SYSC 4810A: Introduction to Software and Network Security

Assignment

Fall 2025

Prof. Hala Assal
Dept. of Systems and Computer Engineering Due: Dec 4, 2025

Assignment Instructions

You will practice topics relating to user authentication and access control mechanisms. A significant portion of this assignment requires you to do extra reading and research. For example, you will need to read up on cryptographic libraries and tools to support user authentication and access control such as Python `Cryptographic Services`. All sources used **must be documented** in the assignment solution. Programming portions of this assignment can be completed in any of the following programming languages: **C++, Python, or Java**. The use of AI-generated code and/or text is strictly prohibited and is considered an academic integrity violation.

Submission Requirements

You are responsible for ensuring that your assignment is submitted correctly and without corruption. For a complete assignment submission, make sure to submit all of the following to BrightSpace:

- **PDF Report.** Submit a PDF report (**3 pages max**) documenting your approach and design decisions. The label REPORT: will be used (below) to guide you whenever you need to include information in your report.
- **ZIP Archive of Source Code.** Submit a ZIP file of all the code you will write for this assignment. Files should be named according to the problem number, *e.g.,* `Problem1c.py`. Ensure that your files include sufficient code comments to describe what it does.
- **README File.** Submit a simple README file to describe how to run your code, taking into consideration the environment within which your code will be run (see below).

Grading Notes

- Your code and solution should follow Security best practices, as well as Software Engineering best practices (*e.g.,* no hard-coded values).
- Late assignments will be penalized 20% of the full grade per day *up to 48 hours* past the deadline.
- Failure of the submitted code to compile/run will result in 20% mark deduction, and can result in a grade of 0 for the assignment depending on code quality.
- Failure to submit any deliverable in the required format (PDF or ZIP) will result in a deduction of 5% of the full assignment grade.
- Failure to submit any of the deliverables (assignment PDF Report or Source Code files) can result in a grade of 0 for the assignment.
- **VIP:** Your submission will be graded on a machine running Ubuntu 24.04.3 LTS (Noble Numbat) with `g++` 13.3.0 (default language standard: `C++17`, also supported `C++03, C++11, C++14, C++20, C++23, C++98`), `Python` 3.12.3, and `openjdk` 21.0.8. It is your responsibility to ensure that your code will compile (for `C++` and `Java`) and run on such a machine.

Distribution of Marks

Question:	1	2	3	4	Total
Points:	14	17	16	11	58

Assignment Context

You have been contracted by a company called *justInvest* to design and implement a user authentication and access control system prototype for a proprietary software and data system. In this assignment, you will design and implement a prototype to fulfill the client's requirements and document your design decisions.
The following describes the types of *justInvest* users and the access control policy that must be enforced:

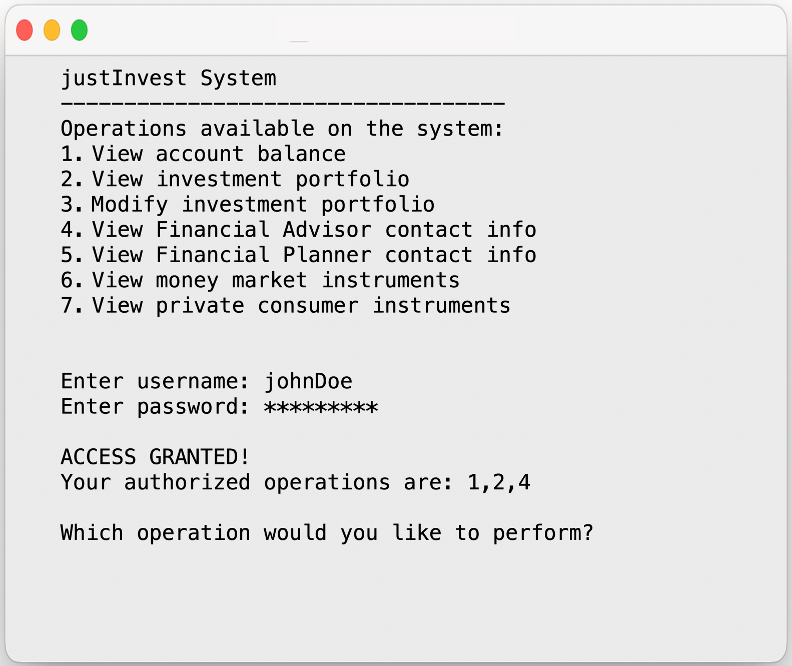
- Clients can view their account balance, view their investment portfolio, and view the contact details of their Financial Advisor.
- Premium Clients can modify their investment portfolio and view the contact details of their Financial Planner.
- All *justInvest* employees can view a Client's account balance and investment portfolio, but only Financial Advisors and Financial Planners can modify a client's investment portfolio.
- Financial Planners can view money market instruments[1].
- Financial Advisors and Financial Planners can view private consumer instruments[1].
- Tellers can only access the system during business hours from 9:00am to 5:00pm.

In addition to the access control policy, the prototype must implement a proactive password checker that ensures that all passwords adhere to the following password policy:

- Passwords must be between 8 and 12 characters in length.
- Passwords must include at least:
  - one upper-case letter
  - one lower-case letter
  - one numerical digit
  - one special character from the following: !, @, #, $, %, *, &
- Passwords found on a list of common weak passwords must be prohibited. Note that the list should be flexible to allow for the addition of new exclusions over time.
- Passwords matching the username must be prohibited.

justInvest has also expressed that the system should balance performance and security, and that the selected algorithms should not have any well-known weaknesses or vulnerabilities. The system should also require no code changes should the company hire a new employee, an employee's role changes, or a client's status changes.
justInvest have provided a sketch (Fig. 1) of what is expected from the system prototype. Once the user logs in, the prototype shall display a list of operations that the user is able to perform on the system (e.g., view account balance, view investment portfolio, ...). You are required to design and implement a prototype with a simple user interface to demonstrate that the system meets the client's needs.

[1]Examples of financial instruments, which are assets that can be traded, including cash or evidence of ownership of an entity



```
justInvest System
-----------------------------------
Operations available on the system:
1. View account balance
2. View investment portfolio
3. Modify investment portfolio
4. View Financial Advisor contact info
5. View Financial Planner contact info
6. View money market instruments
7. View private consumer instruments


Enter username: johnDoe
Enter password: *********

ACCESS GRANTED!
Your authorized operations are: 1,2,4

Which operation would you like to perform?
```

Figure 1: An example of a simple (incomplete) prototype for *justInvest* system.

To assist with testing, the client has provided the following sample list of employees and clients:

Name	**Role**
Sasha Kim	Client
Emery Blake	Client
Noor Abbasi	Premium Client
Zuri Adebayo	Premium Client
Mikael Chen	Financial Advisor
Jordan Riley	Financial Advisor
Ellis Nakamura	Financial Planner
Harper Diaz	Financial Planner
Alex Hayes	Teller
Adair Patel	Teller

Problem 1: Access control mechanism.

Consider the problem context outlined above and the access control policy that *justInvest* provided. You are required to design and implement an access control mechanism to control access to the various functions/objects in the system. To complete this problem, do the following:

(a) (2 points) **Select the access control model.** Choose an appropriate access control model (*e.g.,* DAC, MAC, RBAC) to be used in the development of your prototype.
REPORT: *List the chosen access control model and justify your answer.*

(b) (2 points) **Sketch a design of the access control mechanism.** Provide a sketch of the access control mechanism using your chosen access control model and an appropriate representation. Your sketch should show how *justInvest* 's access control policy is satisfied.
REPORT: *Include the sketch and annotate it, if needed.*

(c) (10 points) **Implement the access control mechanism.** Implement (using any programming language of your choice) the access control mechanism to ensure the user is only allowed to perform their authorized operations. Test your implementation.
REPORT: *Describe the test cases you used and show that they provide adequate coverage of the access control policy.*

Problem 2: Password file.

In this problem, you are required to create a password file, and design and implement functions for storing and verifying the passwords of users upon account creation and logging in, respectively. To complete this problem, do the following:

(a) (2 points) **Select the hash function.** Consider how password files are created in practice. Choose the specific hashing algorithm that you will employ in your password file mechanism, and determine all the relevant specifics.
REPORT: *List which hash function is used, and clearly justify your selection of all required parameters, including hash length, salt length, salt generation, etc.*

(b) (2 points) **Design the password file structure.** Consider information that you need to store in each record of the password file. Design and explain the structure of each record in your password file.
REPORT: *Provide an example records in the password file, and briefly describe its components and the reason for their inclusion.*

(c) (10 points) **Create the password file and implement relevant functions.** Create a password file based on your design in (a) and (b). Your password file should be a text file called `passwd.txt`. Implement two functions: one to properly add new records to the password file (when enrolling a new user), and another to retrieve records from it (when a user logs in).

(d) (3 points) **Test the password file usage.** Test that your password file works as planned.
REPORT: *Provide a description of the test cases you used to test the password file, and explain how these provide adequate coverage.*

Problem 3: Enrol Users.

In this problem, you will design and implement a mechanism to enrol users in the system. Assume for the purpose of this assignment, that the system allows users to enrol without another entity in *justInvest* pre-authorizing their enrolment.

(a) (5 points) **Design and implement a simple signup user interface.** Design and implement a simple user interface that enables a user to enter a username and chosen password, as well as any other information that may be required. Ensure that the interface only asks for information that is necessary for user authentication and access control.

(b) (8 points) **Design and implement the proactive password checker.** Design and implement the proactive password checker to be used when a user enrols in the system.

(c) (3 points) **Test the enrolment mechanism.** Test your enrolment mechanism and proactive password checker.
REPORT: *Provide a description of the test cases that you used to test both the enrolment mechanism and proactive password checker, and explain how these provide adequate coverage.*

Problem 4: Login Users.

In this problem, you will design and implement a mechanism to enable users to login to the system. You are not required to implement the systems functionality beyond what is described herein.

(a) (5 points) **Design and implement a simple login user interface.** Design and implement a simple login interface that enables an enrolled user to enter their username and password to access the system. Ensure that your implementation includes the password verification mechanism and uses relevant functions you designed in Problem 2.

(b) (3 points) **Determine and display the access privileges.** Once logged in, your implementation should display for the authentication user: their username, role/labels associated with this user, and a list of access rights or permissions according to the access control policy that *justInvest* provided.

(c) (3 points) **Test the user login and access control mechanism.** Test your implementation to ensure that it satisfies the access control policy.
REPORT: *Provide a description of the test cases that you used, and explain how these provide adequate coverage.*