GTCP Sample

Installation

The sample was written in Visual Studio 2002.
You need to unpack the provided archive and open the ***GTCP 1.0 > Project.sln*** solution file. Then rebuild the solution.
Press ***F5*** to run both client and server in Visual Studio.

Configuration

You can edit all network settings in the main dialog of client and server applications.

Design

This applications is designed to show such Genuine Channels features as

- Traffic counters
- Connection status
- Security Session
- Re-establishing a connection if you unplug and then plug back the network cable
- Intercepting the traffic and Genuine Channels events
- Recognizing the restarting of the server
- Queue overflowing
- Compression
- Traffic encryption
- SSPI support, impersonation and delegation
- Performing requests in the dedicated threads
- Logging (the traffic is being intercepted directly from the log)

Known layer contains:

- Interfaces allowing to subscribe and receive an event
- Interface allowing to establish the specified Security Session
- Interface allowing to create file
- Built-in 256 bit Rijndael key

Server implements and provides all the necessary business objects and binds them to the known URI.
Server fires an event three times per second. The timer is located on the server dialog. You can cancel the firing of events by unchecking the corresponding check box.
You can stop the server, change the listening port and start the server again at any moment.

Client establishes a TCP connection with the specified server and subscribes to the event.
Client automatically updates traffic counters and calculates CPS value every 500 milliseconds. Client informs about channel's events and shows the traffic if you check the corresponding check boxes located on the third panel.
The controls on the second panel allow to establish a Security Session and make the server write a file.

Rendering events

Client application implements its own log writer to intercept and show the traffic (***IEventLogger*** interface). You can use built-in ***BinaryLog*** and ***FileLog*** classes for writing log information into a file.
Client subscribes to Genuine Channels events to indicate the network status and report about network events.

Security Session

The sample allows designing and creating ***Security Sessions*** at run-time. In general, you will not need this feature in your application. You will define several ***Security Sessions*** and use them in different contexts.

In order to check the appropriate ***Security Session***, you need to choose it and specify call parameters. The maximum error for the ***Timeout*** parameter is 10 seconds. The specified Security Session is applied only to the invocation that creates the file.
If you want to check ***SSPI***, you need to know several accounts at the server. Having created a file on the ***NTFS*** partition open the properties of the created file in ***Explorer***, choose the ***Security*** tab and see who has created it.
SSPI never sends password through the network. You can switch off server's events, enable traffic writing, create a file and then analyze the traffic.

Restarting the server

You will not be able to perform this test in Visual Studio. Open Client and Server ***bin\debug*** directories in the ***Explorer***.
Start the server and the client. Connect to the server from the client. Then close the server application and start it again. The client recognizes that the server has been restarted and re-subscribes the event listener again.
The connection can be broken due to queue overflow. You can disable the sending of events to prevent this.

Breaking the connection

Try to unplug the network cable and plug it back after a time. Either client will re-establish the connection or the connection will be closed due to the queue overflow.