launch_code

LC101 2.13

Class Agenda

1. Announcement (Geek Gala)
2. Last Class Review
3. Lecture
4. TF Work groups!

Announcement

GeekGala - LaunchCode’s birthday party is happening on October 19th! (tomorrow)

If you are interested in attending please refer to the announcement on canvas for more information.

Review

MVC - Model, View, Controller

Model - The Data! (Database, classes, db object, SQLAlchemy)

View - What the user sees and interacts with! (HTML, CSS, Templates)

Controller - The logic! (Decorators and Functions)

Security

So far we have been using before_request(), a whitelist, and a session to determine if a user is logged in.

We have been storing user data (passwords) as strings in our DB.

If our DB was compromised, hackers would be able to see that information.

We need to incorporate some better security standards to ensure our data is as secure as possible.

Security

The field of Cyber Security within Information Technology is a massive entity. A constant game of cat and mouse.

Hackers find a vulnerability, programmers patch those vulnerabilities up, hackers find more vulnerabilities, programmers patch them up, ad infinitum.

We are going to be talking about some of these concepts in very high level ways.

Encryption

Encryption - a two way operation. We use an algorithm that scrambles user input based on some key. Using the key, and our reversed algorithm we can decrypt information.

If keys are stolen, encrypted information can be compromised.

Example: a codex

Hashing

Hashing - A one way operation. A hash function takes user input, and modifies it. It doesn’t randomly change things, it modifies it in the same way every time.

A user gives us sensitive information. We run it through our hash function. We store that hash in the DB.

When they login the next time, they give us their sensitive information. We run it through our hash function. We compare that hash to the hash stored in our database. If they are the same we can validate their password, and let them in.

hashlib

In our flask application we are going to hash our user passwords by using the the hashlib library.

Documentation:
https://docs.python.org/3/library/hashlib.html

The hashlib contains multiple hashing algorithms we saw Sha256 in action. It required us to send unicode, which required us to take a string and encode it. And then it returned a hash object.

That syntax looked like:
hashlib.sha256(str.encode(password)).hexdigest()

Hashing continued

Hashes are not perfect solutions. The relatively small number of hashing algorithms are well known.

Hackers can use these algorithms to generate hashes for common passwords (‘password’), and generate a large table of hashes, known as rainbow tables.

They can then use these thousands of hashes of common passwords to try to gain access to your user accounts.

Salting

To eliminate the effectiveness of rainbow tables people can stop using bad passwords (never going to happen) OR you can add a salt that adds a small amount of randomness.

A salt is a random string added to the end of a hash.

Since the random string is generated by us, and adds to the hash rainbow tables are rendered completely useless.

TF Groups

For the rest of class. You have time to work on your assignments. Review Prep-works, or old studios.

Take this time to get caught up if you're behind, or to get a head start on the assignments.