# Автоматизированная система удаленного копирования данных

Yasir M. Arsanukaev

23 февраля 2010 г.

# Содержание

# 1 Предпосылка

Перед использованием приложения необходимо установить следующие продукты:

- Erlang/OTP[1]
- OpenSSL[2]

OpenSSL как правило входит в состав Unix-подобных OS, в т. ч. — в FreeBSD, GNU/Linux. В этом случае установка не требуется.

Для установки Erlang/OTP на FreeBSD выполните:

```
> portsnap fetch update
> cd /usr/ports/lang/erlang-lite/
> make install clean
```

# 2 Введение

Программное обеспечение предназначено для копирования файлов с компьютера (-ов) на удаленный сервер по защищенному каналу связи. ПО состоит из двух частей — серверной и клиентской, работающих на разных узлах сети. Для их настройки используются конфигурационные файлы (см. Конфигурация).

# 3 Безопасность

Обмен данными осуществляется по протоколу SSL, что снижает возможность прослушивания канала передачи данных злоумышленником и его вмешательства в процесс передачи данных до минимума. Снижает, но не исключает, поскольку в современном ПО могут содержаться ошибки, устраняемые с течением времени. Если тем не менее принять, что в зависимостях программы отсутствуют ошибки, то возможность атаки бесконечно мала. Атака "человек посередине" исключена, т. к. с клиентской компонентой распространяется сертефикат сервера, обеспечивающий клиенту в момент соединения возможность сверки передаваемого сервером сертификата с имеющимся. Если сертификаты идентичны, либо принадлежат к одному центру сертификации (Certificate Authority), то соединение считается безопасным. В противном случае клиентская часть разрывает соединение с сервером.

[1] Свежие версии доступны по адресу http://www.erlang.org/download.html

[2] Двоичные сборки для Windows: http://www.shininglightpro.com/products/Win32OpenSSL.html

### 3.1 Авторизация

После установки безопасного соединения сервер затребует пароль у клиента, и если клиент предоставляет пароль, идентичный тому, который имеется на стороне сервера, то передача файлов начинается, в противном случае сервер разрывает соединение.

### 3.2 Генерация ключа и сертификата

Генерация секретного ключа:

```
$ openssl genrsa -des3 -out key.pem 1024
```

При создании ключа будет запрошен пароль для его шифрования, который необходимо присвоить опции key_pass файла конфигурации сервера. Ключ необходимо поместить в директорию, указанную в опции certs_dir конфигурационного файла.

Создание самоподписного сертификата, действительного в течение пяти лет:

```
$ openssl req -new -x509 -days 1825 \
    -key key.pem -out cert.pem
```

Сертификат нужно поместить на стороне сервера и всех клиентских компьютеров в директории, указанные в опции certs_dir файлов конфигурации.

Операции по созданию секретного ключа и сертификата можно совместить:

```
$ openssl req \
    -new -x509 -newkey rsa:1024 \
    -subj "/CN=UK/O=Russel's Teapot foundation/OU=
       Secularization dept./CN=Flying.Spaghetti.Monster
       .com/emailAddress=Invisible@Pink.Unicorn.net" \
    -days 1825 -keyout key.pem -out cert.pem
```

Обратите внимание, что опция `-subj` позволяет указать детали сертификата не интерактивно.

## 4 Конфигурация

Компоненты программы перед их запуском получают настройки из файлов server.conf и client.conf. Опции конфигурации серверной и клиентской частей представлены ниже.

### 4.1 Опции сервера

| | |
|---|---|
| dest_dir | Директория, куда помещаются передаваемые файлы. |
| logfile | Файл для журналирования событий. |
| address | IP-адрес, на котором сервер принимает соединения. |
| port | Порт, по которому будут ожидаться соединения с клиентами. |
| key_pass | Пароль секретного ключа серверной части. |
| certs_dir | Директория, в которой размещаются секретный ключ и сертификат серверной части. |
| secret | Пароль, который должны предоставлять узлы для получения возможности передачи данных. |

### 4.2 Опции клиента

| | |
|---|---|
| source_dirs | Список директорий, откуда копируются файлы. |
| host | Адрес сервера. |
| port | Порт сервера. |
| certs_dir | Директория, в которой размещается цепочка сертификатов, либо корневой сертификат сервера. |
| secret | Пароль, который должны предоставлять узлы для получения возможности передачи данных. |

## 5 Переменные окружения

В случае Windows необходимо добавить в переменную окружения PATH путь к исполняемому файлу `erl`, например, "C:\Progra~1\erlX.Y.Z\bin", где X, Y, Z – номер версии.

## 6 Сборка

Для автоматизированной сборки необходимо наличие установленной утилиты GNU make[3]. В этом случае для компиляции достаточно перейти в директорию компоненты (server или client соответственно) и запустить команду `make`, либо `gmake`, если компиляция выполняется на FreeBSD. Скомпилированные модули должны появиться соответственно в директории ebin.

Для сборки в Windows можно просто запустить файл win-make.bat[4].

[3] Для Windows эта утилита входит в различные наборы инструментов, например, GnuWin32.

[4] Убедитесь, что правильно установлены переменные окружения (см. параграф 5).

# 7 Выполнение

Перед запуском програмы проверьте, что вы правильно установили переменные окружения (см. параграф 5). Для запуска обеих частей в Unix-подобных ОС используется следующая команда:

```
$ ./start.sh
```

Для запуска в Windows достаточно выполнить скрипт win-start.bat.

Серверная часть запускается в отдельной сессии, что позволяет продолжать работу в консоли после запуска. Тем не менее может быть полезно видеть, что происходит в Erlang-shell. Для этого запустите скрипт start-attached.sh, либо win-start-attached.bat для Windows.

Серверная часть журналирует все события в файл, указанный опцией `log_file`.