BONUS

Advanced Exercise: Cybersecurity Simulation

> If there is an AI detection on your Bonus submission, you will receive a grade of 0 for all of assignment 10. You have nothing to worry about if you're not using AI.

Bonus Exercise: "Crack the Code"

Create a Python script to simulate a basic brute force attack on a numeric password. This exercise requires effective use of both `for` and `while` loops in a cybersecurity context.

Exercise Instructions

Divide your script into the following functions:

1. `set_password(length [,seed])`:
    - Prompt the user to set the password *length* (number of digits).
    - Generate and return a random numeric password of that length.
    - Allow a random seed as a parameter.
2. `brute_force_attack(password [,start])`:
    - Accepts the password as a parameter.
    - Allows a starting number other than 0
    - Implement a brute force attack to 'crack' this password.
    - Use a `for` loop to iterate through possible combinations.
    - Use a `while` loop to keep the attack going until the password is cracked.
    - Return the cracked password and the number of attempts it took.
3. `performance_analysis(attempts)`
    - Accepts start time, end time, and total attempts as parameters.
    - Analyze and return the time taken and average attempts per second.
4. `main()`:
    - Combine all the above functions to simulate the entire brute force attack process.
    - Display success messages and performance analysis results.
    - Include a comment on the importance of strong passwords in cybersecurity.

Hints

- Utilize Python's `random` module for password generation.
- Utilize the seed function so that your test is always the same
- Be mindful of computational limits when setting password length.

Submission Requirements

Submit the Python script named `PythonH4CK3R.py` to the Final Bonus question.

The script should be well-commented, explaining the purpose and functionality of each function, especially in the context of simulating a brute force attack.