TITLE

GP200-Listener - TCP listener for incoming GP200 packets

VERSION

current version = 1.0
last updated on May 12, 2008

SUMMARY

This is a TCP listener that accepts incoming GP200 tracker packets, decodes the packets and stores the result in a database.

The listener accepts 2 types of connections :

* GP200 connections : tracker devices send GP200 packets to the listener
* management connections : telnet to the server and execute some simple management commands

The listener is based on the POE framework, since perl does not (yet) support 'real' multithreading POE uses sort of a multitasking framework.

INSTALLATION

A very simple makefile is provided to copy the files in /opt/GP200-Listener.
A sample init.d startup file will be copied to /etc/init.d to run the listener as a daemon.

Before running the script make sure the database is installed and the MySQL settings have been updated in the script !!

REQUIREMENTS

The listener requires the following perl modules :

* TrackerProtocol (v 0.02) - for decoding GP200 packets
* DBI - mysql database connection
* POE - TCP listener
* POE::API::Peek

INSTALLATION

As user 'root' (or sudo) :

* untar/unzip the package
* run make install

This has copied all the files to /opt/GP200-Listener and the startup script to /etc/init.d

DATABASE

Make sure a username / password and database are created before executing the init.db file.

Execute the initdb.sql to initialize the database. There are 2 tables created :

Events => actual table that contains all information
Events_debug => if the listener has DEBUG enabled then the payload of all packets that are received are stored in this table as well

Running the listener

The listener is written in perl so just run it as any other perl script :

```
perl gp200-listener.pl
```

There are several startup parameters possible, add --help to see them all. These parameters can also be configured directly in the script as a script parameter.

DAEMONIZED MODE

A sample startup script is provided to run the listener as a daemon :

```
/etc/init.d/gp200  start | stop
```

As an alternative the script can be started with the --daemon option but then it will be killed automatically when the terminal is closed.

SCRIPT PARAMETERS

The script has several configurable parameters in the top section of the script :

POEDEBUG = enable / disable extensive POE debugging
POETRACE = enable / disable even more POE debugging
VERBOSE = enable / disable verbose mode
LOGLEVEL = 0 - 9 -> DEBUG level where 0 means no output
SERVERPORT = default port that the server listens for incoming connections (default = 9999)
MAX_SESSIONS = number of maximum simultaneous incoming connections (default = the maximum number of allowed connections by the system)

MYSERVER = hostname of the SQL server
MYDB = MySQL database name
MYUSER = MySQL database username
MYPASS = MySQL database password
MYPORT = MySQL server port (default = 3306)
USEDB = enable / disable usage of database (default = enable)

ALLOWED_GPRS_IP = list of ip addresses that are allowed to connect and send GP200 packets (default = empty list, this means all addresses are allowed)
ALLOWED_MGMT_IP = list of ip addresses that are allowed to connect and send management commands to the server (default = only localhost is allowed)
BLOCKED_IP = list of ip addresses that are blocked, if they connect they will be immediately disconnected (default = empty list, this means no addresses are blocked)

SECURITY

There are a few security measures built in the listener :

* the maximum number of connections can be restricted
* lists of ip addresses can be configured that are either blocked or that define if they can send GP200 packets and / or management packets

MANAGEMENT CONNECTION

It is possible to connect to the listener via telnet and execute some basic management commands. The purpose is to find out how many active connections there are, how many known IMEI numbers are connecting etc. This way it's also possible to find out if a certain unauthorized ip address is trying to connect to the server.

Following commands are currently known :

quit = quit the current management session
shutdown = completely shut down the listener (remark: if the server was started via init.d startup script then it's better to shut it down via the init script)
info = print some info about the server, the number of connections etc.
help = print the available management commands