Week 8 Deliverables

Overview: This week, you have studied Web application vulnerabilities, password complexity, logs and cryptographic algorithms. The Lab for this week demonstrates your knowledge of this additional knowledge applied using Python functionality.

Submission requirements for this project include 2 files. (Zipping them into one file is acceptable and encouraged):

- Python Web Application Code (Python code for all routes, templates, static files and other files)
- Word or PDF file containing your test, pylint results and Cryptographic results.

Python Applications for this lab: (total 100 points):

1. **(50 points)** In this exercise you will update your web site to include a password update form and provide additional validation on the password check. Specifically you should create:

a. Password update Form – This Python form allows a previously registered user to reset their password after they have successfully logged in.

b. Authentication functions – These Python functions will check the following NIST SP 800-63B criteria are met upon password update:
   - Use the previous criteria for password length and complexity. (This work should already be done.)
   - Compare the prospective secrets against a list that contains values known to be commonly-used, expected, or compromised (Provided as CommonPasswords.txt).
   - If the chosen secret is found in the list, the application SHALL advise the subscriber that they need to select a different secret.

c. Logger – Create a log to log all failed login attempts. The Log should include date, time and IP address.

Hints:

1. Start early. This will take you longer than you think.
2. Leverage the File I/O, Flask and Data structures work previously performed in the class.
3. Use functions to enhance code reuse and modularity.
4. Use Python Lists or other data structures to store the Common Passwords and then appropriate search functions to expedite comparisons.
5. Use comments to document your code
6. Test with many combinations.
7. Use pylint to verify the code style – the goal is a 10!

2. **(30 points)** Using the Decrypting Secret Messages sites found in this week's readings, decrypt the following messages**.**

a. `- .... .. ... / ... -.. . ...- / ...-- ----- ----- / -.-. .-.. .- ... ... / .... .- ... / ... --- -- . / ... - .-. .- -. --. . / .-. . --.- ..- . ... - ... .-.-.-`

b. `U28gdGhpcyBpcyBiYXNlNjQuIE5vdyBJIGtub3cu`
c. `--- Psuwb Ysm ----`
   `W oa gc qzsjsf. Bc cbs qcizr dcggwpzm twuifs hvwg cih.`
   `--- Sbr Ysm ---`

Provide the decoded message along with the Cipher and any other parameters you used to solve each puzzle.

Hints:

1. Use the rumkin site
2. You will need to experiment some to narrow down the possible algorithms used. Some are more obvious than others.
3. You will know when you have selected the correct Cipher

3. **(20 points)** Document your results of the application running from your programming environment. You should also include and discuss your pylint results for the application. Provide your test results for each requirement in the Web application, associated functions and provide your resulting log files. Discuss the log file and how it could be used to possibly detect patterns of abuse. Describe the results of your NIST password complexity functions and how you tested each requirement. Include the Cipher tool results and write up in this document as well.

Any submissions that do not represent work originating from the student will be submitted to the Dean's office and evaluated for possible academic integrity violations and sanctions.