CS 410: Web Security
A2: Labs, Homework, and Program

WFP2: Authentication

- **Example #1**
  - **Default usernames and passwords are often left unchanged for many network devices and services.**
  - **This admin username and password is trivially guessed.**
- **Example #3**
  - **Cookies are often used as an authentication token that validates a client has authenticated in the past**
  - **Use your browser to reverse-engineer the cookie being used and write a Python script to obtain admin access to the site.**
- **Example #4**
  - **To hide the format of the cookie, cryptographic hash functions are sometimes employed. Weak hash functions such as md5, however, are easily brute-forced and several sites currently provide hash lookups that produce plaintext**
  - **Reverse-engineer the cookie format and write a Python program that sends an admin cookie to obtain admin access to the site.**
- **Example #5**
  - **Mismatches between the web application and backend databases can cause security errors**
  - **Case-sensitivity is one such conflict**
  - **The page is case-sensitive to usernames, but the database is not**
  - **Use this to register an admin user**
- **Example #6**
  - **Another mismatch is the treatment of whitespace between the web application and backend database**

- **Use this to register an admin user**

Homework

- **Lessons: Session Management**
- **Challenges: Session Management Challenges #1-6**
  - **Note: If attempting to solve Session Management #6 Challenge via a script, an additional cookie parameter must be added (ac=...) manually. (It is added in the browser via JavaScript so the Python script doesn't get it).**

Program #2 (WFP2: Authentication #2)

- **The authentication routine leaks timing information that allows adversary to guess characters of both the username and password**
- **Write a Python program that uses the vulnerability to automatically determine the username and password**
- **Note that both are alpha-numeric**