the day it is scheduled to be due, barring any extensions that may be given out. They will be due at 11:59 PM. Not all of the projects will have tests; they will be graded according to a rubric which will also be provided. All projects must be submitted online at the submit server.

Late Policy: Projects may be submitted up to one day late for 10% off your earned grade. After this, no more projects will be accepted. The highest score you get on the project, counting late and on-time submissions, will be your grade for that project. There are **no exceptions** unless you've talked with us beforehand or provide a valid excuse.

We will look at your most recent on-time and late (if applicable) submissions when grading.

Every project will have 10% of the project grade reserved for style: proper formatting and commenting.

Midterm

The midterm will test your knowledge of Python, Flask, and all security topics we discussed prior.

There will be conceptual questions on Python, Flask, and web application security. There will be 1-2 Python/Flask coding questions, and the rest will be fill-in-the-blank or short response questions on Python, Flask, and security.

All material discussed in lectures before the midterm will be tested.

Outside-of-class communication with course staff

We'll communicate through students mainly through Piazza and through office hours.

Office hours: TBD (one hour a week), or by appointment

Email should only be used for emergencies, please use Piazza, otherwise. We'll get back to you more quickly on Piazza.

Instructor:

Dr. Michael Marsh - mmarsh@cs.umd.edu

Facilitators:

Yashas Lokesh - yashloke@terpmail.umd.edu

Kenton Wong - kdubbs0@umd.edu

Excused Absence and Academic Accommodations

See the section titled "Attendance, Absences, or Missed Assignments" available at Course Related Policies.

Disability Support Accommodations

See the section titled "Accessibility" available at Course Related Policies.

Academic Integrity

Note that academic dishonesty includes not only cheating, fabrication, and plagiarism, but also includes helping other students commit acts of academic dishonesty by allowing them to obtain copies of your work. In short, all submitted work must be your own. Cases of academic dishonesty will be pursued to the fullest extent possible as stipulated by the Office of Student Conduct.

It is very important for you to be aware of the consequences of cheating, fabrication, facilitation, and plagiarism. For more information on the Code of Academic Integrity or the Student Honor Council, please visit http://www.shc.umd.edu.

Course Evaluations

If you have a suggestion for improving this class, don't hesitate to tell the instructor or TAs during the semester. At the end of the semester, please don't forget to provide your feedback using the campus-wide CourseEvalUM system. Your comments will help make this class better.