

# Confidential Weather

We will add an authentication step for our weather service.

It will be secure - an eavesdropping attacker will not be able to impersonate the user!

## Instructions

1. Use attached Weather server/client files and `confidential_weather_server.py` and `confidential_weather_client.py`. This is the solution for exercise `serializedweather` - the communication protocol is based on exchanging JSON messages.
2. Notice the server has one username/password defined in variable `USERS_DB`.
3. Make the client ask the user to enter the username and password via the command line interface (use python's `input()` function)
4. Add an authentication step before the client is allowed to ask for weather data. All steps will be with appropriate JSON messages exchanged between the sides.
  - 4.1. Client sends an authentication request, supplying the username as well
  - 4.2. Server responds with a challenge (a 10-byte random string)
  - 4.3. Client responds with response: a sha256 hash of the challenge concatenated with the user's password (`SHA256(Challenge+Password)`)
5. Try to keep the new JSON messages formats similar to existing JSON messages. You will encounter some problems - understand them and find the solutions!

## To submit

Submit server code `confidential_weather_server.py` and client code `confidential_weather_client.py`.