CS425 Mini Project

Chat Application

Group -29

**Introduction**

The aim of our project was to get hands-on experience in developing client-server messaging applications. Text message passing is the simplest form of communication over a network and only after understanding the basics of point to point message passing, broadcasting and asynchronous messaging can more complex applications be developed.

**Objectives**

We had started with the following objectives in mind –

1. Users must authenticate before they can message
2. No duplicate logins allowed
3. IP blocked after 3 failed login attempts
4. Features to determine who else is/was logged in
5. Send private message to online user
6. Offline users receive messages sent to them once they log in
7. Broadcast message to all currently logged in users
8. Block/unblock users
9. Asynchronous messaging

**Assumptions**

1. Only basic authentication used without encrypting passwords
2. If the target is offline , message to be stored in a txt file
3. At most 5 users allowed to be logged in concurrently

**Architecture**

Step 1 - Server starts

Step 2 - Clients can connect using username and password. Duplicate logins not allowed

Step 3 – Clients can use one of the following commands while logged in –

1. *Un > msg*  - Send message “msg” to usename “Un”

a. If username ”Un” is online , message is delivered

b. If username ”Un” is offline ,message stored in a txt file to be delivered later

2. *broadcast > msg* - Message “msg” is delivered to all current online users

3. *whoelse –* Displays list on currently logged in users

4. *wholasthr –* Displays list of users logged in in last hour

5. *block Un –* puts username “Un” in this user’s block list

6. *unblock Un –* removes username “Un” from this user’s block list

**Implementation environment**

1. Language – Python 2.7
2. Libraries – socket, select, thread

**Summary**

All of the required features from our objectives have been implemented with a couple of drawbacks. Authentication is done by reading from a txt file which is quite unsafe. This can be easily remedied by using a secure database instead of simple txt file. Also unsafe messages can be easily sent to harm/hijack server.

**Project members' details**

14433 Nishit Asnani

150740 Sunny Kant

14377 Mayank Patel