

Понадобилось мне как-то создать новые сертификаты для сервера и раскидать их по клиентским серверам. Делалось для того, чтобы иметь защищенное соединение у PostgreSQL.

Приступим.

Следующими действиями мы сгенерируем сертификаты для PostgreSQL

Предположим, у вас **уже установлена PostgreSQL(с ssl) и OpenSSL**.

Для начала **нужно** сгенерировать **3 сертификата** самого **сервера**.

- **root.crt** (доверенный корневой сертификат)
- **server.crt** (сертификат сервера)
- **server.key** (закрытый ключ)

Все команды мы будем выполнять от имени **root** (входить кто как привык sudo — или sudo -i)

Переходим в каталог с PostgreSQL

Для Gentoo:

```
| cd /var/lib/postgresql/data/
```

Для CentOS:

```
| cd /var/lib/pgsql/data
```

Далее, **создадим закрытый ключ** (server.key)

```
| openssl genrsa -des3 -out server.key 1024
```

Далее, **удалим пароль из ключа**, чтобы не вводить его каждый раз:

```
| openssl rsa -in server.key -out server.key
```

Установим необходимые права доступа и владельца файла

```
| chmod 400 server.key
```

```
| chown postgres.postgres server.key
```

Создадим сертификат сервера: (для параметра subj **обязательны одинарные кавычки**)

```
| openssl req -new -key server.key -days 3650 -out server.crt -x509 -  
subj '/C=RU/ST=Moscow/L=M/O=MyBlabla/CN=pg server'
```

Этой командой мы создадим сертификат сроком действия 10 лет, сгенерированный для России (C), штата/города Москва (ST), организации MyBlabla, и сервера pg server

Т.к. это самоподписанный сертификат, сделаем его доверенным

```
| cp server.crt root.crt
```

Далее отредактировать **postgresql.conf**, чтобы, активировать SSL:
ssl = on

После чего перезапускаем сервер

```
| /etc/init.d/postgresql restart
```

Сервер может сразу и не запуститься, если он не запустился — смотрите логи и

исправьте причину

На этом мы закончили генерацию серверного сертификата, но нам предстоит еще сгенерировать их для клиентов

Чтож, приступим.

Для клиента требуются следующие файлы:

root.crt (доверенный корневой сертификат)
postgresql.crt (сертификат сервера)
postgresql.key (закрытый ключ)

Мы будем генерировать их на сервере, например в каталог **/tmp/**

Сначала **создадим закрытый ключ**

```
| openssl genrsa -des3 -out /tmp/postgresql.key 1024
```

Удалим из него **пароль**

```
| openssl rsa -in /tmp/postgresql.key -out /tmp/postgresql.key
```

Создадим сертификат сервера (postgresql.crt)

```
| openssl req -new -key /tmp/postgresql.key -out /tmp/postgresql.csr -  
subj '/C=RU/ST=Moscow/L=M/O=MyBlabla/CN=vasya'
```

Здесь нужно в **парамetre CN** указать имя пользователя **postgre**, с именем которого мы подключаемся.

Создаем сам сертификат

```
openssl x509 -req -in /tmp/postgresql.csr -CA root.crt -CAkey  
server.key -days 3650 -out /tmp/postgresql.crt -CAcreateserial
```

Сертификат создается также на 10 лет (иначе на месяц)

С генерацией клиентских сертификатов завершили, теперь нужно скопировать их(3 файла указанные выше) на клиентские машины.

Для Windows:

```
%appdata%\postgresql\
```

Для Linux:

```
~/.postgresql/
```

P.S. обязательно проверьте (для Linux), чтобы у postgresql.key права доступа были 600

Похожие записи:

1. Работа MySQL с SSL
2. Установка SSL сертификата на nginx