Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

SECURITY DEFINER в PostgreSQL: почему одной схемы мало

Как мы закрываем подмену объектов через pg_temp, REVOKE и аудит pg_proc

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

Заказчик исключает из search_path схему public и считает SECURITY DEFINER-функцию защищённой. Но pg_temp, если не перечислить его явно, ищется первым — даже раньше pg_catalog — и писать в него может любой подключённый пользователь. Он создаёт временную таблицу с коротким именем реального объекта, и код владельца молча работает с подставной таблицей в правах владельца функции.

Почему это важно бизнесу

- Функция начисления пишет в подставную таблицу — расхождение всплывает только на сверке за период, когда сессия уже закрыта
- Разбор занимает недели: временная схема удаляется вместе с сессией, в pg_stat_activity и логах приложения аномалий нет
- ИБ-аудит требует доказательства: строка search_path в pg_proc.proconfig, а не устное «схему public мы исключили»
- Патч на 10-20 боевых функций после инцидента — это окно простоя и ручная сверка каждой функции с владельцем
- Функция с доступом к персональным данным исполняется правами владельца — подмена объекта даёт чтение всей таблицы

ЧТО МЫ ЗАКЛАДЫВАЕМ

Ключевые параметры реализации

18

актуальная мажорная ветка PostgreSQL, по докам которой мы держим стандарт search_path

postgresql.org, CREATE FUNCTION 18

15

версия, с которой CREATE на public отозван у PUBLIC — но только для новых кластеров и БД

postgresql.org, Release Notes 15

pg_temp

схема, которая ищется первой, даже раньше pg_catalog, если не указана в search_path явно

postgresql.org, search_path

2

минимум записей в search_path защищённой функции: доверенная схема и pg_temp последним

postgresql.org, CREATE FUNCTION 18

100%

доля SECURITY DEFINER-функций, которые мы прогоняем через обязательный SET search_path

стандарт ITfresh

0

целевое число функций с EXECUTE у PUBLIC без явного обоснования после аудита

стандарт ITfresh

РЕАЛИЗАЦИЯ 1 ИЗ 3

Пересборка SECURITY DEFINER-функций в контуре биллинга

Что настраиваем

Продовый кластер PostgreSQL 15-18, схема app с 30+ функциями начисления и сверки от роли-владельца billing_owner

Как мы это делаем

1. Снимаем срез: SELECT proname, prosecdef, proconfig FROM pg_proc WHERE prosecdef — видно, у каких функций SET search_path вообще нет
2. Для каждой такой функции: ALTER FUNCTION app.f(...) SET search_path = app, pg_temp; pg_catalog не указываем — он ищется всегда
3. pg_temp ставим строго последним: если его не перечислить, временная схема остаётся первой в поиске и исключение public ничего не даёт
4. Ссылки на таблицы и типы в теле квалифицируем схемой (app.balances, не balances), динамический SQL собираем через format с %I и %L
5. Регресс-тест: под непривилегированной ролью делаем CREATE TEMP TABLE balances и проверяем, что функция всё равно читает app.balances

РЕЗУЛЬТАТ

Функция перестаёт резолвить короткое имя во временную схему: pg_temp остаётся в пути только для собственных временных таблиц функции и никогда не подставляется вместо app.*. Проверка воспроизводима — тест с temp-таблицей падает, если кто-то снимет SET-клаузу.

КЛЮЧЕВОЙ НЮАНС

Смотрим не на факт наличия SET search_path, а на порядок: public исключён, pg_temp стоит последним. Чаще всего в «уже защищённых» функциях pg_temp просто не упомянут — и защита не работает.

РЕАЛИЗАЦИЯ 2 ИЗ 3

Отзыв CREATE на public и EXECUTE по умолчанию

Что настраиваем

Кластеры, поднятые до PostgreSQL 15 и переехавшие через pg_upgrade или restore дампа: дефолт 15-й ветки на них не срабатывает

Как мы это делаем

1. Проверяем права схемы командой \dn+ public — на апгрейженных кластерах у PUBLIC остаётся CREATE даже под PostgreSQL 18
2. Выполняем REVOKE ALL ON SCHEMA public FROM PUBLIC, затем возвращаем USAGE ролям приложения, если объекты в public реально используются
3. ALTER DEFAULT PRIVILEGES FOR ROLE app_owner REVOKE EXECUTE ON FUNCTIONS FROM PUBLIC — иначе EXECUTE на новые функции выдаётся всем
4. Новую функцию создаём и урезаем одной транзакцией: BEGIN; CREATE FUNCTION ...; REVOKE ALL ON FUNCTION ... FROM PUBLIC; GRANT EXECUTE ...; COMMIT
5. В шаблон миграции (Flyway/Liquibase) зашит блок SET search_path прямо в CREATE FUNCTION — без него скрипт не проходит ревью

РЕЗУЛЬТАТ

Пользователь приложения больше не создаёт объекты в public, а новая функция недоступна на EXECUTE до явного GRANT. Окна, в котором функция уже создана, а привилегии PUBLIC ещё не отозваны, не возникает — обе команды идут в одной транзакции.

КЛЮЧЕВОЙ НЮАНС

Дефолт PostgreSQL 15 действует только для новых кластеров и новых БД: pg_upgrade и restore дампа сохраняют прежние права и владельца public, поэтому на каждой мигрированной базе REVOKE выполняется руками.

РЕАЛИЗАЦИЯ 3 ИЗ 3

Аудит pg_proc и event trigger в пайплайне релиза

Что настраиваем

Продовые и стейджинг-кластеры на регламентном обслуживании ITfresh: проверка встроена в цикл релиза, отдельного окна на неё нет

Как мы это делаем

1. Раз в релиз прогоняем запрос к pg_proc: отбираем prosecdef, у которых proconfig IS NULL либо в нём нет элемента вида search_path=%
2. Джойн с pg_namespace даёт схему и владельца (proowner) — сразу видно, чьими правами исполняется каждая найденная функция
3. Любая SECURITY DEFINER-функция без search_path — блокер релиза: пайплайн миграций падает, деплой не проезжает до исправления
4. На стейджинге event trigger на ddl_command_end (CREATE FUNCTION, ALTER FUNCTION) пишет в журнал аудита факт создания такой функции
5. По каждой функции фиксируем ответ, нужен ли ей SECURITY DEFINER — часто хватает обычной функции и GRANT на конкретную таблицу

РЕЗУЛЬТАТ

Функция с повышением привилегий не попадает в прод без закреплённого search_path: проверка живёт в пайплайне и не зависит от памяти инженера. Регрессия ловится на ближайшем релизе, а не на разборе инцидента через квартал.

КЛЮЧЕВОЙ НЮАНС

Источник правды — pg_proc.proconfig и prosecdef, а не вывод \df+ и не парсинг DDL: запрос к каталогу видит функции независимо от того, кто и каким скриптом их создавал.

ЧТО МЫ ВЫУЧИЛИ НА ПРАКТИКЕ

Подводные камни

- ✕ **Исключили public, но не добавили pg_temp**
  Если pg_temp не перечислен в search_path явно, он ищется первым — даже раньше pg_catalog. Указывать его надо последним элементом списка.

- ✕ **search_path задан на роли, а не в функции**
  ALTER ROLE ... SET search_path действует на сессии этой роли; вызов из другой роли идёт со своим путём. Параметр должен лежать в proconfig функции.

- ✕ **Имена в теле функции не квалифицированы**
  Короткое имя резолвится по search_path, а в динамическом SQL ещё и склеивается с вводом. Пишем app.balances и собираем EXECUTE через format %I/%L.

- ✕ **EXECUTE остаётся у PUBLIC по умолчанию**
  На новую функцию EXECUTE выдаётся PUBLIC автоматически. Без REVOKE ALL ON FUNCTION ... FROM PUBLIC вызвать её может любая подключённая роль.

- ✕ **Дефолт public из PG 15 не доехал апгрейдом**
  Отзыв CREATE у PUBLIC применяется к новым кластерам и новым БД. После pg_upgrade или restore права переезжают как есть — REVOKE делаем руками.

- ✕ **Проверка search_path вручную, не по pg_proc**
  Ревью по \df+ пропускает функции, созданные скриптом миграции. Запрос к pg_proc.proconfig в CI ловит их независимо от способа создания.

- ✕ **SECURITY DEFINER там, где хватало GRANT**
  Повышение прав берут по инерции. Каждая такая функция — отдельная точка контроля search_path, поэтому их число мы сокращаем, а не только защищаем.

- ✕ **Доверенная схема без REVOKE CREATE на неё**
  SET search_path = app бесполезен, если создавать объекты в app может широкий круг ролей. Схема доверенная только после REVOKE CREATE ON SCHEMA app.

УРОВНИ РЕАЛИЗАЦИИ

Как правильно

МИНИМУМ

- SET search_path с pg_temp последним у каждой SECURITY DEFINER-функции
- REVOKE CREATE ON SCHEMA public FROM PUBLIC на всех апгрейженных кластерах
- Квалификация схемой всех объектов в теле функции, включая динамический SQL

НОРМАЛЬНО

- ALTER DEFAULT PRIVILEGES ... REVOKE EXECUTE ON FUNCTIONS FROM PUBLIC
- CREATE FUNCTION и REVOKE/GRANT — одной транзакцией, без окна доступности
- Блок SET search_path зашит в шаблон миграции Flyway/Liquibase

ХОРОШО

- Запрос к pg_proc.proconfig в CI как блокер релиза, а не ручной чек-лист
- Event trigger на ddl_command_end логирует SECURITY DEFINER без search_path
- Ежеквартальный пересмотр: нужен ли функции прыжок привилегий вообще
- Регресс-тест с CREATE TEMP TABLE под непривилегированной ролью

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] У каждой SECURITY DEFINER-функции SET search_path виден в pg_proc.proconfig?
- [ ] pg_temp указан последним элементом списка, а не пропущен вовсе?
- [ ] REVOKE CREATE ON SCHEMA public FROM PUBLIC выполнен на базах, мигрированных с версий до 15?
- [ ] EXECUTE на новые функции не достаётся PUBLIC — ALTER DEFAULT PRIVILEGES настроен?
- [ ] Объекты в теле квалифицированы схемой, динамический SQL собран через format %I/%L?
- [ ] Доверенная схема закрыта: REVOKE CREATE ON SCHEMA app FROM PUBLIC выполнен?
- [ ] Проверка proconfig встроена в CI-пайплайн миграций как блокирующий шаг?
- [ ] Есть регресс-тест с временной таблицей-двойником под непривилегированной ролью?

Если хотя бы на два вопроса ответ «нет» или «не знаю» — тема требует внимания.

ЧТО МЫ ДЕЛАЕМ

Как поможет ITFresh

ITFresh — ИТ-аутсорсинг для юридических лиц до 50 рабочих мест в Москве и области. 15+ лет практики, собственная инфраструктура в дата-центре МТС (8 серверов Dell Xeon Platinum).

- Аудит SECURITY DEFINER-функций по pg_proc с отчётом по каждой незащищённой функции
- Пересборка search_path и квалификации имён в боевых функциях без остановки БД
- Настройка REVOKE CREATE/EXECUTE и ALTER DEFAULT PRIVILEGES на схемах и ролях
- Встраивание проверки proconfig в CI-пайплайн миграций как блокирующего шага
- Регламентный квартальный аудит новых функций и избыточных привилегий

15+
лет в ИТ-поддержке

50
рабочих мест — наш профиль

МТС
дата-центр, Москва

КОНТАКТЫ

Обсудить вашу задачу

Сайт	**itfresh.ru**
Телефон	**+7 903 729-62-41**
Telegram	**@ITfresh_Boss**

Бесплатно посмотрим вашу инфраструктуру по этому чек-листу и скажем, где тонко — без обязательств.

itfresh.ru

Техническая база

01 CREATE FUNCTION — Writing SECURITY DEFINER Functions Safely (postgresql.org — 18)

02 Client Connection Defaults — search_path (postgresql.org — 18)

03 Schemas — Usage Patterns (ddl-schemas) (postgresql.org — 18)

04 Release Notes 15 — public schema privileges (postgresql.org — 15)

05 System Catalogs — pg_proc (proconfig, prosecdef) (postgresql.org — 18)

06 ALTER DEFAULT PRIVILEGES (postgresql.org — 18)

07 REVOKE (postgresql.org — 18)

08 Privileges (ddl-priv) (postgresql.org — 18)

09 Шаблон ревью миграций PostgreSQL (стандарт ITfresh — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.