

Programming Project I
CS 6238 Secure Computer Systems
Due Date: March 29, 2007

For this project, you can work in a group of two.

In this programming project, you will implement a highly simplified “protected” document store. More specifically, you will implement an access control module that enforces specified access control policy for the documents. The focus of the project is on implementing an access control list mechanism but you will also implement space management associated with document storage. You can make the following simplifying assumptions in this project.

1. You will create a large file which is used to store documents that users create and update. This file is created by the document system at initialization time. You will keep track of where documents are stored in the file and what areas are free.
2. Documents can be accessed using a document system API. In particular, once a document is created, users can checkout a document when they have read access for it, and they can check-in a document back into the system if they have write access. In addition, the API includes calls that allow users to manage access control lists associated with documents. Both negative and positive access rights can be defined.
3. Users will authenticate to the document system using passwords. Once a user is authenticated, a user identifier (UID) is returned. All calls to the document system include the UID as the first argument. You can assume that the UID is unforgeable like a capability.
4. A certain user can create a group and allow other users to be part of the group. In fact, group members can be other groups as well. Access rights can also be associated with groups.
5. A document can be deleted by its owner only.

You will implement the access controller and the document storage systems along with code that tests the functionality provided by them. The test code will do multiple logins, obtain UIDs, create documents and groups, update access rights, check-in and checkout documents and clearly demonstrate that the system enforces the access policy that is defined. To make the testing more dynamic you should read simple commands from a file. By default, the owner of the document has both read and write access, but other users must be granted access by the owner (either to a specific user or to a group to which the user belongs).

The first task is to clearly define the document system API. Since you can use a programming language of your choice, the exact API calls may differ. You will then implement and demonstrate that the implementation works correctly. You are particularly required to pay close attention to secure coding practices (you may want to try fuzzing to ensure that there are no vulnerabilities such as buffer overflows).

You will need to submit a brief report that provides details of your design, implementation and testing that is performed by you. You will do a demonstration of your project to the TA who will ask you to handle a sequence of requests to the document system. A convenient user interface to the document system will help with such a demonstration. You should perform detailed testing and no test sequence will be made available in advance.

The evaluation rubric will be used:

- Expressive API that includes intended document access and access control functions (20 pts)
- Correct and complete implementation of document access system (20 pts)
- Correct and complete implementation of authentication and access control lists, including groups, positive and negative access rights (30 pts)
- Demonstration of key functions (20 pts)
- Quality of report (10 pts)

If your code cannot handle bad input or found to have other vulnerabilities, the maximum credit you will get will be 50 points.