SCC.363 Security and Risk Coursework
Part 1

This groupwork is to be completed as part of the SCC.363 module and is designed to create an inductive reasoning for security and better appreciation of security in technical systems. This coursework comprises 30% of the overall grade for this module.

Prerequisites

- Programming: Java is recommended, but Python implementations will also be accepted;
- Recommended libraries for use:
    - Java Security Libraries https://www.oracle.com/java/technologies/javase/javase-tech-security.html
    - The Legion of the Bouncy Castle https://www.bouncycastle.org/
    - The GNU Crypto project https://www.gnu.org/software/gnu-crypto/
    - Pycrypto for Python https://pypi.org/project/pycrypto/
    - Cryptography for Python https://pypi.org/project/cryptography/

Learning Objectives

- To develop a better and informed appreciation and understanding for security tools and systems;
- To formulate appropriate methods for troubleshooting and apply relevant theoretical concepts to identify and solve problems;
- Evaluate research and different types of information and evidence arguments critically;
- Synthesise and select appropriate information from a number of sources;
- Structure and communicate ideas effectively in writing.

Scenario

Your team has been approached by a (medical) service provider (such as a hospital) to help develop a secure Authentication, Authorisation, Accountability (AAA) service around their operations and data (electronic health records). Presently, the service provider has a regulatory body that oversee their operations. The main categories of users that require access to the data are patients, hospital staff, and the regulator.

The main system requirements and features expected to be developed are:

Task 1: System Design (weight 25%)

You should prepare a design document to provide a high-level solution to the problem presented. Aim to design a secure system, justifying your design choices. The information should be detailed enough that somebody who already understands the problem could code the project without having to make any significant design decisions. Examples of operations to be described are:

- A potential key-exchange/password scheme, password update policies, protocols, encryption/decryption schemes, algorithms and other relevant technologies to satisfy the AAA service;
- A description of how your system may be protected against potential attacks (e.g. replay attacks, poor password/guessing attack, access revocation, malicious insider, threats to data storage in transit and at rest) should be included and discussed in the design document.

Note: You should have a design document by end of Week 11. However, you may update it over the next weeks to depict any changes in your final system.

Milestone

Week 11: By the end of this week you should have a basic design document detailing the above requirements that can be updated over the coming weeks to depict any changes in your final system.

Task 2: Registration and Authentication Scheme (weight 25%)

You should implement a registration service and an authentication scheme for your system. Below is a list of indicative operations that should be supported by your developed services.

- Support for signup and login operations;
- Support for password strength evaluation to prevent weak/common passwords;
- Support for multi-factor authentication (e.g. One Time Password);
- Session-key and credential negotiation. You could have support for multiple schemes or hybrid schemes by public and private key encryption schemes, e.g. Message Authentication Codes (MACs), hash functions and signatures;
- Full message structure that has been decided upon by your group.

Milestone

Week 12: By the end of this week, you should have implemented a basic client/server supporting the above functionality.

Task 3: Authorisation and Secure Data Exchange (weight 25%)

You should implement an authorisation service on top of your authentication scheme to let users of the system access data based on their assigned permissions. Below is a list of indicative operations that should be supported by your developed services.

- Preferably a role-based access control model and policies should be adopted and used. Other models/policies could be used, if justified appropriately;
- After a successful authentication and session credentials negotiation, clients/users should be able to access (create, read, update, and/or delete, etc.) based on the permissions assigned to their role;
- All available roles and assigned permissions should be detailed and implemented for submission.

Milestone

Week 13: By the end of this week, you should have implemented a more advanced client/server supporting the above functionality.

Task 4: Accountability, Audit and Logs (weight 25%)

An accountability system should be implemented to retrieve appropriate information from all implemented services/operations in the system. Below is a list of indicative operations that should be supported by your developed services.

- Audit logs should be created for various (sensitive) activities or data access within the system;
- The structure of the access logs should contain relevant fields to ensure accurate reconstruction of events within the system;
- Access logs should have the right read/write permissions to ensure they cannot be tampered (deleted or destroyed). For example, logs can be append-only data structures (containing hash digest of preceding entry in the log).

Milestone

Week 14: By the end of this week, you should have implemented an advanced AAA client/server that supports the above functionality.

Submission Information

Assuming that you have hit the milestones outlined in this document, you can use Week 15 to finalise your design document, implemented services, and any other supporting material. The contribution(s) of each member (name/surname and ID number) should be clearly stated in the report under a section entitled "Members' contributions". Individuals that do not participate in the group effort may lose marks or get a zero mark. A readme file should describe any dependencies of the built system and how this can be run/tested.

ONE member from each group should upload the report and all supporting material on Moodle by **Friday 12th February at 16:00.**