Client-Server Reporting System

April 17, 2014

Your task is to write a client – server application. The client is in charge of analyzing router logs and sending questionable activities to the server. The server application will receive those reports from many clients simultaneously and store them in memory.

Requirements

Client

The client is a command-line application that will accept one or more log files as arguments.
It will establish one connection to the server and send all the questionable activities found. Then it will terminate.

Server

The server will process each connection in a separate thread. An extra thread will be printing to the screen all the activities received during the last 5 seconds.

Router logs format

The client must be able to parse logs in the following format (see attached log file):

```
device: 01-42-ff-ae-3b-25
url: https://google.com/search?q=cars
timestamp: 1397564362
device: 45-60-bd-51-1e-fe
url: https://porntube.com/
timestamp: 1397558823
device: 33-24-d6-85-0a-b2
url: http://en.wikipedia.org/wiki/Special:Random
timestamp: 1397572541
```

The log file contains a set of internet browsing events composed of three lines each. First line is the device MAC-address. The second line is the visited website. The third line is the time in seconds since the UNIX epoch when the event took place.

Questionable activities

An activity is deemed questionable when it contains an offending word in its domain name. The list of offending words consists of "porn", "xxx", "sex" and "Bieber". You must ignore case.

Communications

The client and the server will communicate through a TCP connection using a simple protocol of your design, with the following restrictions:

1. The server will listen to the port 42422.
2. The client will send all his events in a single connection.
3. For every event sent, the client will wait for a response by the server.
4. The server response will tell the client if the event has been stored successfully or not and the total number of events received by the server since it was started.

Libraries

Make good use of C++03 or C++11 standard libraries (at your choice). Use well-known libraries if needed. Any other library usage must be justified.

Deliverables

Source code for both applications.
Build instructions.

Evaluation goals

1. Requirements are satisfied.
2. Quality and neat code. Easy to understand and simple. Follows established best practices.
3. No memory leaks, bugs, race conditions, security issues, undefined behavior.
4. Portable code.