COMP1531

5.2 - HTTP - Auth & Auth

Auth vs Auth

Authentication: Process of verifying the identity of a user

Authorisation: Process of verifying an identity's access privileges

Authentication

Naive method:

- User registers, we store their password
- When user logs in, we compare their input password to their stored password

Let's observe *auth.py*

(found in lectures repo)

Authentication

What's wrong with this?

Authentication

Using **hashlib** to create a hash

hash.py

```
1 import hashlib
2 print("mypassword")
3 print("mypassword".encode())
4 print(hashlib.sha256("mypassword".encode()))
5 print(hashlib.sha256("mypassword".encode()).hexdigest())
```

Authentication

Now let's improve auth.py

Authorisation

Authorisation typically involves giving the user some kind of pseudo-password that they store on their computer (client-side) which is a shortcut method for authorising a particular user.

An SSH key is an example of this.

Authorisation

What is a "token"?

A packet of data used to authorise the user.

What kind of tokens exist?

- **User ID:** The ID number of the particular user .
- **JWT'd User ID:** The ID number of a particular user stored in a JWT.
- **Session:** Some kind of ID representing that unique login event, whereby the session is tied to a user ID.
- **JWT's Session:** Some kind of ID representing a session that is stored in a JWT.

Authorisation

	User ID	**Session ID**
Non JWT	One login session + insecure	Concurrent login sessions + insecure
JWT	One login session + secure	Concurrent login sessions + secure

What is a JWT?

"JSON Web Tokens are an open, industry standard RFC 7519 method for representing claims securely between two parties."

They are lightweight ways of encoding and decoding private information via a secret

Play around:

https://jwt.io/

Let's practice with python

Using a JWT in python:
https://pyjwt.readthedocs.io/en/latest/

webtoken.py

```
1 import jwt
2
3 SECRET = 'sempai'
4
5 encoded_jwt = jwt.encode({'some': 'payload'}, SECRET, algorithm='HS256').decode('utf-8')
6 print(jwt.decode(encoded_jwt.encode('utf-8'), SECRET, algorithms=['HS256']))
```

Let's practice with python

Now let's improve auth.py