3 topics

15 minutes
Session Control
HTTP is a stateless protocol

What does that mean?
HTTP is a stateless protocol

HTTP has no built-in way of maintaining state between 2 transactions.
html stateless

no memory
session variables

- session variables stored on the server
- they persist across multiple pages
- unlike cookies - no expiration date. They die when the session ends.
- one way for a session to end is to close the browser. (cookies can persist through a browser close)
session variables in Flask:

Step 1: starting a session

from flask import session
import os
app = Flask(__name__)

app.secret_key = os.urandom(24).encode('hex')
...

from flask import session
import os
app = Flask(__name__)
app.secret_key = os.urandom(24).encode('hex')
...

session variables in Flask:

Step 1: starting a session

Return a string of n random bytes suitable for cryptographic use.
returns random bytes from an OS-specific randomness source.

used to sign cookies
session variables in Flask:

**Step 2: using session variables**

```python
session['username'] = 'zacharski'
session['zipcode'] = '88005'

print(session['username'])
q = "select * from users WHERE username = "%s" % session['username']"
```
session variables in Flask:

Step 3: deleting session variables

session.pop('username', None)
Dangerous Characters

Topic 2
solution:
you already know
Use this for ALL user input!

not 100% effective but a start!

demo
passwords

topic 3
The 20 Most Popular Passwords Stolen From Adobe

The Top 20
1. 123456
2. 123456789
3. password
4. adobe123
5. 12345678
6. qwerty
7. 1234567
8. 111111
9. photoshop

Yahoo email account passwords stolen

More than 2 million stolen passwords found on hacker server
How would you feel if it was your code that contributed to passwords getting stolen?
Simple Solution:
Never store passwords in the clear:
<table>
<thead>
<tr>
<th>id</th>
<th>username</th>
<th>password</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>raz</td>
<td>p00d13</td>
</tr>
<tr>
<td>2</td>
<td>ann</td>
<td>changeme</td>
</tr>
<tr>
<td>3</td>
<td>lazy</td>
<td>qwerty</td>
</tr>
</tbody>
</table>

extremely bad idea!!!!
<table>
<thead>
<tr>
<th>id</th>
<th>username</th>
<th>password</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>raz</td>
<td>p00d13</td>
</tr>
<tr>
<td>2</td>
<td>ann</td>
<td>changeme</td>
</tr>
<tr>
<td>3</td>
<td>lazy</td>
<td>qwerty</td>
</tr>
</tbody>
</table>

extremely bad idea!!!!
postgresql ->
many hash function options.

once you create a db:

CREATE EXTENSION pgcrypto

Adds crypto functions to your database.
One possibility: blowfish

```
insert into users
(username, password, zipcode)
VALUES ('ann',
crypt('password', gen_salt('bf')), 88005);
```
<table>
<thead>
<tr>
<th>username</th>
<th>password</th>
</tr>
</thead>
<tbody>
<tr>
<td>raz</td>
<td>p00d13</td>
</tr>
<tr>
<td>ann</td>
<td>changeme</td>
</tr>
<tr>
<td>lazy</td>
<td>qwerty</td>
</tr>
<tr>
<td>foo</td>
<td>$2a$06$LNEwiBctQkKjVatn0QSht.LJDRraYyC8AL15RweitjBssMrAG0l7a</td>
</tr>
<tr>
<td>foo</td>
<td>$2a$06$xSHddxN1HldRPwDVB.uBoeRzLghNBY/TkLCp0IVtQ9wkdZfgPAk5S</td>
</tr>
<tr>
<td>foo</td>
<td>$2a$06$ns8fXTCHanoXduRoHmWHre8aKvIwceU8eFz80D1GvkAyHwsipr5Qq</td>
</tr>
<tr>
<td>foo</td>
<td>$1$d4x0dbu4$u8M6hl/BTkz/7R4ev0vs7.</td>
</tr>
<tr>
<td>bar</td>
<td>$1$hG8URQvU$s02hg0Hjzhc6xJ.0KM7cL..</td>
</tr>
</tbody>
</table>

(8 rows)
One possibility: blowfish

session=# select username, password from users where password =
crypt('p00d13', password);
  username |                             password
----------+---------------------------------------------------------------------
     bar   | $2a$06$oT8bsN/kp1Cek2Eqe8Zehe2u9dqg5qJlwygFn.cmUovpjAvpuBrqq
(1 row)
how long to crack?

- 8 character pw a-z: 246 years
- 8 character pw A-Za-z0-9: 251,322 years
Summary

1. saw how to implement sessions
2. saw how to escape dangerous characters (more later)
3. saw how to handle passwords
Task

- clone session repository (see website)
- implement sessions so when a user logs in it is remembered on future searches
- protect user input strings (so a user can search on Peet’s)
- implement create account and new login using hashed passwords.
- demo up to next Thursday get max XP. (work in team demo individually)