Distributed Systems
COMP90015 2015 SM2
Project 2 - Security

Project 2

The project involves adding security to your existing Chat application. You need to extend your existing Chat application to do so. Work in groups of 2 and bring code from your first project.

In this project you may make use of certificates, secure sockets, public/private keys and/or shared session keys depending on your approach.

You may need to read through the documentation provided by Oracle or elsewhere to see specific details on the API for working with public/private keys and other cryptographic functions.

Security Policy

Modify your chat system to implement the following:

- Communication between clients and the server should be secure.
- The server should remember each client's identity after the client has quit/disconnected, and should be able to authenticate identities upon connection.

Authenticated Identities

You need to devise new messages to allow a client to authenticate with the server. You may allow a guest identity to be upgraded to an authenticated identity when the client changes identities. It is up to you to decide how to do this.

Rooms owned by authenticated identities should not be set to ownerless when the authenticated client disconnects. Only an authenticated user should be able to send commands as the owner of a room.

The server need not persist authentication information, it may simply be kept in memory. Thereby if the server quits or crashes, all authentication information may be lost.

Working with Cryptographic Functions

DO NOT UNDERESTIMATE THE DIFFICULTIES OF WORKING WITH CRYPTOGRAPHIC FUNCTIONS.

Consider the fundamental operations that you may need to implement:

- secure socket layer
- generating a public/private key
- reading/writing keys from/to a file

- generating a shared key for encryption
- encrypting and decrypting data
- encoding encrypted data for transport

If you intend to use these things, develop and test your own wrapper classes that can do each of these things. Then when they are working, build them into your Chat program.

Technical aspects

Your chat client and server must be executable exactly as they were in the first project. No additional files should be needed.

Briefly explain in your report how the client authenticates on the command line.

Report

Your report (maximum cover page + 4 pages) should detail all aspects of your security implementation. Discuss the technology that you use and explain how it achieves the security objectives of the project. Discuss any limitations or additional benefits that the technology provides.

Provide details of the changes to your existing protocol.

You should submit your report plus your modified system; instructions will be given on LMS.

The due date is Friday Week 12, 23rd October.