C2c Server

Design Overview

**Release Date**: December 4, 2019

**Creator**: Jeff Wilson

# Overview

The server is implemented on the C Socket API and uses very basic data structures. Once the server is created and the initial socket established, the server is put into a loop that hangs until an agent attempts to contact it. From there we branch as necessary to perform the correct actions.

# Structures

**Connected\_Agent**: This structure has 3 members.

* connect\_Start (time\_t): This hold the time in which the agent joined the server
* IP\_Addr (char\*): Hold the Agent’s IP address in a string.
* Active (bool): Determines whether the agent is still “connected”.

If Active is set to false, this lets the server know that this agent can be overwritten with one that is attempting to connect.

**DateAndTime**: This structure holds integers for the following values:

* Year
* Month
* Day
* Hour
* Minutes
* Seconds
* Milliseconds

Using the gettimeofday() function we can populate this struct to provide the time of events to the log file. The get\_server\_time() function handles the creation, formatting, and processing of this info.

# Active Agent List

The list of Agents is stored in an Array of type Connected\_Agent of size MAX\_AGENTS. For this code MAX\_AGENTS is defined as 10. In the server it is named “**Master\_Active\_Agents\_List**”.

# Main Loop

The main loop will always loop until the program is terminated externally by the user. The server reads the message from the agent (#ACTION) and does a series of string comparisons to see if it matches any of the valid commands. If a valid command is found then the server checks if the IP is already in Master\_Active\_Agents\_List as well if it is considered “Active”. If both are true, then the connecting agent is considered valid to ask for a log, list, or to leave. If an agent is just joining for the first time, it is added into the Array at a location that is not considered “Active”.

**Example of checking if an agent is active:**

if(strcmp(Master\_Active\_Agents\_List[index].IP\_Addr, IP\_str) == 0 && Master\_Active\_Agents\_List[index].Active == true) {

// Do Stuff

}

# Decisions I Made

* I decided to make the Master List be an array, so it was easy to iterate over. If we wanted o scale this to a larger capacity, I would implement a linked list.
* I also decided to store the IP as a string so I could use strcmp(). Since we will not have to use that IP object again there was no point in storing it, all our communication to the client was done via a new socket file descriptor (newsockfd).
* time\_t was used to store and calculate the time connected since it deals specifically with seconds and was more reliable than my original method of using clock().