

INFORMATION SECURITY PROJECT REPORT BAATKARO

GROUP CHAT SYSTEM

OVERVIEW: BaatKaro system is a multi-user chat application designed for group communication in real time just like any other chat application. It allows users to create or join a group by taking their username as an input as well as group name. Other than that it also allows users to exchange messages, manage group memberships, approve requests, see group members and transfer files. The system uses client server architecture using TCP/IP sockets for communication.

COMPONENTS:

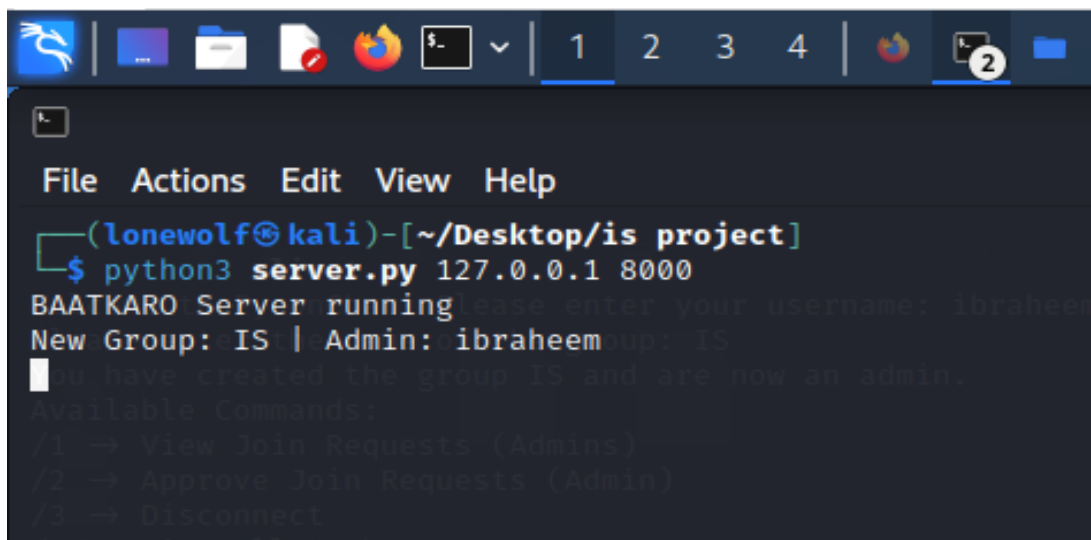
- **Server Side:**
 - Manages Group chat functionalities
 - Handles incoming client connections
 - Facilitates group creation, user authentication and message routing
 - Implements chat group management features such as approving requests, changing admin, kicking members, and file transfers.
 - Utilizes threading to handle multiple clients simultaneously
- **Client Side:**
- **Socket Client (client.py):**
 - Connects to the server for group chat functionalities.
 - Provides a command-line interface for user interactions.
 - Allows users to send messages, view group members, manage group requests, and initiate file transfers.
- **Client State management:**

- Handles client-side state, including user input processing, socket communication, and user interface logic.
- Uses threading to manage concurrent tasks such as listening to server messages and waiting for user input.

FEATURES:

- Server Features:

- **Group Management:** This feature manages and create chat groups. Other than that it is also responsible for handling of join requests, approvals and member removals. Admin rights can also be transferred to other user.
- **Message Broadcasting:** This feature allows user to send and receive message between group members. It also supports private and group-wide communications among the users.
- **File Transfer:** This feature allows user to upload and share file among the group and other users. Overall this feature is responsible for broadcasting of shared files to online users in the group.



```

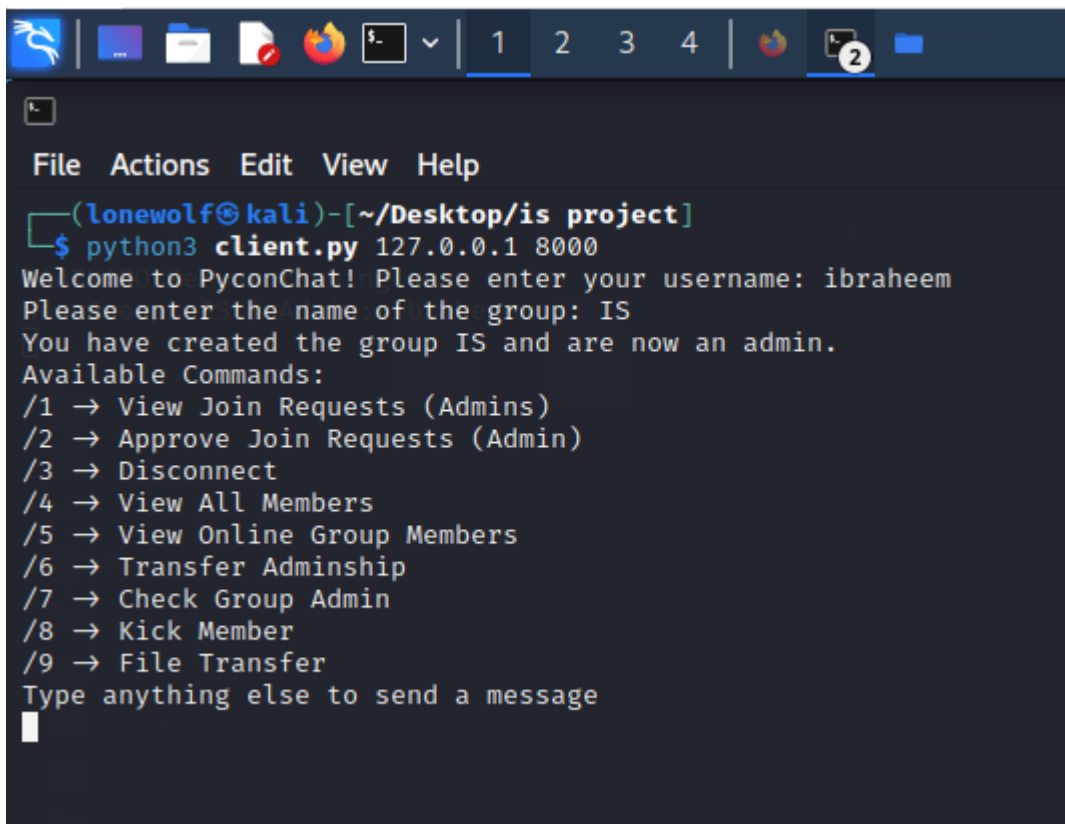
(lonewolf@kali)-[~/Desktop/is project]
$ python3 server.py 127.0.0.1 8000
BAATKARO Server running
Please enter your username: ibraheem
New Group: IS | Admin: ibraheem
You have created the group IS and are now an admin.
Available Commands:
/1 -> View Join Requests (Admins)
/2 -> Approve Join Requests (Admin)
/3 -> Disconnect

```

Server Side

- Client Features:

- **Interactive Interfaces:** This feature includes command-line interface for user interactions. It also includes input processing for various commands like sending messages, managing groups, viewing members etc.
- **Group Actions:** This client side feature includes viewing group members, online users and pending requests. This feature also allows admin to perform administrative tasks in a group like adding and removing a member.
- **File Operations:** This feature allows us to share, view and download file among the users or within the group.

A screenshot of a Kali Linux terminal window. The terminal has a dark background with a menu bar at the top containing 'File', 'Actions', 'Edit', 'View', and 'Help'. Below the menu bar, the prompt is '(lonewolf@kali)-[~/Desktop/is project]'. The user has entered the command '\$ python3 client.py 127.0.0.1 8000'. The output shows a welcome message, prompts for a username (ibraheem) and group name (IS), and a confirmation that the user is now an admin. A list of available commands is displayed, including /1 to /9, each with a description. The prompt 'Type anything else to send a message' is shown at the bottom with a cursor.

```
(lonewolf@kali)-[~/Desktop/is project]
$ python3 client.py 127.0.0.1 8000
Welcome to PyconChat! Please enter your username: ibraheem
Please enter the name of the group: IS
You have created the group IS and are now an admin.
Available Commands:
/1 → View Join Requests (Admins)
/2 → Approve Join Requests (Admin)
/3 → Disconnect
/4 → View All Members
/5 → View Online Group Members
/6 → Transfer Adminship
/7 → Check Group Admin
/8 → Kick Member
/9 → File Transfer
Type anything else to send a message
```

Client Side

USAGE:

1. Run **server.py** with specified I.P address and port number
2. Start the server.
3. Run the **client.py** with same I.P address and port number as of server.
4. Enter username and group name to create a group and become an admin.
5. Use command-line prompts to perform actions to manage the group chat.
6. Follow instructions to perform any action listed in the Command-line interface.

CONCLUSION:

BaatKaro group chat system provides a functional platform for collaborative communication within groups. It leverages socket programming and multithreading to support real-time interactions and file sharing. This project demonstrates the implementation of a simple but effective chat application using Python's socket library. User authentication and a Graphical User Interface can really enhance this project's performance and would make it much better.