CS 5431 – Computer Security Practicum Spring 2018

Project Overview

Instructor: Eleanor Birrell

Project Description

CS 5431 students are expected to participate in a group project to build a software system that has non-trivial security functionality.

You have considerable flexibility in choosing what system to build. But because the course project is intended to provide an opportunity for you to exercise the material covered in CS 5430 and CS 5431, projects are acceptable only if the following "gold standard" security elements are necessary for the system to fulfill its mission:

Authentication. The system must authenticate its clients and/or allow its clients to authenticate the system. Clients may include human users as well as programs executing on machines. You must implement reasonable means of authentication, which may include passwords, user registration, generation of secrets, distribution of keys, etc.

Authorization. The system must enforce some non-trivial authorization policy to control some subset of its operation. Some systems will require DAC; others will require MAC. Implement an access control mechanism that is appropriate and natural for your system's functionality and expected scale.

Audit. The system must provide infrastructure for audit or other means of establishing accountability for actions. The security of that infrastructure must itself be ensured.

Projects must also intrinsically require information security:

Confidentiality and Integrity. The system must involve information that resides in long-term storage or that is transmitted over a network. The system's mission must require that information to be kept secret and/or be protected from corruption.

The list of essential security elements above defines only a subset of the security functionality your project will implement. What is the rest of that functionality? Answering that question will be the primary task of Milestone 1. If you need some ideas, a list of example projects is available on the course websites.

Example Projects

Here are sketches for a few example systems that could involve all of the above essential elements. Each sketch has important elements missing, as befits a sketch. Nevertheless, each could be refined into an acceptable course project, and you should feel free to do so. But also feel free to invent your own project idea!

Secure Anonymous Communication. This system enables users to communicate with each other secretly, accurately, and anonymously. Users can specify what information other users may learn about them and their communications.

Electronic Voting System. This system enables users to privately express their preferences about some issue. The system produces a verifiably correct aggregate of all the users' preferences.

Grade Management System. This system allows student grades to be stored by course staff, which may include TAs and professors, and to be retrieved by students. Grade information is stored in a back-end file system.

Multi-player Game Service. This system might implement a game, where clients are players; or it might implement a virtual world, where clients control participants. There might or might not be a back-end server.

Password Manager. This system allows users to create and store usernames and passwords for other systems. Users could manage their passwords across different devices.

Implementation

You must use Java to implement your system. Java prevents buffer overflows and other vulnerabilities, and we will be using source-code analysis tools that work only with Java.

Your system should be designed for public distribution. It should run on Ubuntu 16 LTS, and all milestones should be accompanied by a README.txt file containing installation instructions and, as necessary, installation scripts. If users cannot run your system, they can't use it. If we cannot run your system, we can't grade it.

When building a system in industry, it is generally a good idea to extend existing components rather than build your own. For example, there are many third-party systems and tools available for building web services. But using these tools in CS 5431 would preclude activities the project is supposed to cover. This is because, when you use a third-party tool, you must (i) accept somebody else's choices about what is useful security functionality and (ii) accept somebody else's assurance argument. We therefore impose the following rules about using code or systems written by others:

- Java's standard libraries (i.e., those part of the distribution) may be used. This includes various cryptographic routines, which you shouldn't be writing yourself anyway.

- GUI builders that are part of, or plug into, Eclipse or NetBeans may be used.

- Operating systems installed on the CS department lab machines may be used. This includes the networking infrastructures and file systems native to those operating systems.

- Database management systems that function as local, library-level services may be used. This includes the Java interfaces to Berkeley DB and SQLite. However, databases that run as separate servers and are accessed over the network may not be used. You should be able to understand and justify any security assumptions adopted by the system(s) you use.

- Existing web browsers, web servers, or any other web services infrastructure may *not* be used. These technologies make too many security decisions for you.

- Of course, the above rules are incomplete. If you believe it makes sense to incorporate other third-party code into your project, you must (i) obtain the instructor's prior, written approval, (ii) confirm that the license of that code is amenable, and (iii) clearly acknowledge the source of that code in your documents and demos.

Groups

Part of the purpose of this course is to give you experience in building software, including engineering its security, as a member of a development team. Why? Because...

- Working in a group offers you the powerful tool of discussing ideas with others.

- Working in a group affords the opportunity for parallel development activities and specialized expertise.

- Working in a group helps hone skills needed to be effective in the workplace (where groups are the norm) and impresses potential employers.

- Working in a group enables you to complete a more ambitious course project.

All members of your group are ultimately responsible for understanding all security aspects of the system you build.

Group size. Your group must start with 3-5 members. If through attrition your group size becomes too small, personnel may be re-assigned by "Management" (i.e., the instructor) from another group.

Milestones

The majority of the work—and the grade—for this course will come from the course project. This project is broken down into six phases. For each milestone, you will submit a written report and a working implementation of the project (up through that milestone). For Milestones 2-4 you will also demo your project for the class, and you will present your completed project after you submit Milestone 5. Detailed descriptions of each milestone will provided later, but a high-level description is provided below:

Milestone 0: Groups. Form a group of 3-5 students and submit a one-paragraph project idea. Due: February 7, 2018.
Milestone 1: Project Proposal. Describe the project you plan to implement this semester. You will also provide a detailed list of the security goals you plan (or hope) to implement. Due: February 14, 2018.
Milestone 2: Project Prototype. Implement some core functionality for your system. Due: March 7, 2018.
Milestone 3: Authentication. Implement the authentication-related security goals for your project. Due: March 28, 2018.
Milestone 4: Authorization. Implement the authorization-related security goals for your project. Due: April 18, 2018.
Milestone 5: Final Project. Implement all remaining functionality and security goals for your course project. Due: May 9, 2018.

Grading

Your project grade will comprise 98% of your course grade in CS5431. The remaining will be other factors, including participation. Grades will be assigned as follows:

- Milestone 0 (2%): Milestone 0 will be assigned a pass-fail grade.
- Milestone 1 (10%): Your project proposal will be assigned a letter grade, with the same grade being given to all members of the group. If this document seems incomplete or seems to be taking your group in the wrong direction, we may invite you to resubmit it with substantial improvements.
- Milestones 2-4 (10% each): For each milestone, you will submit your completed implementation and written documentation for your project. You will be evaluated on the completeness and quality of your documentation, the completeness and security of your implementation, and the ease with which we can get your project running. For each milestone, your group will be assigned a letter grade, with each group member receiving the same grade.

- Demos for Milestones 2-4 (2% each): In-class demos will be assigned a pass-fail grade. You will pass if you have a working prototype that demonstrates adequate progress since the last milestone, and if you present that prototype coherently. Failure to show up will result in a failing grade.

- Final Project (30%): You will submit your completed implementation and written documentation for your course project. You will be evaluated on the completeness and quality of your documentation, the completeness and security of your implementation, and the ease with which we can get your project running. The completed implementation for each milestone is assigned a letter grade, with each group member receiving their own (possibly different) grade. That grade will be determined in part by the quality of your completed project, the quality of your completed documentation, and your peer reviews. This grade will also be influenced by the originality, difficulty, and non-artificiality of your project.

- Final Presentation (10%): You will present and demo your completed course project. Your final presentation will be assigned a letter grade that reflects the quality of the presentation and the ease with which each group member answers questions about the security aspects of your project. All group members are responsible for being familiar with all components of the project.

- Challenge Factor (10%): For those students who truly seek an extra challenge, we include this factor as part of the grade. Most projects receive a challenge factor of around 0-5%. However, by making your project especially original or difficult, or by providing that extra "wow" factor, you can increase the challenge factor we assign to your project at the end of the semester. One way to increase your challenge factor is to learn and implement a cryptographic protocol from the research literature, or to (successfully) secure your system against a more challenging threat model. Another way to increase your challenge factor is to build a system that is new and exciting. New means that your system should have some aspect that is novel, rather than just replicating some existing system. Exciting means that your system should have some aspect that provokes a response of "hey, that's cool!". In the ideal case, the exciting aspect of your software is also new. In that case, you might become rich and famous. :-)

MEng Project Option

MEng students may use their CS 5431 project as the basis for the required MEng degree project. If this is your intention, your group members must all be MEng students who are all electing to use the project in this manner, and you must notify the instructor in your Milestone 0 submission. Note: if you select this option, expect the project to be significantly more work. Your group will be expected to go above and beyond the requirements for the CS5431 course project as agreed with the instructor.