Авторы: Алексей Береснев, Илья Баштанов, Павел Толмачев

Обратная связь
Отзывы, замечания и предложения направляйте по адресу:
edu@postgrespro.ru

Темы

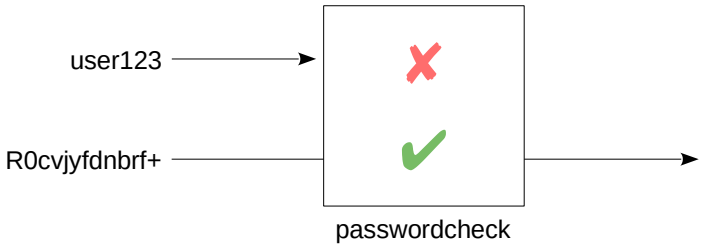
Запрет простых паролей

Профили пользователей

Запрет простых паролей

Модуль passwordcheck

Модуль passwordcheck



Проверка паролей пользователей на простоту

CREATE ROLE
ALTER ROLE
\password

Модуль passwordcheck проверяет пароли пользователей, задаваемые командами CREATE ROLE и ALTER ROLE, а также устанавливаемые встроенной командой \password в psql.

Модуль должен быть загружен с помощью shared_preload_libraries или LOAD.

Модуль passwordcheck выполняет следующие проверки паролей на простоту:

- пароль должен содержать не менее 8 символов;
- пароль не должен содержать имя роли;
- пароль должен содержать и буквы, и иные символы.

Модуль не предусматривает конфигурационных настроек.

Модуль passwordcheck

```
student$ psql
```

Расширение passwordcheck требует загрузки одноименной разделяемой библиотеки:

```
=> ALTER SYSTEM SET shared_preload_libraries = 'passwordcheck';
```

```
ALTER SYSTEM
```

После подключения библиотеки необходимо перезагрузить СУБД.

```
student$ sudo systemctl restart postgrespro-ent-13.service
```

```
student$ psql
```

Создадим базу данных и роль:

```
=> CREATE DATABASE profile;
```

```
CREATE DATABASE
```

```
=> CREATE ROLE bob LOGIN;
```

```
CREATE ROLE
```

Проверим, можно ли установить пользователю нестойкий пароль bob123:

```
=> ALTER ROLE bob PASSWORD 'bob123';
```

```
ERROR:  password is too short
```

Теперь попробуем установить сложный пароль:

```
=> ALTER ROLE bob PASSWORD 'GGU2015ujlf';
```

```
ALTER ROLE
```

Удалим модуль passwordcheck и перезагрузим СУБД.

```
=> ALTER SYSTEM RESET shared_preload_libraries;
```

```
ALTER SYSTEM
```

```
student$ sudo systemctl restart postgrespro-ent-13.service
```

Профили пользователей

Профиль

Попытки входа

Время жизни пароля

Уровень сложности пароля

Профиль

Профиль ограничивает использование системы пользователем

- количество неудачных попыток входа
- время жизни пароля и время неактивности
- уровень сложности пароля

По умолчанию к ролям применяется профиль default

- не устанавливает никаких ограничений, но это можно изменить

Ядро Postgres Pro обеспечивает гораздо более гибкое управление парольными политиками, чем простое расширение passwordcheck.

Профиль ограничивает использование базы данных и устанавливает парольную политику для ролей. Профили определяются на уровне кластера баз данных и назначаются ролям. Создает новый профиль команда CREATE PROFILE, изменяет — ALTER PROFILE, удаляет профиль — DROP PROFILE.

Профиль назначается при создании роли или командой ALTER ROLE. Новым ролям по умолчанию назначается профиль default. Исходно профиль default не устанавливает ограничений, их можно добавить командой ALTER PROFILE. Если роли назначен профиль, отличный от default, модуль passwordcheck, рассмотренный ранее, не работает.

Попытки входа

FAILED_LOGIN_ATTEMPTS

количество неудачных попыток входа до блокировки пользователя
ALTER ROLE … ACCOUNT UNLOCK

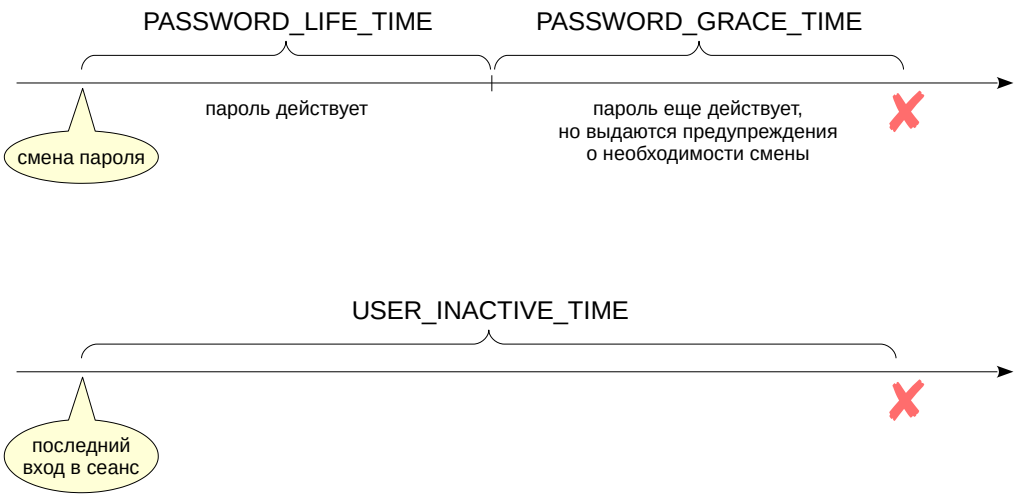
FAILED_AUTH_KEEP_TIME

время в днях до сброса счетчика неудачных попыток

Параметр профиля FAILED_LOGIN_ATTEMPTS ограничивает количество неудачных попыток входа в сеанс. По превышении этого количества пользователь блокируется. Суперпользователь может разблокировать заблокированную роль командой ALTER ROLE с предложением ACCOUNT UNLOCK.

Параметр профиля FAILED_AUTH_KEEP_TIME задает время в днях с момента первой неудачной попытки входа; при попытке входа по истечении этого времени счетчик неудачных попыток сбрасывается, а пользователь разблокируется.

Время жизни пароля



Параметр профиля PASSWORD_LIFE_TIME задает период в днях, в течение которого пароль действителен.

Если установлен PASSWORD_GRACE_TIME, действие пароля продлевается, но в течение этого периода выдаются предупреждение о необходимости смены пароля. Если пароль так и не был изменен, последующие попытки входа в сеанс будут отвергнуты.

Параметр профиля USER_INACTIVE_TIME задает максимальный период в днях с момента последнего входа пользователя, в течение которого пользователь должен снова войти в сеанс. В противном случае пользователь блокируется.

Уровень сложности пароля

PASSWORD_MIN_UNIQUE_CHARS

минимальное количество уникальных символов

PASSWORD_MIN_LEN

минимальная длина

PASSWORD_REQUIRE_COMPLEX

проверка сложности пароля

PASSWORD_REUSE_TIME и PASSWORD_REUSE_MAX

время в днях и количество смен паролей, после которых старый пароль можно использовать повторно

Параметры профиля PASSWORD_MIN_UNIQUE_CHARS и PASSWORD_MIN_LEN ограничивают минимальное количество уникальных символов в пароле и его минимальную длину.

Параметр PASSWORD_REQUIRE_COMPLEX требует, чтобы пароль содержал и буквы, и небуквенные символы, а также не содержал имя пользователя.

Параметры профиля PASSWORD_REUSE_TIME и PASSWORD_REUSE_MAX определяют, когда пароль может быть использован повторно: для этого должно пройти указанное время и пароль должен быть сменен указанное количество раз.

Профили пользователей

```
student$ psql -d profile
```

Пока ничего не мешает пользователю установить простой пароль:

```
=> ALTER ROLE bob PASSWORD 'bob123';
```

```
ALTER ROLE
```

Создадим профиль:

```
=> CREATE PROFILE IF NOT EXISTS prof LIMIT
  PASSWORD_LIFE_TIME 60
  PASSWORD_GRACE_TIME 7
  PASSWORD_MIN_LEN 8
  PASSWORD_REQUIRE_COMPLEX;
```

```
CREATE PROFILE
```

Имя профиля prof, пароль необходимо менять через 60 дней. Еще в течение 7 дней будут выводиться предупреждения о необходимости поменять пароль.

Минимальная длина пароля — 8 символов, причем пароль должен содержать как буквы, так и другие символы, а имя пользователя не должно входить в пароль.

Назначим роли bob профиль prof:

```
=> ALTER ROLE bob PROFILE prof;
```

```
ALTER ROLE
```

Как убедиться, что с ролью связан профиль, не являющийся профилем по умолчанию?

```
=> SELECT r.rolname, p.pflname
FROM pg_authid r
  JOIN pg_profile p ON p.oid = r.rolprofile
WHERE r.rolname = 'bob';
```

```
 rolname | pflname
---------+---------
 bob     | prof
(1 row)
```

Попробуем установить короткий пароль:

```
=> ALTER ROLE bob PASSWORD 'bob123';
```

```
ERROR:  password must be at least 8 characters long
```

Попробуем сделать пароль длиннее:

```
=> ALTER ROLE bob PASSWORD 'bob12345';
```

```
ERROR:  password must not contain user name
```

Усложним:

```
=> ALTER ROLE bob PASSWORD 'GGU2015ujlf';
```

```
ALTER ROLE
```

Модифицируем профиль prof, установив ограничение в три неудачные попытки начать сеанс до блокировки учетной записи.

```
=> ALTER PROFILE prof LIMIT FAILED_LOGIN_ATTEMPTS 3;
```

```
ALTER PROFILE
```

Боб пытается подключиться три раза с неправильными паролями:

```
student$ psql 'host=localhost dbname=profile user=bob password=12345678'
```

```
psql: error: FATAL:  password authentication failed for user "bob"
```

```
student$ psql 'host=localhost dbname=profile user=bob password=qwerty'
```

```
psql: error: FATAL:  password authentication failed for user "bob"
```

```
student$ psql 'host=localhost dbname=profile user=bob password=q1w2e3r4'
```

```
psql: error: FATAL:  password authentication failed for user "bob"
```

Теперь роль заблокирована:

```
=> SELECT CASE rolstatus
  WHEN 0 THEN 'роль активна'
  WHEN 1 THEN 'заблокирована вручную'
  WHEN 2 THEN 'заблокирована из-за бездействия'
  WHEN 4 THEN 'заблокирована по превышению числа попыток входа'
END status
FROM pg_roles
WHERE rolname = 'bob';

                     status
-------------------------------------------------
 заблокирована по превышению числа попыток входа
(1 row)
```

Разблокируем роль, дадим Бобу шанс вспомнить пароль:

```
=> ALTER ROLE bob ACCOUNT UNLOCK;

ALTER ROLE

student$ psql 'host=localhost dbname=profile user=bob password=GGU2015ujlf' -c '\conninfo'

You are connected to database "profile" as user "bob" on host "localhost" (address "127.0.0.1") at port "5432".
```

Следующим образом можно установить профиль по умолчанию:

```
=> ALTER ROLE bob PROFILE default;

ALTER ROLE
```

Удалим профиль:

```
=> DROP PROFILE prof;

DROP PROFILE
```

Удалим роль:

```
=> DROP ROLE bob;

DROP ROLE
```

Итоги

Расширение passwordcheck проверяет сложность паролей

Встроенный механизм профилей позволяет настраивать политики использования системы пользователями

Профиль позволяет задавать уровень сложности пароля и ограничивать время его жизни

Можно ограничивать количество неудачных попыток ввода пароля

Практика

1. Создайте профиль с ограничением на минимальное количество разных символов в пароле и проверьте его.
2. Установите ограничение в две неудачные попытки ввода пароля для пользователя и проверьте его работу.

1. Создайте профиль mgr1 с ограничением PASSWORD_MIN_UNIQUE_CHARS = 8. Зарегистрируйте пользователя mgr1 с профилем mgr и проверьте возможность установки пароля не менее, чем с восемью отличающимися символами.

2. Используйте настройку LIMIT FAILED_LOGIN_ATTEMPTS по аналогии с демонстрацией.

1. Профиль пользователя

Создадим базу данных profile:

```
student$ /opt/pgpro/ent-13/bin/createdb profile
```

```
student$ psql -d profile
```

Создадим профиль с ограничением на минимальное количество отличающихся символов в пароле = 8.

```
=> CREATE PROFILE mgr LIMIT PASSWORD_MIN_UNIQUE_CHARS 8;
```

```
CREATE PROFILE
```

Зарегистрируем пользователя с этим профилем.

```
=> CREATE ROLE mgr1 LOGIN PROFILE mgr;
```

```
CREATE ROLE
```

Проверим профиль роли.

```
=> SELECT r.rolname, p.pflname, p.pflpasswordminuniqchars
FROM pg_roles r
JOIN pg_profile p
ON r.rolprofile = p.oid
WHERE r.rolname = 'mgr1';
```

```
 rolname | pflname | pflpasswordminuniqchars
---------+---------+-------------------------
 mgr1    | mgr     |                       8
(1 row)
```

Пароль, разрешенный профилем.

```
=> ALTER ROLE mgr1 PASSWORD '12345678';
```

```
ALTER ROLE
```

Такой пароль был успешно назначен роли.

Пароль с меньшим количеством отличающихся символов.

```
=> ALTER ROLE mgr1 PASSWORD '12345677';
```

```
ERROR:  password must contain at least 8 unique characters
```

2. Ограничение неудачных попыток ввода пароля

Разрешим лишь две попытки.

```
=> ALTER PROFILE mgr LIMIT FAILED_LOGIN_ATTEMPTS 2;
```

```
ALTER PROFILE
```

Пару раз введем неверный пароль.

```
student$ psql 'host=localhost dbname=profile user=mgr1 password=manager'
```

```
psql: error: FATAL:  password authentication failed for user "mgr1"
```

```
student$ psql 'host=localhost dbname=profile user=mgr1 password=manager'
```

```
psql: error: FATAL:  password authentication failed for user "mgr1"
```

Блокировка должна быть уже установлена.

```
=> SELECT CASE rolstatus
  WHEN 0 THEN 'роль активна'
  WHEN 1 THEN 'заблокирована вручную'
  WHEN 2 THEN 'заблокирована из-за бездействия'
  WHEN 4 THEN 'заблокирована по превышению числа попыток входа'
END status
FROM pg_roles
WHERE rolname = 'mgr1';
```

```
                     status
-------------------------------------------------
 заблокирована по превышению числа попыток входа
(1 row)
```

Удалим роль:

```
=> DROP ROLE mgr1;
```

```
DROP ROLE
```

Удалим профиль:

```
=> DROP PROFILE mgr;
```

```
DROP PROFILE
```