Модуль подсистемы “Транспорты” <SSL>

Модуль:	SSL
Имя:	SSL
Тип:	Транспорт
Источник:	tr_SSL.so
Версия:	0.9.1
Автор:	Роман Савоченко
Описание:	Предоставляет транспорт основанный на слое безопасных сокетов. Используется OpenSSL и поддерживаютя SSLv2, SSLv3 and TLSv1.
Лицензия:	GPL

Оглавление

Введение

Модуль транспорта SSL предоставляет в систему поддержку транспортов основанных на слое безопасных сокетов (SSL). В основе модуля лежит библиотека OpenSSL. Поддерживаются входящие и исходящие транспорты протоколов SSLv2, SSLv3 и TLSv1.

Добавить новые входящие и исходящие транспорты можно посредством конфигурации транспортной подсистемы в любом конфигураторе системы OpenSCADA.

1. Входящие транспорты

Сконфигурированный и запущенный входящий транспорт открывает серверный SSL-сокет для ожидания соединения клиентов. SSL-сокет являются много-поточными, т.е. при подключении клиента создаётся клиентское SSL-соединение и новый поток в котором производиться обслуживание клиента. Серверный SSL-сокет, в этот момент, переходит к ожиданию запросов от нового клиента. Таким образом достигается параллельное обслуживание клиентов.

Каждый входящий транспорт обязательно связывается с одним из доступных транспортных протоколов, которому передаются входящие сообщения. В связке с транспортным протоколом поддерживается механизм объединения кусков раздробленных, при передаче, запросов.

Диалог конфигурации входящего SSL-транспорта изображён на рис.1.

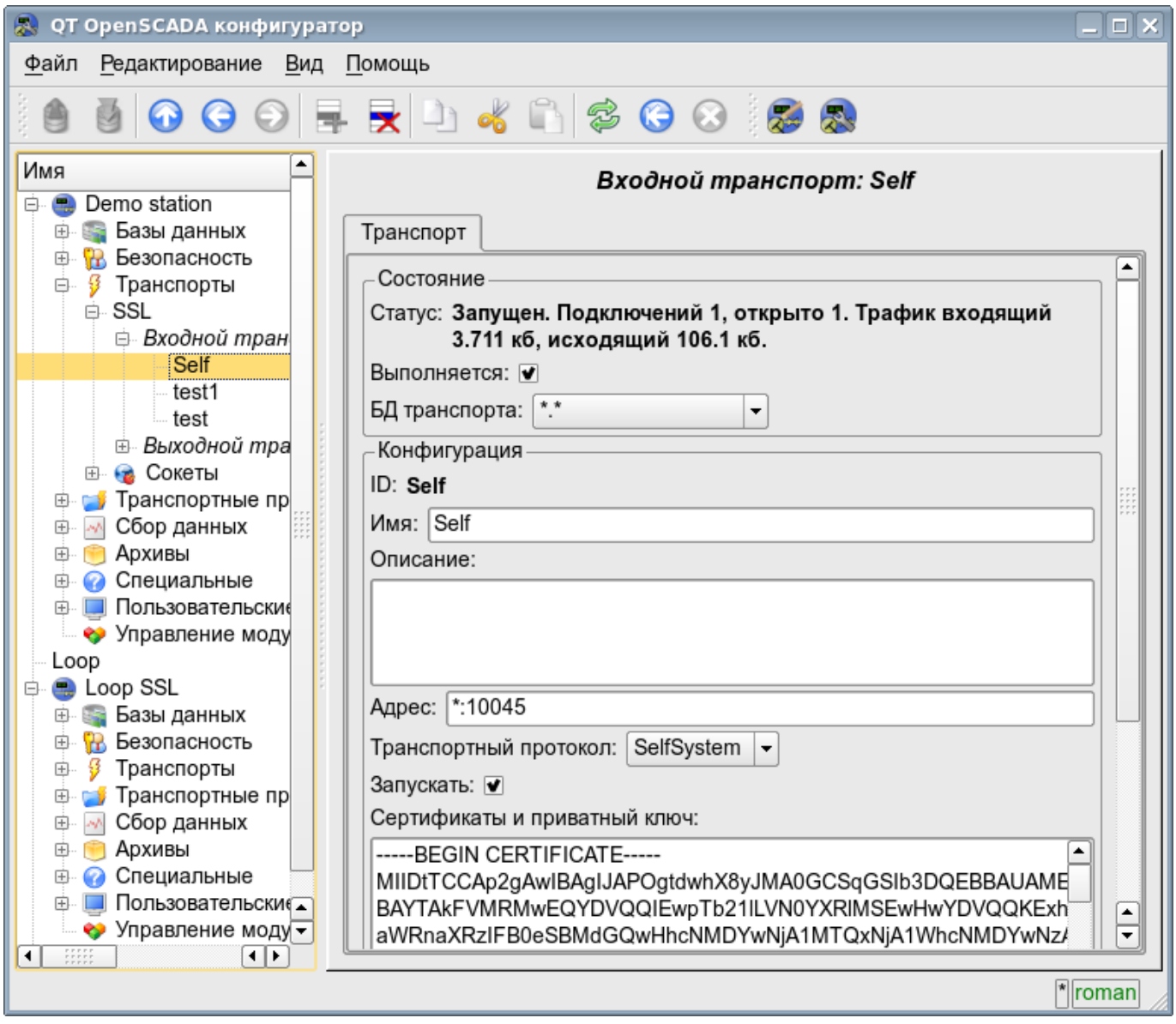

Рис.1. Диалог конфигурации входящего SSL-транспорта.

С помощью этого диалога можно установить:

- Состояние транспорта, а именно: «Статус», «Запущен» и имя БД содержащей конфигурацию.
- Идентификатор, имя и описание транспорта.
- Адрес транспорта в формате "[адрес]:[порт]:[режим]" где:
  - адрес – Адрес, на котором открывается SSL. Должен быть одним из адресов хоста. Если указано "*" то SSL будет доступен на всех интерфейсах хоста. Допускаются как символьное, так и IP представление адреса.
  - порт – Сетевой порт, на котором открывается SSL. Возможно указание

символьного имени порта (в соответствии с /etc/services).
- режим – SSL-режим и версия (SSLv2, SSLv3, SSLv23, TLSv1). По умолчанию и при ошибке используется SSLv23

- Выбор транспортного протокола.
- Состояние, в которое переводить транспорт при загрузке: «Запущен».
- Сертификаты, приватный ключ SSL и пароль приватного ключа SSL.
- Максимальное количество обслуживаемых клиентов и размер входного буфера.
- Ограничения режима "Keep-alive" по количеству запросов и времени ожидания.
- Приоритет задач транспорта.

2. Исходящие транспорты

Сконфигурированный и запущенный исходящий транспорт открывает SSL соединение с указанным сервером. При разрыве соединения, исходящий транспорт отключается. Для возобновления соединения транспорт нужно опять запустить.

Главная вкладка страницы конфигурации исходящего SSL-транспорта изображёна на рис.2.

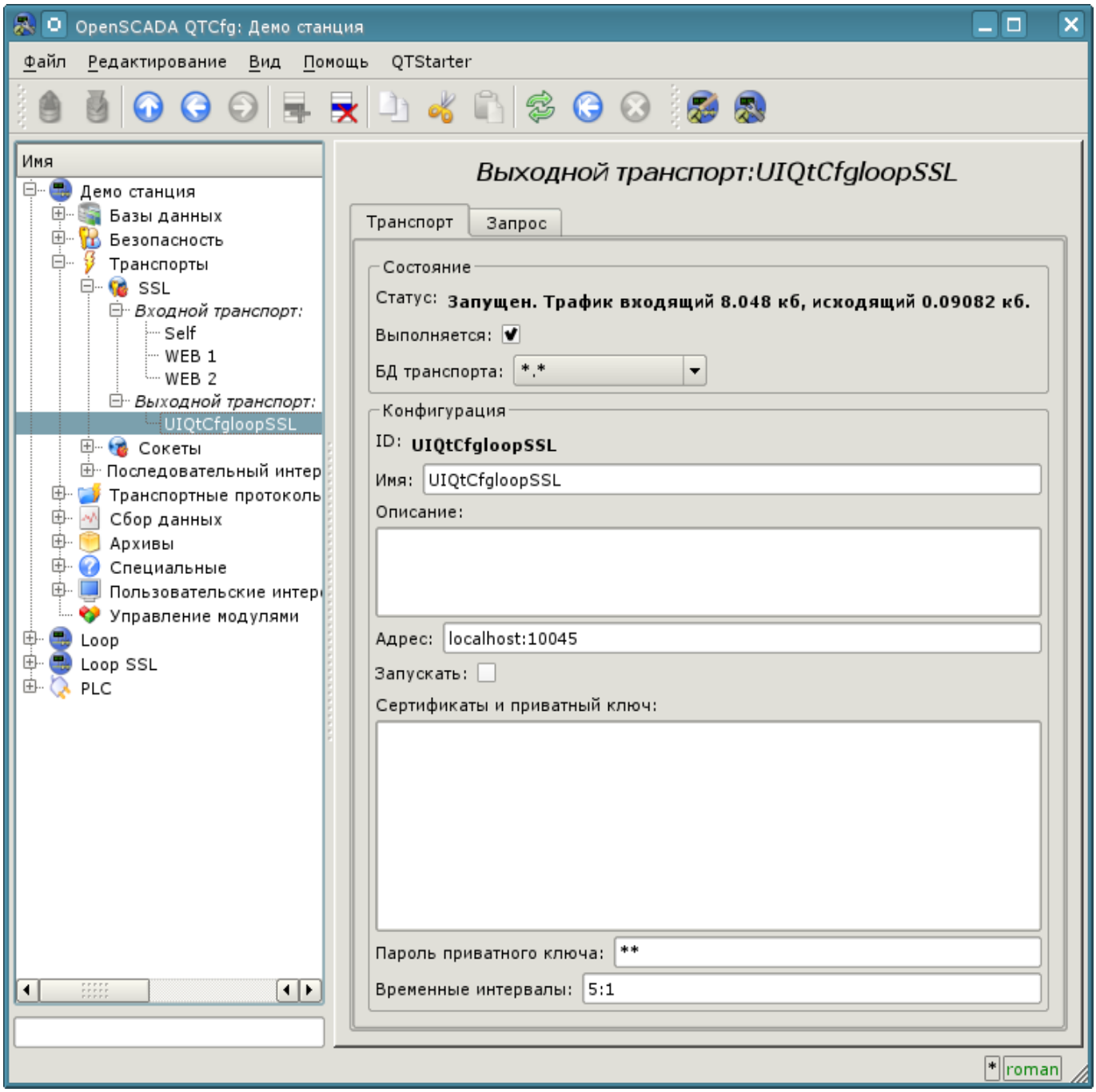

Рис.2. Главная вкладка страницы конфигурации исходящего SSL-транспорта.

С помощью этого диалога можно установить:

- Состояние транспорта, а именно: «Статус», «Запущен» и имя БД содержащей конфигурацию.
- Идентификатор, имя и описание транспорта.
- Адрес транспорта в формате "[адрес]:[порт]:[режим]" где:
  - адрес – Адрес, с которым выполняется соединение. Допускаются как символьное так и IP представление адреса.
  - порт – Сетевой порт, с которым выполняется соединение. Возможно указание символьного имени порта (в соответствии с /etc/services).
  - режим – SSL-режим и версия (SSLv2, SSLv3, SSLv23, TLSv1). По умолчанию и при ошибке используется SSLv23
- Состояние, в которое переводить транспорт при загрузке: «Запущен».
- Сертификаты, приватный ключ SSL и пароль приватного ключа SSL.
- Таймаут по умолчанию для ожидания соединения и ответа, отдельно.

3. Сертификаты и ключи

Для полноценной работы модуля необходимы сертификаты и приватные ключи. В случае с входящим SSL-транспортом (сервером) они обязательны. В случае с исходящим SSL-транспортом они могут и не устанавливаться хотя их использование желательно.

Простейшей конфигурацией сертификата является самоподписной сертификат и приватный ключ. Ниже описана процедура их создания с помощью утилиты openssl:

```
# Генерация секретного ключа
$ openssl genrsa -out ./key.pem -des3 -rand /var/log/messages 2048
# Генерация самоподписанного сертификата
$ openssl req -x509 -new -key ./key.pem -out ./selfcert.pem -days 365
```

Далее содержимое файлов selfcert.pem и key.pem копируется в текстовое поле сертификата и ключа. Пароль приватного ключа устанавливается в соответствующем поле.