Environment to run KaiwaChat:

Please use a Windows Operating system to run these executables.

Server.exe is the concurrent KaiwaChat server. (Source code is server.cpp)

Client.exe is the KaiwaChat client. (Source code is client.cpp). You can run this executable multiple times to open multiple clients.

There are no arguments necessary to run these executables. Have fun!

Analysis:

I think my assignment is fairly robust in that it follows the DFA states very well. For example, you cannot fake your way to the connected state without providing the right commands in messages (PING_PONG, VERSION_NEGOTIATION, and CLIENT_INFO). In this way, it is robust against purely random fuzzing. The code in both the client and server follows Boolean if/else nested logic that parallels the movement through the DFA. Moving through this logic, it is not possible for a user to reach their intended state without moving through the prescribed DFA path. However, my assignment would be vulnerable to a more adaptive fuzzing that follows the DFA and provides the right commands. If a malicious user knew the DFA, it would not be tough to crack through a targeted process of fuzzing and such a user could cause some undesired effects on the server's interactions.

I tested my implementation using both random fuzzing and a random fuzzing of PDU's with a targeted command field to get through the DFA.

Additional Information

Due to limitations in the terminal interface, typing a message and displaying received messages at the same time is not a feature that was implemented. Instead, client terminals that are in the Chat Server Mode DFA state will operate in two modes:

Reading messages mode:

Once a client joins a channel, the user will see any received messages as they arrive until they press the Space key to create their own message.

Writing messages mode:

Users will not see any received messages after the moment that they press Enter and start typing their own message. The client will receive new messages and display them to the terminal after the user has returned to the reading messages state. A user returns to the reading messages state by completing their message by pressing the Enter key.

Extra credit was not completed.