SYSC 4810: Introduction to Network and Software Security
Module 3 Assignment

Fall 2021

Dr. J. Jaskolka
Carleton University
Department of Systems and Computer Engineering

Posted: October 7, 2021

Due: November 1, 2021

Due on Monday, November 1, 2021 by 11:59PM

This assignment contains 13 pages (including this cover page) and 5 problems. You are responsible for ensuring that your copy of the assignment is complete. Bring any discrepancy to the attention of your instructor.

Special Instructions:

1. **Do as many problems as you can.**
2. Start early as this assignment is much more time consuming than you might initially think!
3. The burden of communication is upon you. Solutions not properly explained will not be considered correct. Part of proper communication is the appearance and layout. If we cannot "decode" what you wrote, we cannot grade it as a correct solution.
4. You may consult outside sources, such as textbooks, but *any use* of *any source* **must** be documented in the assignment solutions.
5. You are permitted to discuss *general aspects* of the problem sets with other students in the class, but you must hand in your own copy of the solutions.
6. Your assignment solutions are due by 11:59PM on the due date and must be submitted on Brightspace.
   - Late assignments will be graded with a late penalty of 20% of the full grade per day *up to 48 hours past the deadline.*
7. You are responsible for ensuring that your assignment is submitted correctly and without corruption.

Problem	1	2	3	4	5	Total
Points:	25	25	25	25	10	110

In this assignment, you will participate in activities related to the operation and use of user authentication and access control mechanisms. This assignment aims to assess your understanding of security policies, as well as protocols to implement such policies. It also aims to assess your ability to develop basic security enhancements in stand-alone applications by implementing and using basic security tools to enhance and enforce user authentication and access control policies.

Background Research

A significant portion of this assignment is to do the required background research on working with basic cryptographic libraries and tools to support user authentication and access control such as Python `Cryptographic Services` and/or the C `OpenSSL` library. Keep in mind that a substantial component of any software or computer systems project is to solve and/or eliminate the underlying technical difficulties. This often means exploring user manuals and documentation.

Submission Requirements

Please read the following instructions very carefully and follow them precisely when submitting your assignment!

The following items are required for a complete assignment submission:

1. **PDF Assignment Report**: Submit a detailed report that carefully and concisely describes what you have done and what you have observed. Include appropriate code snippets and listings, as well as screenshots of program outputs and results. You also need to provide an adequate explanation of the observations that are interesting or surprising. You are encouraged to pursue further investigation beyond what is required by the assignment description.

2. **ZIP Archive of Source Code**: In addition to embedding source code listings in your assignment report, create and submit a ZIP archive of all programs that you write for this assignment. Please name each of your source code files appropriately to indicate the purspose of each file and. A simple naming scheme may be to name files according to the problem number to which they correspond (e.g., for Problem 7(a), the source code file should be named `Problem7a.c`). Your source code must compile and run in the VM environment, producing the desired output. Also, please remember to provide sufficient comments in your code to describe what it does and why.

3. **ZIP Archive of Screenshot Image Files**: In addition to embedding screenshots of program outputs and results in your assignment report, create and submit a ZIP archive of all of the raw screenshot images that you capture for this assignment.

Grading Notes

An important part of this assignment is following instructions. As such, the following grade **penalties** will be applied for failure to comply with the submission requirements outlined above:

- Failure to submit an Assignment Report will result in a grade of 0 for the assignment.
- Failure to submit the Source Code files will result in deduction of 10% of the full grade of the assignment.
- Failure to submit the Screenshot Image files will result in deduction of 10% of the full grade of the assignment.
- Failure of Source Code to compile/run will result in a grade of 0 for the corresponding problem(s).
  - *You are required to ensure that your code will compile and run in the VM!*
- Failure to submit any deliverable in the required format (PDF or ZIP) will result in deduction of 5% of the full grade of the assignment.

Part I Assignment Challenge

1 Introduction

Imagine that you are an employee of a computer security consulting firm. Because of the recent work you have been doing, your consulting firm has once again been approached and contracted by MedView Imaging, a medical imaging clinic that cooperates with local physicians offices and hospitals to provide diagnostic medical imaging services. MedView Imaging has requested the design and implementation of a user authentication and access control system prototype for their medical information management systems. You have been assigned as the lead developer for this contract and are responsbile for developing and documenting the prototype design and implementation to fulfill the contractual obligations of your consulting firm with MedView Imaging. The details of these contractual obligations are provided in the sections below.

The different parts of this assignment are designed to guide your investigation into the client's concerns. At the end of the assignment, you will be required to summarize your findings and provide recommendations to MedView Imaging addressing their concerns.

2 Context

MedView Imaging specializes in diagnostic medical images, with access to numerous medical images from a variety of imaging units. Imaging unit include ultrasound, computed tomography (CT) scans, X-rays (including mammography), magnetic resonance imaging (MRI), nuclear medicine imaging, and positron emission tomography (PET). When patients are referred by physicians for diagnostic medical imaging services, radiologists at MedView Imaging perform the required imaging service and provide a preliminary diagnosis. The images that are obtained from the service are stored in a central database, along with a short description indicating the nature of the diagnosis from the radiologist. The database contains protected health information (PHI), which refers to the demographic information, medical histories, test and laboratory results, mental health conditions, insurance information and other data that a healthcare professional collects to identify an individual and determine appropriate care. MedView Imaging seeks to have a new user authentication and access control system for their medical information management systems. Details of their previous user authentication and access control system have not been provided.

It is clearly stated in the contract that the following access control policy must be enforced:

1. Patients can view their patient profile (e.g., name and contact info), view their patient history, and view the contact details of their Physician.
2. All MedView Imaging employees (except for Technical Support) can view a patients's profile, but only Administrators can modify a patient's profile.
3. Administrators can only access the system during business hours from 9:00AM to 5:00PM.
4. Physicians, Radiologists, and Nurses can view a patient's history and medical images.
5. Radiologists and Physicians can suggest a diagnosis which gets written into a patient's history.
6. Only Physicians can prescribe a treatment which gets written into a patient's history.
7. Technical Support can run diagnostic tests on imaging units to check on the status of the equipment.

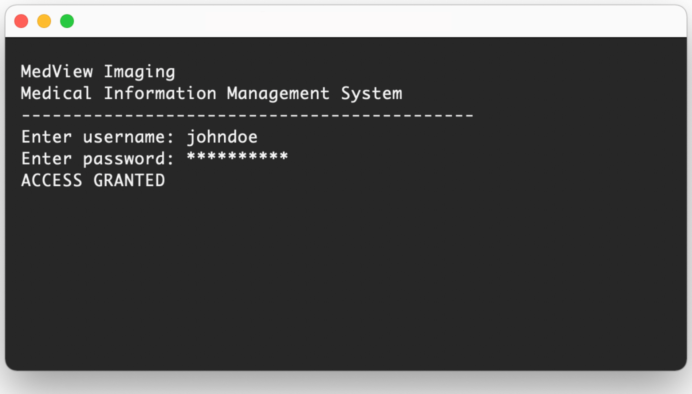
In addition to the access control policy, the prototype must implement a proactive password checker that ensures all passwords adhere to the following password policy:

- Passwords must be least 8-12 characters in length
- Password must include at least:
  - one upper-case letter;
  - one lower-case letter;
  - one numerical digit, and
  - one special character from the set: $\{!, @, \#, \$, \%, ?, *\}$
- Passwords found on a list of common weak passwords (e.g., `Password1`, `Qwerty123`, or `Qaz123wsx`) must be prohibited
  - *Special Note*: The list should be flexible to allow for the addition of new exclusions over time.
- Passwords matching the format of calendar dates, license plate numbers, telephone numbers, or other common numbers must be prohibited
- Passwords matching the user ID must be prohibited

In addition to the access control and password policies described above, MedView Imaging has expressed the following requirements and constraints of their system, which must be considered in the eventual design and implementation of the prototype.

1. A balance between performance and security is required.
2. Selected algorithms should not have any well-known weaknesses or vulnerabilities.

MedView Imaging has provided a sketch of what is expected from the prototype system (see Figure 1). Once a user logs in, the prototype shall display a list of the operations that the user is able to perform in the system (these do not need to be implemented). MedView Imaging has expressed that the prototype does not require any "fancy" user interface as they have already contracted another firm for that purpose. Instead, they are interested specifically in the design and implementation of the security mechanisms and a prototype that is able to demonstrate that the system will meet their needs.



```
MedView Imaging
Medical Information Management System
-----------------------------------------
Enter username: johndoe
Enter password: **********
ACCESS GRANTED
```

Figure 1: Sample prototype interface for the MedView Imaging medical information management system

To assist with testing and validation of the system, MedView Imaging has provided the following sample list of employees and clients:

Name	Role
Howard Linkler	Radiologist
Veronica Perez	Radiologist
Winston Callahan	Physician
Leslie Stewart	Physician
Kelsey Chang	Nurse
James Preston	Nurse

Name	Role
Rahul Garza	Patient
Arjun Singh	Patient
Heather Anderson	Administrator
Keikilana Kapahu	Administrator
Harold Zakara	Technical Support
Malina Romanova	Technical Support

3 Obligations

At the end of this assignment, you will be required to deliver the following information and outcomes to MedView Imaging:

1. Provide a detailed report documenting the design choices and details of the prototype implementation. This is necessary to enable MedView Imaging to make important decisions about whether to proceed with the implementation of the prototype.
2. Provide a functioning prototype system. You must demonstrate and provide a convincing argument that the system satisfies all of the requirements outlined by MedView Imaging.

Part II Environment Setup

This assignment will be conducted using a pre-built virtual machine (VM) image. We will assume that you already have a virtual machine set up from the Module 1 Assignment. If you have not yet completed the Module 1 Assignment, you will need to do so before continuing with this assignment.

Important Notes

1. It is essential that you set up the virtual machine as early as possible to ensure that you have time to address any technical difficulties that you may face. The instructor and the TA will not be able to provide adequate technical support close to the assignment due date.
2. ***You are required to ensure that your code will compile and run in the VM.*** This requirement provides a common platform to assist in the assessment of the assignment solutions.
   - You may use any programming language that you like, provided that it compiles and runs in the VM without any modifications to the VM. Popular choices include Python and C and this assignment has been written to provide some guidance for these choices.
     - The VM supports Python 3.8.5 and `gcc` 9.3.0.
     - The VM does not support Java.
   - While there may be additional cryptographic libraries that are available, the VM image is built with standard libraries for a variety of languages that should be suitable for conducting the assignment tasks.
3. It may be very tempting to "borrow" source code for this assignment. You may consult outside sources, such as textbooks or other resources, and documentation as part of your background research for this assignment, but you are expected to complete the assignment (i.e., design and implement the prototype system) on your own. ***Any use of any source must be documented in the assignment solutions.***

Part III Designing and Implementing a User Authentication and Access Control Prototype

1 Introduction

The principal objectives of computer security are to prevent unauthorized users from gaining access to resources, to prevent legitimate users from accessing resources in an unauthorized manner, and to enable legitimate users to access resources in an authorized manner. We can view user authentication and access control as two of the most central elements of computer security. In most computer security contexts, user authentication is the basis for most types of access control and for user accountability. User authentication encompasses two functions. First, the user identifies themself to the system by presenting a credential, such as user ID. Second, the system verifies the user by the exchange of authentication information. Once authenticated, authorization decisions can be made for the user. This involves enforcing an access control policy that determines the set of rights or permissions of the user to access a system resource or perform a specific system function.

In this part of the assignment, you will perform a variety of tasks that are designed to guide you in the design and implementation of a software prototype of a user authentication and access control system to satisfy the needs of MedView Imaging as outlined in Part I.

2 Background

2.1 Maintaining a Password File

A widely used password security technique is the use of hashed passwords and a salt value stored in a password file. The following procedure is employed (see Figure 2).

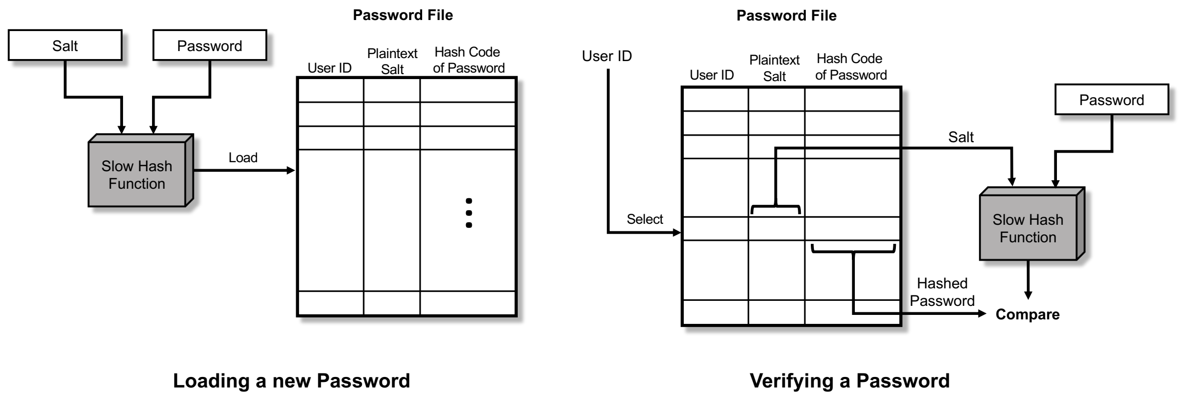


Figure 2: Loading and Verifying an Password in a Password File

2.1.1 User Enrolment: Loading a Password

To load a new password into the system, the user selects a password. This password is combined with a fixed-length salt value. In older implementations, this value is related to the time at which the password is assigned to the user. Newer implementations typically use a pseudorandom or random number. The password and salt serve as inputs to a hashing algorithm to produce a fixed-length hash code. The hash algorithm is designed to be slow to execute in order to thwart attacks. There are a number of hash functions that can be used for this purpose. Popular choices often include MD5 (even though it has been found to suffer from extensive vulnerabilities), variants of SHA, and Bcrypt. The hashed password is then stored, together with a plaintext copy of the salt, in the password file for the corresponding user ID.

2.1.2 User Verification: Verifying a Password

When a user attempts to log on to the system, the user provides an ID and a password. The system uses the ID to index into the password file and retrieve the plaintext salt and the encrypted password. The salt and user-supplied password are used as input to the encryption routine. If the result matches the stored value, the password is accepted.

2.1.3 Password Files

In many UNIX-based systems, the `/etc/passwd` file is a text-based database of information about users that may log into the system. The `/etc/passwd` file typically has file system permissions that allow it to be readable by all users of the system, although it may only be modified by the superuser or by using a few special purpose privileged commands.

The `/etc/passwd` file is a text file with one record per line, each describing a user account. Each record consists of seven fields separated by colons. The ordering of the records within the file is generally unimportant. An example record may be:

```
jsmith:saltedhash:1001:1000:Joe Smith,Room 1007,(234)555-8910,email:/home/jsmith:/bin/sh
```

The fields, in order from left to right, are:

- `jsmith`: User name: the string a user would type in when logging into the system; it must be unique across users listed in the file.
- `saltedhash`: Information used to validate a user's password; this field would typically contain a cryptographic hash of the user's password (in combination with a salt).
- `1001`: User identifier number, used by the system for internal purposes; it need not be unique.
- `1000`: Group identifier number, which identifies the primary group of the user.
- `Room 1007...`: Commentary that describes the user; typically, this is a set of comma-separated values including the user's full name and contact details.
- `/home/jsmith`: Path to the user's home directory.
- `/bin/sh`: Program that is started every time the user logs into the system. For an interactive user, this is usually one of the system's command line interpreters (shells).

Note that for this assignment, you may not require all of the above-listed fields in your password file.

2.1.4 Salted Hashing

There are a number of libraries that can be leveraged to implement salted hashing for the purpose of loading and verifying passwords. A starting point for various choices of programming languages include:

- Python Cryptographic Services: Useful libraries include: `hashlib`
- C OpenSSL Library: Useful libraries include: `<openssl/sha.h>`, `<openssl/evp.h>`

Note that these libraries are available in the VM.

2.2 Password Policies and Proactive Password Checkers

A promising approach to improved password security is a complex password policy and the use of a *proactive password checker*. In this scheme, a user is allowed to select their own password. However, at the time of selection, the system checks to see if the password is allowable and, if not, rejects it. Such checkers are based on the philosophy that, with sufficient guidance from the system, users can select memorable passwords from a fairly large password space that are not likely to be guessed in a dictionary attack.

2.3 Implementing Access Control Mechanisms

An access control policy, which can be embodied in an authorization database, dictates what types of access are permitted, under what circumstances, and by whom. Access control policies are generally grouped into the following categories:

- *Discretionary access control (DAC)*: Controls access based on the identity of the requestor and on access rules (authorizations) stating what requestors are (or are not) allowed to do. This policy is termed discretionary because an entity might have access rights that permit the entity, by its own volition, to enable another entity to access some resource.
- *Mandatory access control (MAC)*: Controls access based on comparing security labels (which indicate how sensitive or critical system resources are) with security clearances (which indicate system entities are eligible to access certain resources). This policy is termed mandatory because an entity that has clearance to access a resource may not, just by its own volition, enable another entity to access that resource.
- *Role-based access control (RBAC)*: Controls access based on the roles that users have within the system and on rules stating what accesses are allowed to users in given roles.
- *Attribute-based access control (ABAC)*: Controls access based on attributes of the user, the resource to be accessed, and current environmental conditions.

These four policies are not mutually exclusive. An access control mechanism can employ two, three, or even all four of these policies to cover different classes of system resources.

There are also a number of ways in which to represent an access control policy. This could include access control matrices, access control lists, policy rules stores (boolean functions), or some combination of these. In each of these cases, the access control policy is represented by a *data structure*. It is important to remember that each representation has strengths and weaknesses, and the choice of which to use must be carefully selected and implemented to achieve the goals of the access control mechanisms and to satisfy other system requirements.

3 Problems and Tasks

Important Notes

1. This is largely a design-focussed assignment, which means that there may be many suitable (alternative) solutions to the problems. You should focus your attention on providing adequate justification of the design decisions that you make.

2. You may need to work on more than one problem at a time, as the following tasks cannot necessarily be clearly delineated and carried out independently. You are encouraged to consider how your solutions to one problem may affect how to approach another problem. For this reason, it is highly recommended that you carefully read and understand what is required for each of the following problems and tasks before you begin.

3. When developing your solutions for each of the following problems and tasks, you are encouraged to consider and incorporate the fundamental security design principles into your solutions to the best of your ability.

Common Pitfalls to Avoid

- Make sure that you provide full and detailed explanations of what your code is doing. DO NOT respond to problems that ask you to "implement X" with "I implemented X, here is a screenshot of the whole code for X." Provide smaller code fragments and point to specific line numbers in your screenshots to explain your solutions.

- Make sure you provide a convincing argument that you have tested your code. DO NOT respond to problems asking how you tested your code with "I tested X with many values." Provide a description of what those values are, and how and why you chose them.

- Make sure you read the problems and tasks carefully and fully understand what is expected. DO NOT over-complicate the problem. Stay focussed on the tasks and what they are asking you to do. It is very easy to make this assignment much more difficult than it intends to be.

- Make sure you give yourself plenty of time to complete this assignment. DO NOT try to complete this assignment in its entirety close to the deadline. Open-ended design problems with implementations are time-consuming and require some thinking and planning.

Problem 1 [25 points]

Design the Access Control Mechanism: Consider the problem context outlined in Part I, and in particular, the access control policy provided by MedView Imaging. In this problem, you are required to design and implement an access control mechanism that is suitable for control access the various functions/objects in the system. At this stage, assume that you are able to authenticate users. 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, ABAC, some combination) to be used in the development of your proptype. Justify your selection.

(b) [3 points] **Select the access control representation**: Choose an appropriate representation for your access control policy control model (e.g., access control matrix, access control list, policy store, etc.) to be used in the development of your proptype. Justify your selection.

(c) [5 points] **Sketch a design of the access control mechanism**: Provide a sketch of the access control mechanism using your chosen access control model from Task (a), and your chosen representation from Task (b). For example, if you choose to design a DAC model using an access control matrix, sketch what the access control matrix will look like for the access control policy provided by MedView Imaging. If you choose to design an ABAC model, specify the policy rules for the access control policy provided by MedView Imaging. If you choose to design an ABAC model, describe the security labels including levels and compartments for the access control policy provided by MedView Imaging.

(d) [10 points] **Implement the access control mechanism**: Provide an implementation of the access control mechanism that you designed in Task (c). Be sure to include code fragments in your report and to clearly describe how you implemented your design.

(e) [5 points] **Test the access control mechanism**: Provide a description of how you tested the access control mechanism.

Problem 2 [25 points]

Design the Password File: In this problem, you are required to design and implement a password file that is suitable for loading and verifying the passwords of system users. To complete this problem, do the following:

(a) [5 points] **Select the hash function**: Consider the procedure described in Figure 2). Choose the specific hashing algorithm that you will employ in your password file mechanism. Be sure to clearly justify your selection of all required parameters including hash length, salt length, salt generation, etc.

(b) [5 points] **Design password file record structure**: Consider the 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.

HINT: This may need to be done by considering the information you require for your access control mechanism designed in Problem 1.

(c) [10 points] **Implement the password file**: Provide an implementation of the password file that you designed in Tasks (a) and (b). Your password file should be a plain text file called `passwd.txt`. You should have functions to add records to the password file and to retrieve records from the password file. Be sure to include code fragments in your report and to clearly describe how you implemented your design.

(d) [5 points] **Test the password file mechanism**: Provide a description of how you tested the password file mechanism.

Problem 3 [25 points]

Enrol Users: In this problem, you are required to design and implement a mechanism to enrol users in the system. To complete this problem, do the following:

(a) [5 points] **Design a simple user interface**: Provide the design of a simple user interface that enables a user to enter their user ID and chosen password, as well as any other information that may be required. For this problem, you should ensure that you only ask for information that is necessary to operate the system, or that is required for user authentication and access control.

(b) [5 points] **Design the proactive password checker**: Provide the design of a proactive password checker to enforce the password policy outlined in Part I. For this problem, you may wish to use pseudocode or another appropriate tool to sketch the design of the password checker. Be sure to justify your design decisions.

(c) [10 points] **Implement the enrolment mechanism and the proactive password checker**: Provide an implementation of the enrolment interface and the proactive password checker that you designed in Tasks (a) and (b). Be sure to include code fragments in your report and to clearly describe how you implemented your designs.

(d) [5 points] **Test the enrolment mechanism and the proactive password checker**: Provide a description of how you tested the enrolment mechanism and the proactive password checker.

Problem 4 [25 points]

Login Users: In this problem, you are required to design and implement a mechanism to enable users to login to the system. To complete this problem, do the following:

(a) [2 points] **Design a simple user interface**: Provide the design of a simple login user interface that enables a user to enter their user ID and password.

(b) [10 points] **Implement the password verification mechanism**: Provide an implementation of the password verification mechanism that corresponds to the password file mechanism you designed in Problem 2. Be sure to include code fragments in your report and to clearly describe how you implemented your designs.

(c) [8 points] **Enforce the access control mechanism**: Upon successful user authentication, provide an implementation to enforce the access control mechanism developed in Problem 1. Once logged in, the following information must be displayed for the authenticated user: user ID, role/attributes/labels associated with that user and that have been used for access control purposes, and a list of the access rights or permissions according to the access control policy provided by MedView Imaging.

(d) [5 points] **Test the user login and access control enforcement mechanism**: Provide a description of how you tested the user login and access control enforcement mechanism.

Part IV Summary of Findings

1 Reminder: Obligations

You are required to deliver the following information and outcomes to MedView Imaging:

1. Provide a detailed report documenting the design choices and details of the prototype implementation. This is necessary to enable MedView Imaging to make important decisions about whether to proceed with the implementation of the prototype.
2. Provide a functioning prototype system. You must demonstrate and provide a convincing argument that the system satisfies all of the requirements outlined by MedView Imaging.

2 Problems and Tasks

Problem 5 [10 points]

Summary and Demonstration of Full Prototype: Write a summary of your design and implementation of the prototype. Your summary should include a detailed explanation of how to compile and run your prototype. It should also provide at least one sample use case, complete with clearly specified input data that needs to be entered to enroll a user and to subsequently login that user. Be sure to include appropriate screenshots after each step to demonstrate the operation of the prototype. For example, after enrolling the user in the system, you should show the resulting password file demonstrating that the user has been added successfully. This is necessary for MedView Imaging to evaluate your prototype solution.

Write this summary as if you are going to submit it to MedView Imaging. This means that it should be clear and concise. It should address all concerns outlined in the contractual obligations above, and also take into consideration the client's requirements and constraints.

END OF ASSIGNMENT