# Аутентификация в PostgreSQL

Муравьёв С.К.

Национальный исследовательский ядерный университет «МИФИ»
Кафедра №36 «Информационные системы и технологии»

December 4, 2013

# Аутентификация

Аутентификация (англ. Authentication) – процедура проверки подлинности, например: проверка подлинности пользователя путём сравнения введённого им пароля с паролем в базе данных пользователей.

Аутентификацию не следует путать с авторизацией (процедурой предоставления субъекту определённых прав) и идентификацией (процедурой распознавания субъекта по его идентификатору).

# Методы аутентификации в PostgreSQL

- Trust
- Password
- MD5
- GSSAPI
- SSPI
- Kerberos
- Ident
- Peer
- LDAP
- RADIUS
- Certificate
- PAM

# Аутентификация Turst

PostgreSQL авторизует каждого, кто может подключиться к серверу, под любым именем (даже с именем суперпользователя).

Настройка pg_hba.conf:

```
host   all   all   127.0.0.1/32   trust
```

# Аутентификация Password

PostgreSQL производит аутентификацию по паролю, который передаётся на сервер в открытом виде. Данный метод не рекомендуется использовать в недоверенных сетях.

Настройка pg_hba.conf:

```
host   all   all   192.168.0.0/24   password
```

# Аутентификация MD5

PostgreSQL производит аутентификацию по паролю. Пароль передаётся на сервер в виде хэша MD5.

Настройка pg_hba.conf:

```
host   all   all   192.168.0.0/24   md5
```

# Аутентификация GSSAPI

Аутентификация в соответствии с протоколом GSSAPI, описанным в RFC 2743. Сам протокол аутентификации безопасен, но данные, передаваемые между сервером и клиентом, не шифруются. Для защиты данных необходимо использовать SSL. по паролю. Пароль передаётся на сервер в виде хэша MD5.

Настройка pg_hba.conf:

```
host   all   all   192.168.0.0/24   gss ...
```

# Аутентификация SSPI

Технология Windows для реализации безопасной аутентификации.

Настройка pg_hba.conf:

```
host   all   all   192.168.0.0/24   sspi ...
```

# Аутентификация Kerberos

Kerberos - промышленный стандарт для безопасной аутентификации в распределённых системах. Данный протокол считается устаревшим. Вместо него рекомендуется использовать GSSAPI.

Настройка pg_hba.conf:

```
host   all   all   192.168.0.0/24   krb5 ...
```

# Аутентификация Ident

PostgreSQL запрашивает у сервера идентификации имя пользователя на клиентской машине и сравнивает его с заявленным именем пользователя. Может использоваться только для TCP/IP-соединений.

Протокол описан в RFC 1413.

Настройка pg_hba.conf:

```
host   all   all   192.168.0.0/24   ident
```

# Аутентификация Peer

PostgreSQL запрашивает имя пользователя через ядро операционной системы. Может использоваться только для локальных соединений.

Протокол описан в RFC 1413.

Настройка pg_hba.conf:

```
host   all   all   192.168.0.0/24   peer
```

# Аутентификация LDAP

Данный метод похож на метод password но для проверки пар имя/пароль используется LDAP сервер. При этом соответствующий пользователь также должен существовать в БД.

Настройка pg_hba.conf:

```
host   all   all   192.168.0.0/24   ldap ...
```

# Аутентификация RADIUS

Данный метод похож на метод password но для проверки пар имя/пароль используется RADIUS сервер. При этом соответствующий пользователь также должен существовать в БД.

Настройка pg_hba.conf:

```
host   all   all   192.168.0.0/24   radius ...
```

# Аутентификация Certificate

При использовании данного метода аутентификации сервер требует от клиента действующий SSL-сертификат. Атрибут сертификата cn (Common Name) сравнивается с запрашиваемым именем пользователя базы данных, и если они совпадают, то подключение разрешается.

Метод работает только для SSL-подключений.

Можно настроить правила сопоставления, если значение атрибута cn и имя пользователя должны отличаться.

# Инфраструктура открытых ключей

Инфраструктура открытых ключей (англ. PKI - Public Key Infrastructure) - набор средств, распределенных служб и компонентов, в совокупности используемых для поддержки криптозадач на основе закрытого и открытого ключей.

В основе PKI лежит использование криптографической системы с открытым ключом и несколько основных принципов:

- закрытый ключ известен только его владельцу;
- удостоверяющий центр создает сертификат открытого ключа, таким образом удостоверяя этот ключ;
- никто не доверяет друг другу, но все доверяют удостоверяющему центру;
- удостоверяющий центр подтверждает или опровергает принадлежность открытого ключа заданному лицу, которое владеет соответствующим закрытым ключом.

# Удостоверяющий центр

Создание удостоверяющего центра:

```
$ /etc/pki/tls/misc/CA -newca
Enter PEM pass phrase: 1234
Verifying - Enter PEM pass phrase: 1234
...
Country Name:ru
State or Province Name:msk
Locality Name (eg, city):msk
Organization Name (eg, company):mephi
Organizational Unit Name:kaf36
Common Name:root
Email Address:root@kaf36.
```

# Создание ключа и сертификата для сервера СУБД

Создание закрытого ключа сервера:

```
# su - postgres
$ cd /var/lib/pgsql/9.3/data/
$ openssl genrsa -out server.key 2048
$ chmod 0600 server.key
```

Создание заявки на подпись сертификата (certificate signing request):

```
$ openssl req -new -text -key server.key -out server.csr
```

Удостоверяющий центр по заявке выпускает сертификат:

```
$ su
# openssl ca -out server.crt -in server.scr
# rm server.scr
```

# Настройка сервера СУБД

Установка сертификата УЦ:

```
# cd /var/lib/pgsql/9.3/data/
# cp /etc/pki/CA/cacert.pem ./root.crt
```

Настройка postgresql.conf:

```
ssl = on
ssl_cert_file = 'server.crt'
ssl_key_file = 'server.key'
ssl_ca_file = 'root.crt'
```

Настройка pg_hba:.conf:

```
hostssl all all 192.168.0.0/24 cert clientcert=1
```

Перезапуск сервера:

```
# service postgresql-9.3 restart
```

# Создание ключа и сертификата клиента

Создание закрытого ключа клиента:

```
# su - student
$ mkdir ~/.postgresql
$ cd ~/.postgresql
$ openssl genrsa -out postgresql.key 2048
$ chmod 0600 postgresql.key
```

Создание заявки на подпись сертификата (certificate signing request):

```
$ openssl req -new -text -key postgresql.key
          -out postgresql.csr
```

Удостоверяющий центр по заявке выпускает сертификат:

```
$ su
# openssl ca -out postgresql.crt -in postgresql.scr
# rm postgresql.scr
```

# Настройка клиента

Установка сертификата УЦ:

```
# cd /home/student/.postgresql
# cp /etc/pki/CA/cacert.pem ./root.crt
```

Подключение к серверу:

```
# su - student
$ psql -h 192.168.0.1 mephi
```

# Проверка сертификатов

Проверка сертификата:

```
# /etc/pki/tls/misc/CA -verify server.crt
server.crt: OK
```

# Аутентификация PAM

Данный метод похож на метод password но для проверки пар имя/пароль используется механизм подключаемых модулей аутентификации (Pluggable Authentication Modules) . При этом соответствующий пользователь также должен существовать в БД.

Настройка pg_hba.conf:

```
host   all   all   192.168.0.0/24   pam ...
```

# Практика

- Создать нового пользователя в базе данных и операционной системе.
- Для созданного пользователя:
  - Настроить и проверить метод аутентификации trust.
  - Настроить и проверить метод аутентификации peer.
  - Настроить и проверить метод аутентификации MD5.
  - Создать ключ и сертификат.
  - Настроить и проверить метод аутентификации cert.