Project Description

CS 161 - Joseph/Tygar/Vazirani/Wagner

3rd October 2005

Overview

The goal of this project is to design and implement a secure instant messaging program. Below are a set of functional requirements followed by security requirements of the program. Also, a Denial of Service (DoS) resistance requirement is specified. All aspects are strict requirements of the program, and will be tested thoroughly.

System Functionality

The program must include the following functional goals:

Account Each user has an account within the program. A user must use this account to perform messaging with other users.

Messaging Any user can send a text message to any other user on his or her buddy list.

Buddy List The program must maintain a list of other users that are allowed to communicate with the user. These other users are referred to as buddies. The user can only communicate with buddies on his or her buddy list. Thus, a user must be added to the buddy list before a conversation takes place. A user may remove buddies from his or her buddy list. This will block all future messages from the removed buddy to the user. The exact definition of "block" depends on implementation, but at the very least blocked messages should not be displayed.

Conferencing The program must contain support for conferencing between buddies. The user can specify two or more buddies to participate within a conference, and only those in the conference can see the conversation taking place. Thus, the user who initiates the conference, known as the administrator, has the power to invite to the conference when the conference is created. Additionally, any user participating in a conference has the ability to remove his or herself from the conference.

Security Requirements

The program must include the following security goals:

Authentication The owner of the account is the person who created it. Once an account is created, only the owner can access it unless the owner divulges secret information or otherwise disobeys the authentication protocol.

Confidentiality Sent messages can only be viewed by the buddy or buddies intended to receive them. Thus, no one other than the intended recipient(s) should be able to read messages from the sender.

Integrity Sent messages can not be altered by a third party. All messages leaving the sender must be protected from modification.

Denial of Service (DoS) Resistance

Assume a DoS adversary is present on the network where your instant messenger is running. The program must be able to withstand any attempts made by the adversary to prevent normal operation of the instant messenger. These attacks include, but are not limited to, consumption or overload of system or network resources, such as, bandwidth, disk space, or CPU cycles.

You may assume that the adversary's resources are similar to those of the computer(s) the adversary is attacking, so you do not need to worry about attackers that launch distributed denial of service (DDoS) attacks from machines outside of the instant messaging network. It is assumed that the adversary has the ability to control the network and one or two clients. The TAs will provide information on how this requirement will be tested in section.

Group Formation

Groups of four may be formed within your section to complete this project. Please email your TA with a list of names and email addresses of all group members along with a group name by the second discussion section. If you do not have a group, or are having any issues in forming a group of four, please contact your TA.

Design Review

Before implementing your design, your team will present its system architecture to your TA. Each team will submit their design documentation to their TA on October 14, 2005. Groups will present their design shortly thereafter. The presentation should cover the Security Goals, Threat Model, and Attack Analysis that your group produced when designing the system, and should describe the architecture that your group plans to implement. This will be an excellent opportunity for your TA to provide you with feedback on your design and answer any remaining questions your group may have.

Documentation

In addition to your design review document, your team will be responsible for providing extensive documentation of your project implementation. This final document must include your design documentation (revised from the design review if necessary), implementation details, appropriate "README" information, and a summary of test cases run against your project.

Implementation details

A skeleton IM client implementation based upon the Hamsam Java API will be provided. Each group will design and implement a secure network protocol to allow the clients to communicate. Specifically, each group will be implementing Hamsam's "Protocol" interface.

It is up to each group to design their own architecture, but it is recommended that a centralized server be used as part of the connection process instead of implementing a purely decentralized system.

A sample GUI and text based test program will be provided to each group shortly. A number of crypto libraries are freely available for Java 2, including Bouncy Castle (http://www.bouncycastle.org/) and Jessie (http://www.nongnu.org/jessie/). Jessie implements javax.net, javax.net.ssl, and javax.security.cert, while Bouncy Castle implements javax.crypto.

Feel free to use any of the API's provided with Java 1.5 to implement your project. If you come across any third party libraries that you think will be helpful, contact your T.A. Also, if you are uncomfortable using Java for this class, let us know.

This project is due October 31, 2005.