



Разработки Postgres Pro для повышения безопасности и защиты данных

Вебинар

4 июля 2024 г., г. Москва

Гусаков Андрей,

Старший технический консультант

Реальны ли угрозы?

- СУБД – «лакомый кусочек» для злоумышленников и «последний рубеж обороны» для сотрудников отдела ИБ (см. tadviser.ru/index.php/Статья:Утечки_данных_в_России)
 - «В 2023 году из финансовых организаций в России утекло 170,3 млн записей персональных данных клиентов, что в 3,2 раза превосходит показатель предыдущего года»
 - «Общий объем утекших данных оценивается в 103,4 терабайта. Попавшие в общий доступ базы данных совокупно содержат 4,8 млн строк, в том числе 225 млн номеров телефонов и 145 млн адресов электронной почты»
- Риски несанкционированного доступа, потери целостности данных и самих данных – следствия ненадлежащего управления разрешениями, атак инъекционного типа, эксплуатации уязвимостей БД, наличия теневой инфраструктуры БД, уязвимости данных резервного копирования
- В апреле 2023 г. ФСТЭК ужесточила требования к разработчикам средств защиты СУБД
- Запрос на bdu.fstec.ru/search/index?q=postgres возвращает около 40 выявленных (и устраненных) уязвимостей на стыке ОС-СУБД -> применение рекомендаций и обновлений
- Применение «накладных» средств защиты не отменяет необходимость защиты на уровне ядра СУБД
- Для построения эффективной системы защиты необходимо сотрудничество между отделами ИБ, сопровождения информационных систем, вендорами и интеграторами

Методы и стратегии защиты баз данных

- оценка уровня опасности уязвимостей базы данных, что включает в себя **обнаружение** скомпрометированных конечных точек и **классификация** конфиденциальных данных;
- постоянное управление **правами доступа** пользователей и предотвращение наличия чрезмерных привилегий и неактивных пользователей;
- **обучение** сотрудников методам снижения рисков, что включает в себя получение знаний о распространенных киберугрозах, таких как фишинговые атаки и эксплуатация электронной почты;
- **отслеживание** всей активности, связанной с получением доступа к базе данных в режиме реального времени, для обнаружения утечек данных, несанкционированных SQL, а также атак на протоколы и системы;
- автоматизация проведения **аудита** с помощью специально подобранной платформы;
- **блокировка** вредоносных веб-запросов;
- архивация внешних данных, **шифрование** информации в базе данных и **маскировка** ее полей, чтобы скрыть конфиденциальную информацию.

Рекомендации Center for Internet Security, Inc. (CIS) и Security Technical Implementation Guides (STIGs)

- Разработка **показателей** и рекомендаций для совершенствования программ обеспечения безопасности и соответствия требованиям
- Создание **базовых уровней** конфигурации безопасности систем
- **Сопоставление** текущего состояния объекта с требуемым

Scoring Information

A scoring status indicates whether compliance with the given recommendation impacts the assessed target's benchmark score. The following scoring statuses are used in this benchmark:

Scored

Failure to comply with "Scored" recommendations will decrease the final benchmark score. Compliance with "Scored" recommendations will increase the final benchmark score.

Not Scored

Failure to comply with "Not Scored" recommendations will not decrease the final benchmark score. Compliance with "Not Scored" recommendations will not increase the final benchmark score.

Recommendations	9
1 Installation and Patches	9
1.1 Ensure packages are obtained from authorized repositories (Not Scored)	9
1.2 Ensure Installation of Binary Packages (Not Scored)	12
1.3 Ensure Installation of Community Packages (Not Scored)	14
1.4 Ensure systemd Service Files Are Enabled (Scored)	17
1.5 Ensure Data Cluster Initialized Successfully (Scored)	19
2 Directory and File Permissions	21
2.1 Ensure the file permissions mask is correct (Scored)	21
2.2 Ensure the PostgreSQL pg_wheel group membership is correct (Scored)	23
3 Logging Monitoring And Auditing	26
3.1 PostgreSQL Logging	26
3.1.1 Logging Rationale	26
3.1.2 Ensure the log destinations are set correctly (Scored)	27
3.1.3 Ensure the logging collector is enabled (Scored)	29
3.1.4 Ensure the log file destination directory is set correctly (Scored)	31
3.1.5 Ensure the filename pattern for log files is set correctly (Scored)	33
3.1.6 Ensure the log file permissions are set correctly (Scored)	35
3.1.7 Ensure 'log_truncate_on_rotation' is enabled (Scored)	37
3.1.8 Ensure the maximum log file lifetime is set correctly (Scored)	40
3.1.9 Ensure the maximum log file size is set correctly (Scored)	42

Имеются готовые решения для выявления атак на СУБД но лучше сфокусироваться на превентивных мерах...

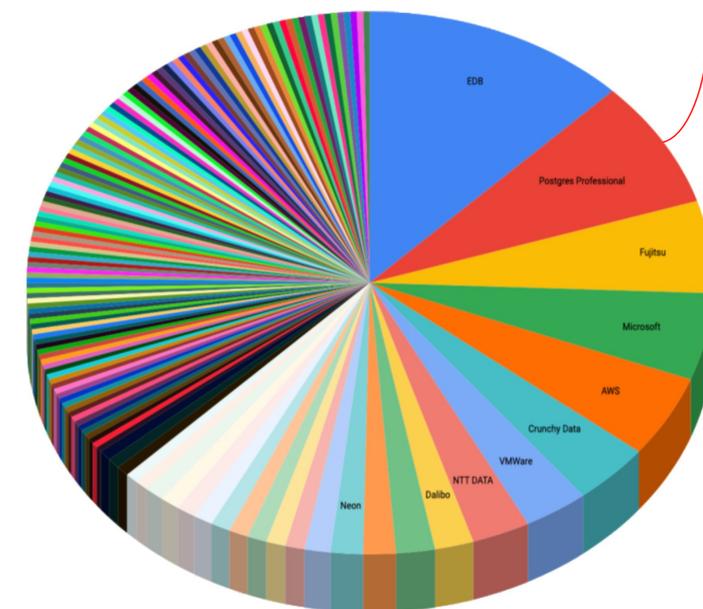
Пакет экспертизы для пользователей PostgreSQL и MaxPatrol SIEM от Positive Technologies помогает «обнаружить такие подозрительные действия, как:

- отправка команд для **определения версии БД** (свидетельствует о начале атаки на СУБД),
- чтение таблиц, содержащих **хеш-суммы паролей**,
- отключение аудита,
- изменение **уровня важности сообщений** аудита для сокрытия действий,
- изменение **метода аутентификации** для повышения вероятности компрометации пользователя или роли в системе,
- отключение **политики защиты** строк для таблицы,
- отключение **шифрования** канала передачи данных (SSL),
- перезагрузка **конфигурации сервера**,
- **запуск** встроенных **приложений** операционной системы с возможностью выполнения произвольных команд.»

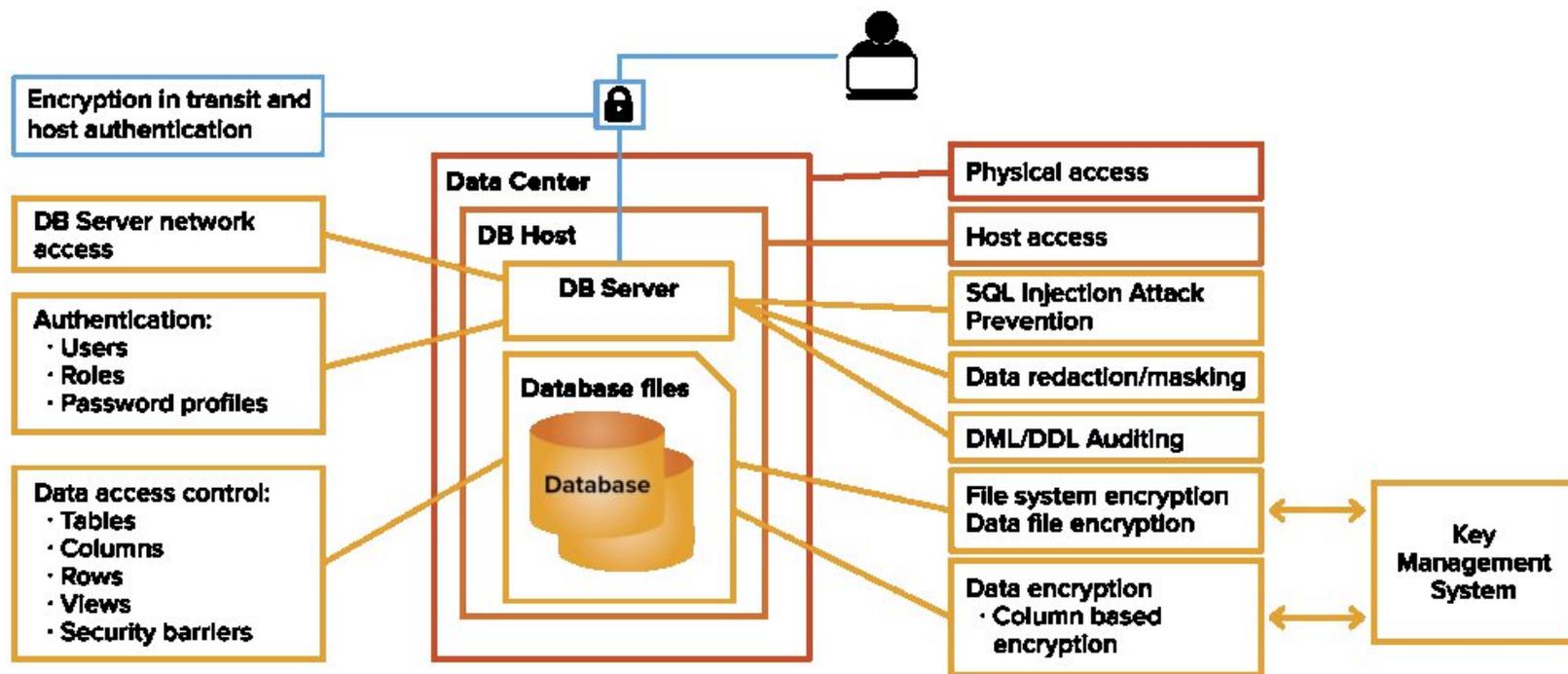
...с помощью наших решений в области ИБ

- ООО "Постгрес Профессиональный" – разработчик и правообладатель ПО класса СУБД на основе PostgreSQL в соответствии с **реестровой** записью №104 от 18.03.2016 в reestr.digital.gov.ru
- ООО «Постгрес Профессиональный» является обладателем **лицензии** ФСТЭК на деятельность по разработке и производству средств защиты конфиденциальной информации (см. reestr.fstec.ru/reg2)
- Сертификация (подтверждение характеристик) проводится с 2016 года сейчас – по «Требованиям по безопасности информации к системам управления базами данных», утвержденным приказом ФСТЭК России от 14 апреля 2023 г. N 64, **самого высокого** для защиты конфиденциальной информации **четвертого класса защиты и уровня доверия** (продукту и его производству)
- Postgres Pro – контрибьютер #2 в код PostgreSQL в мире (после EnterpriseDB)
- Совместимость ПО – это crowdtesting + дальнейшее развитие и поддержка + заплатки для устранения уязвимостей (см. postgresql.org/support/security)

Вклад PostgresPro в разработку PostgreSQL



Подход к организации защиты СУБД



- В PostgreSQL имеется множество решений ИБ
 - Ролевая модель управления привилегиями
 - Row Level Security
 - Безопасность подключений (диапазон IP, порты, таймауты, SSL)
 - Идентификация и аутентификация пользователей (пароли, LDAP, cert, Kerberos)
 - Встроенный аудит (журнал событий)
 - Встроенная криптозащита (пароли, столбцы, на стороне клиента)

Почему этого недостаточно для защиты информации?

- Проблема суперпользователя – бесконтрольность, невозможность отследить опасные действия, риски использования superuser для работы сервисов
- Проблема доступа к чувствительным данным привилегированных пользователей
- Для маскирования данных надо применять внешние утилиты/расширения
- Недостаточная гибкость аудита и повышенная нагрузка от него на СУБД
- Риски безопасности при несанкционированном изменении файлов СУБД
- Неудобная криптография – непрозрачно для приложений, утилит, имеются уязвимости
- Растущие требования регуляторов – к распределению обязанностей, к сложности пароля, к перечню и содержанию регистрируемых событий безопасности, к очистке памяти



Контроль привилегированных пользователей



с 16.1

PostgresPro

- Задача – снизить вероятность того, что суперпользователь в одиночку выполнит несанкционированные действия

Использование запретительного подхода сохраняет лазейки для обхода ограничений, т.к. не существует всеобъемлющего списка того, что у суперпользователя нужно отобрать

Минимальный набор
прав обычного
пользователя



Набор прав
суперпользователя

Контроль привилегированных пользователей



с 16.1

PostgresPro

- Задача – снизить вероятность того, что суперпользователь в одиночку выполнит несанкционированные действия

Использование запретительного подхода сохраняет лазейки для обхода ограничений, т.к. не существует всеобъемлющего списка того, что у суперпользователя нужно отобрать

Минимальный набор прав обычного пользователя



Набор прав суперпользователя

Используем разрешительный подход:

- Создадим администратора БД для настройки конкретной БД
- Создадим администратора СУБД для настройки экземпляра СУБД

```
root: edit pg_hba.conf  
auth type = reject  
chown root & chmod 640
```

- Временно заблокируем суперпользователя с помощью администратора инфраструктуры

- Установим правила журналирования любых действий с пользователями, изменений конфигурации СУБД, изменений хранимых процедур, любых DDL
- Воспользуемся бэкпортом кода (*) из 16-й версии

	13*	14*	15*	16
Postgres Pro Standard		14.10.1+	15.5.1+	16.1.1+
Postgres Pro Enterprise	13.13.1+	14.10.1+	15.5.1+	16.1.1+

Разграничение прав администраторов – реализация

- **Создание роли администратора СУБД**
 - Выполняется от имени суперпользователя
 - Выдача системных привилегий: `CREATEDB`, `CREATEROLE`, `REPLICATION`
 - Выдача функциональных привилегий на большой набор predefined ролей и системных функций (см. следующий слайд)
- **Типовые обязанности администратора СУБД**
 - Создаёт новые БД
 - Создаёт пользователей – администраторов БД
 - Создаёт tablespaces
 - Меняет настройки СУБД
 - Создаёт пользователей для репликации
 - Управляет репликацией
- **Создание роли администратора БД**
 - Выполняется от имени администратора СУБД
 - Выдача системных привилегий: `CREATEROLE`
 - Выдача функциональных привилегий: мини-набор predefined ролей (см. следующий слайд)
- **Типовые обязанности администратора БД**
 - Создаёт таблицы и хранимые функции в БД (т.к. обычным пользователям **запрещены любые изменения кода** информационных систем [`CREATE`]; они не могут создавать или изменять код хранимых процедур, функций, пакетов, триггеров)
 - Создаёт пользователей БД
 - Резервное копирование и восстановление своей БД

«Замена» суперпользователя администратором СУБД...

и адми-
нистра-
тором
БД

- При создании Администратора СУБД ему выдаются права на следующие **предопределенные роли**:
 - `pg_read_all_settings`
 - `pg_read_all_stats`
 - `pg_stat_scan_tables`
 - `pg_monitor`
 - `pg_signal_backend`
 - `pg_checkpoint`
 - `pg_create_tablespace` *(разработка PostgressPro, не требует прав суперпользователя)*
 - `pg_manage_profiles` *(разработка PostgressPro, не требует прав суперпользователя)*

чтение различных полезных параметров конфигурации, статистики и другой системной информации

выполнение экстренных действий на уровне инстанса

создание табличных пространств и управление профилями
- ... и на **системные функции**:
 - `pg_reload_conf()`
 - `pg_rotate_logfile()`
 - `pg_create_restore_point()`
 - `pg_backup_start()`
 - `pg_backup_stop()`
 - `pg_switch_wal()`
 - `pg_promote()`
 - `pg_wal_replay_pause()`
 - `pg_wal_replay_resume()`

управление конфигурацией и логированием

управление бэкапированием и восстановлением

управление журналом предзаписи и репликацией



Как ставить недоверенные расширения без superuser?

- Расширение состоит из управляющего файла и минимум одного SQL-скрипта; оно может быть доверенным или недоверенным. Установка недоверенного расширения обычно требует привлечения суперпользователя...
- Используем то, что trusted-расширение ставится от имени postgres; и при этом SQL-файлы расширения выполняются от суперпользователя без его login к базе, внутренними механизмами
- Безопасность достигается необходимостью вовлечения нескольких человек в установку расширения: Администратора СУБД и Администратора инфраструктуры (безопасности)

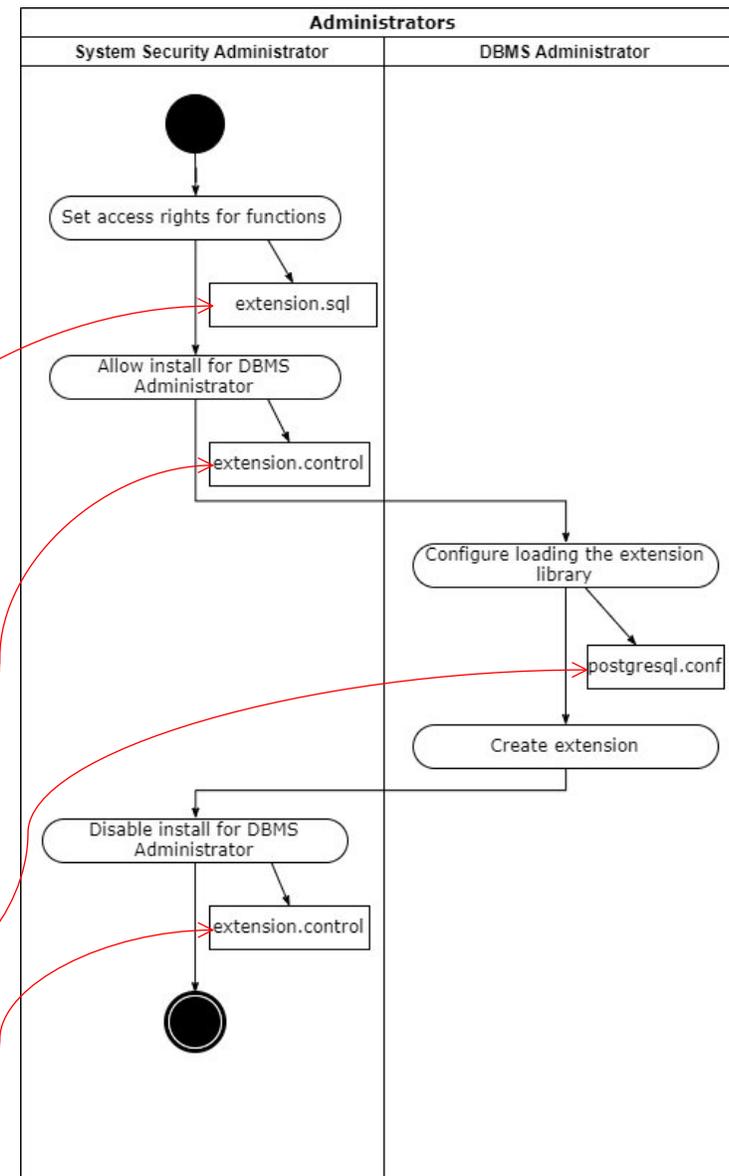
Пример выдачи разрешения Администратору СУБД для установки расширения pg_proaudit

```
-- Create new versions of objects
CREATE FUNCTION pg_proaudit_show()
RETURNS TABLE(
  db_name text,
  event_type text,
  object_type text,
  object_oid oid,
  role_name text)
AS 'MODULE_PATHNAME', 'pg_proaudit_show_conf'
LANGUAGE C VOLATILE;
REVOKE ALL ON FUNCTION pg_proaudit_show() FROM PUBLIC;
GRANT ALL ON FUNCTION pg_proaudit_show() TO PGPRO_DBMS_ADMIN;
```

trusted = true

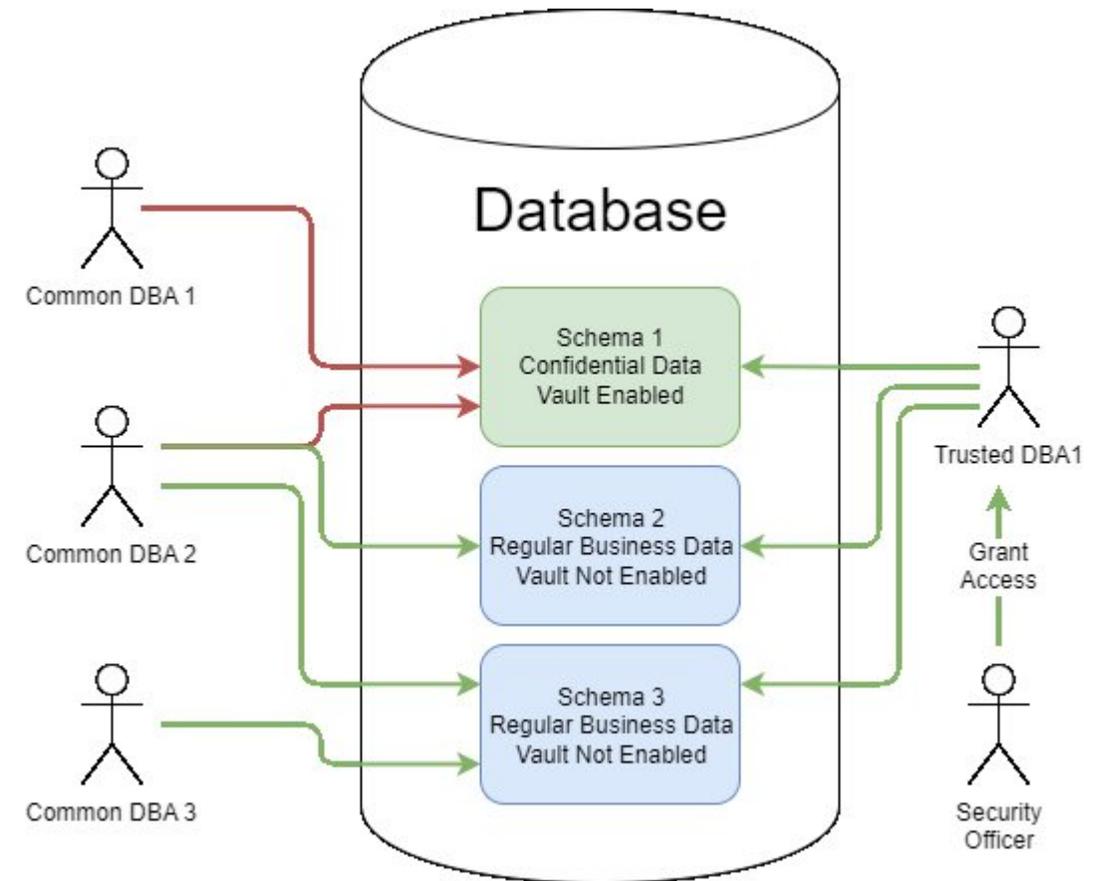
add file to shared_preload_libraries + systemctl restart postgresql

trusted = false



Как ограничить доступ администратора к данным?

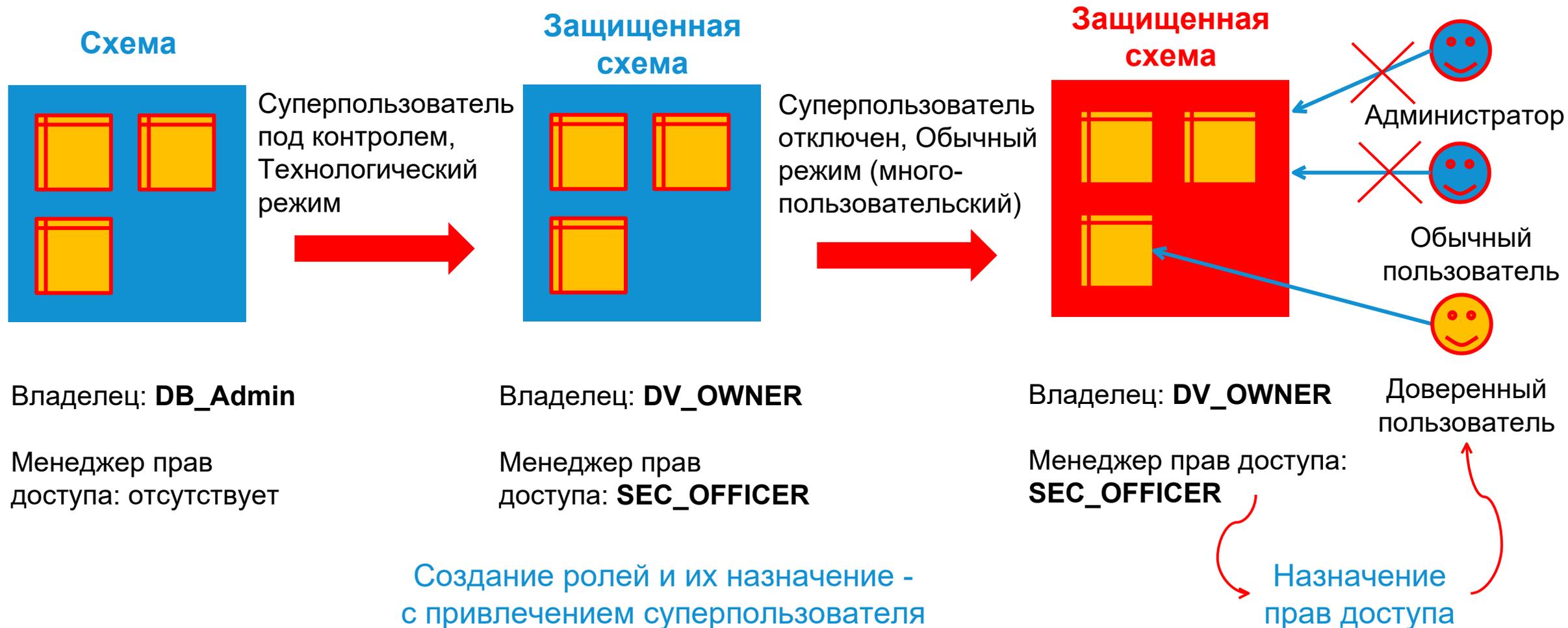
- Предоставить доступ доверенным бизнес-пользователям не проблема, но к сожалению к чувствительным данным получают доступ и администраторы...
- В больших организациях слишком много администраторов имеют доступ к экземпляру СУБД, так как им надо производить его регулярное обслуживание, резервное копирование и т.д.
- Надо отнять у них права доступа к чувствительным данным и коммерческой тайне, передав им «доверенным» администраторам
- В больших организациях доступ к БД раздаёт отдел безопасности, а не Админы СУБД. Нужно, чтобы доступ к чувствительным данным и коммерческой тайне давали тоже только они



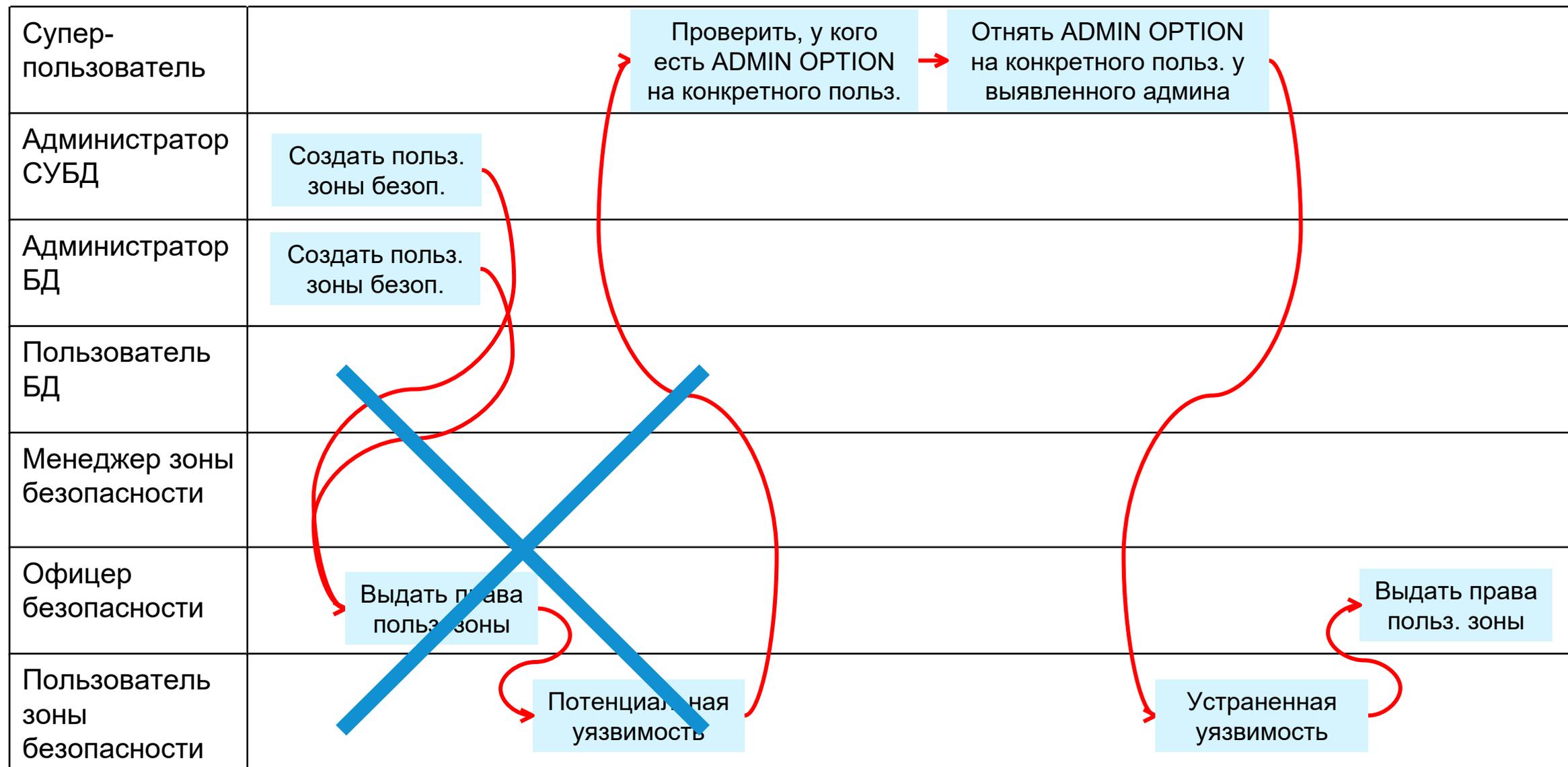
Администратор без доступа к данным – реализация

- Вводится понятие защищенной зоны на уровне схемы
- Вводятся специальные роли для использования в защищенной схеме – ее владельца и офицера безопасности
- Назначение схеме владельца и офицера безопасности превращает ее в защищенную
- Владелец схемы имеет доступ к ее объектам по умолчанию, но теперь администраторы и бизнес-пользователи могут получить доступ к ним только после явного разрешения от офицера безопасности
- Для контроля доступа используется стандартный механизм ACL
- Разработка PostgressPro:
 - Код выдачи прав на доступ к объектам – `pg_class`, `pg_proc`, `pg_type`, `pg_collation`
 - В структуре каталога `pg_namespace` - новое поле `nspsecofficer`, аналогичное `nspowner`
 - Роль **SECURITY OFFICER** внесена в Schema в явном виде
 - В синтаксисе ALTER SCHEMA появилось «**ALTER SCHEMA ... SECURITY OFFICER TO ...**»
 - В `pg_dump` – возможность выгрузки только прав доступа для `pg_restore` под Security Officer

Пример организации защиты данных в схеме



Решение проблемы ADMIN OPTION при организации доступа пользователя защищенной зоны



Решение проблемы ADMIN OPTION при организации доступа пользователя защищенной зоны



Решение проблемы отсутствия общей роли при переносе существующей таблицы в защищённую зону



Изменения в программе для создания резервных копий

- В программе для создания резервных копий базы данных `pg_dump` предусмотрена отдельная выгрузка (а) структуры и данных и (б) прав доступа защищенной зоны

	Порядок выгрузки данных	Порядок загрузки данных
Администратор БД	1. выгрузить все данные, кроме защищённой схемы	1. загрузить все данные, кроме защищённой схемы
Владелец зоны безопасности	2. выгрузить данные защищённой схемы 3. выгрузить права доступа	2. загрузить данные защищённой схемы
Офицер безопасности		3. загрузить права доступа к защищённой схеме



Регистрация событий безопасности

- Задача – расширить охват аудита и снизить нагрузку на СУБД:
pgAudit -> pg_proaudit (v1.0 быстрый, но сложные настройки) -> **pg_proaudit (v2.0)**
- Оптимизированный механизм поиска – правило для аудирования подбирается фильтрацией правил по комбинации
 - Имя базы данных
 - Тип события
 - Тип объекта
 - Имя объекта
 - Имя роли
- Легко настраивать правила
- Логирует, какой пользователь выполнил действие
- Есть логирование по группе пользователей, а не только по конкретному пользователю
- Может логировать любую команду SQL
- Выше быстродействие благодаря параллельной обработке

Типовой «рекомендованный» набор правил аудита

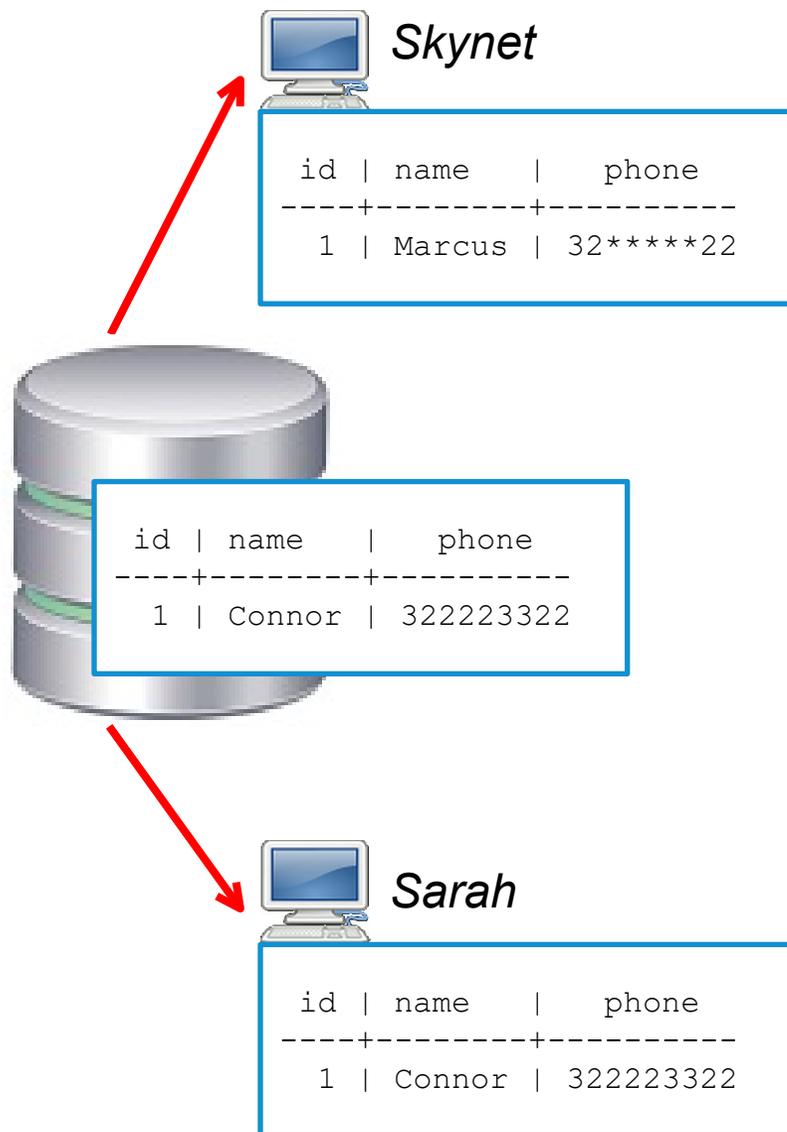
- Удовлетворяет требованиям сертификации ФСТЭК и известным нам типовым требованиям ИБ
- Имитирует типовую настройку с учётом наличия Secured Schema

DB	event_type	object_type	object_name	role_name	comment
	AUTHORIZE				все новые авторизации (соединения)
	DISCONNECT				все окончания соединений
	ALL	ROLE			все действия над пользователями и ролями - create/alter/drop user/role/group
	ALTER SYSTEM				все изменения системной конфигурации
	ALL_DDL				все создания и модификации баз данных, таблиц, представлений, хранимых функция и процедур, ...
	GRANT				все разрешения доступов
HR_DB	ALL_DML	SCHEMA	HR_VAULT		весь доступ к данным внутри безопасной схемы
HR_DB	ALL_PROC	SCHEMA	HR_VAULT		все вызовы хранимок из безопасной схемы
	ALL			SUPPORT	все действия инженеров поддержки
	ALL			PGPRO_DBMS_ADMIN	все действия Администраторов СУБД

Маскирование данных

- Во многих случаях раскрытие данных СУБД нежелательно, и для защиты коммерческих или персональные сведений их надо **заменить либо фейковой, либо частичной информацией**
- Оригинал при этом остается нетронутым
- `pgpro_anonymizer` – расширение для маскирования или замены конфиденциальных данных внутри экземпляра PostgresPro
- В проекте используется декларативный подход к анонимизации. Вы можете объявлять правила маскирования, используя язык описания данных (DDL), и задавать свою стратегию анонимизации внутри определения таблицы
- Используются следующие основные стратегии:
 - **Динамическое** маскирование изменяет представление реальных данных, не модифицируя их. Некоторые пользователи могут читать только замаскированные данные, а другие могут получить доступ к исходной версии
 - **Статическое** маскирование полностью заменяет конфиденциальную информацию несвязанными данными. После такой обработки исходные данные не могут быть восстановлены

Пример динамического маскирования данных



Правила маскирования задаются метками безопасности:

```
SELECT anon.start_dynamic_masking();
```

```
SECURITY LABEL FOR anon ON COLUMN people.name  
IS 'MASKED WITH FUNCTION anon.fake_first_name();'
```

```
SECURITY LABEL FOR anon ON COLUMN people.phone  
IS 'MASKED WITH FUNCTION anon.partial(phone,2,$$*****$$,2);'
```

```
SECURITY LABEL FOR anon ON ROLE Skynet, Mark IS 'MASKED';
```

В результате доверенный пользователь с ролью Sarah получит немодифицированные данные, а недоверенный пользователь с ролью Skynet – строку с фейковым именем и частично сокрытым номером телефона

Варианты изменения данных при маскировании

- **Удаление** просто удаляет данные.
- **Статическая замена** последовательно заменяет данные одним и тем же значением. Например: замена всех значений столбца типа text на значение «КОНФИДЕНЦИАЛЬНО».
- **Отклонение** «сдвигает» даты и числовые значения. Например, после применения отклонения +/- 10% к столбцу зарплаты набор данных не потеряет смысл.
- **Обобщение** снижает точность данных, заменяя их диапазоном значений. Вместо «Бобу 28 лет» можно сказать «Бобу от 20 до 30 лет». Этот метод полезен для аналитики, так как данные остаются верными.
- **Перестановка** перемешивает значения в рамках столбца. Исходные данные могут быть восстановлены, если алгоритм перестановки будет расшифрован.
- **Рандомизация** заменяет конфиденциальные данные случайными, но правдоподобными значениями.
- **Частичное скрывтие** аналогично статической подстановке, но оставляет часть данных нетронутыми
- **Пользовательские правила** предназначены для изменения данных в соответствии с особыми требованиями (например, заменить одновременно почтовый индекс и название города случайными значениями, чтобы они оставались согласованными).
- **Псевдонимизация** защищает персональные данные, скрывая их с помощью дополнительной информации. Шифрование и хеширование — два примера методов псевдонимизации. Псевдонимизированные данные, тем не менее, по-прежнему остаются связаны с исходными данными.



Управление парольными политиками

- Расширенные политики аутентификации, обеспечивающие эффективное управление паролями и контроль доступа
- Реализация – через настройку параметров ролей:
 - с помощью профилей, которые задают парольные политики (длину, сложность, срок жизни, число неудачных попыток входа до блокировки и т.п.) - CREATE PROFILE
 - блокирование и разблокирование - ALTER ROLE



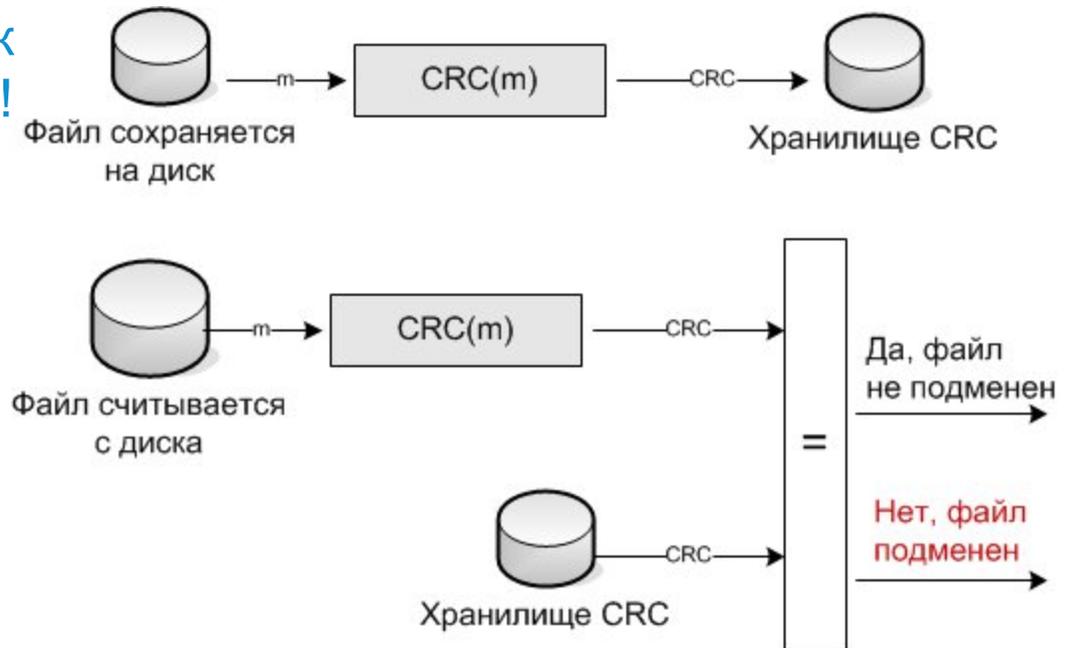
Контроль целостности

- Требования предъявляются не к данным СУБД, а к исполняемым и конфигурационным файлам СУБД!

- Реализуется утилитой **pg_integrity_check**
- Расчет контрольных сумм учитывает содержимое файла, его атрибуты, время изменения
- Имеются отдельные настройки для неизменяемых файлов (**бинарники**) и файлов, которые могут быть изменены администратором баз данных (**конфигурационные файлы, системные таблицы**).

Для первых контрольные суммы поставляются готовыми, для вторых необходимо провести расчеты при вводе в промышленную эксплуатацию

- При старте СУБД контроль целостности бинарников выполняется автоматически; при несовпадении контрольных сумм требуется вмешательство root для принятия изменений или отката



Почему требуется очистка памяти



с 14.1

PostgresPro

Файл помечен, как удаленный, но информацию можно восстановить

Проведена очистка удаленного файла с заполнением адр. пространства нулями

Sector Edit - [HD1: VBOX HARDDISK (4.0 GB, H:)]

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII
0000	33	C0	8E	D0	BC	00	7C	8E	C0	8E	D8	BE	00	7C	BF	00
0010	06	B9	00	02	FC	F3	A4	50	68	1C	06	CB	FB	B9	04	00Ph.....
0020	BD	BE	07	80	7E	00	00	7C	0B	0F	85	0E	01	83	C5	10
0030	E2	F1	CD	18	88	56	00	55	C6	46	11	05	C6	46	10	00V.U.F...F..
0040	B4	41	BB	AA	55	CD	13	5D	72	0F	81	FB	55	AA	75	09	.A..U..]r...U.u.
0050	F7	C1	01	00	74	03	FE	46	10	66	60	80	7E	10	00	74t..F.f'~.t
0060	26	66	68	00	00	00	00	66	FF	76	08	68	00	00	68	00	&fh....f.v.h..h.
0070	7C	68	01	00	68	10	00	B4	42	8A	56	00	8B	F4	CD	13	h..h...B.V.....
0080	9F	83	C4	10	9E	EB	14	B8	01	02	BB	00	7C	8A	56	00V.
0090	8A	76	01	8A	4E	02	8A	6E	03	CD	13	66	61	73	1C	FE	.v..N..n...fas..
00A0	4E	11	75	0C	80	7E	00	80	0F	84	8A	00	B2	80	EB	84	N.u..~.....
00B0	55	32	E4	8A	56	00	CD	13	5D	EB	9E	81	3E	FE	7D	55	U2..V...]}>..}U
00C0	AA	75	6E	FF	76	00	E8	8D	00	75	17	FA	B0	D1	E6	64	.un.v....u.....d
00D0	E8	83	00	B0	DF	E6	60	E8	7C	00	B0	FF	E6	64	E8	75d.u
00E0	00	FB	B8	00	BB	CD	1A	66	23	C0	75	3B	66	81	FB	54f#;u;f..T
00F0	43	50	41	75	32	81	F9	02	01	72	2C	66	68	07	BB	00	CPAu2....r,fh...
0100	00	66	68	00	02	00	00	66	68	08	00	00	00	66	53	66	.fh....fh....fSf
0110	53	66	55	66	68	00	00	00	00	66	68	00	7C	00	00	66	SfUfh....fh. ..f
0120	61	68	00	00	07	CD	1A	5A	32	F6	EA	00	7C	00	00	CD	ah....Z2... ...
0130	18	A0	B7	07	EB	08	A0	B6	07	EB	03	A0	B5	07	32	E42.
0140	05	00	07	8B	F0	AC	3C	00	74	09	BB	07	00	B4	0E	CD<.t.....
0150	10	EB	F2	F4	EB	FD	2B	C9	E4	64	EB	00	24	02	E0	F8+.d..\$.
0160	24	02	C3	49	6E	76	61	6C	69	64	20	70	61	72	74	69	\$.Invalid parti
0170	74	69	6F	6E	20	74	61	62	6C	65	00	45	72	72	6F	72	tion table.Error
0180	20	6C	6F	61	64	69	6E	67	20	6F	70	65	72	61	74	69	loading operati
0190	6E	67	20	73	79	73	74	65	6D	00	4D	69	73	73	69	6E	ng system.Missin
01A0	67	20	6F	70	65	72	61	74	69	6E	67	20	73	79	73	74	g operating syst
01B0	65	6D	00	00	63	7B	9A	6A	A6	2A	3E	00	00	00	20	00	em...c{.j.*>...
01C0	21	00	07	E8	9F	09	00	08	00	00	00	E8	7F	00	00	00	!.....
01D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
01E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
01F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55	AAU.

Offset: 000 Block: N/A Sector size: 512 Total sectors: 8388608

Sector Edit - [HD1: VBOX HARDDISK (4.0 GB)]

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII
0000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
01A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
01B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
01C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
01D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
01E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
01F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Offset: 000 Block: N/A Sector size: 512 Total sectors: 8388608

Очистка памяти – уточнение задачи СУБД

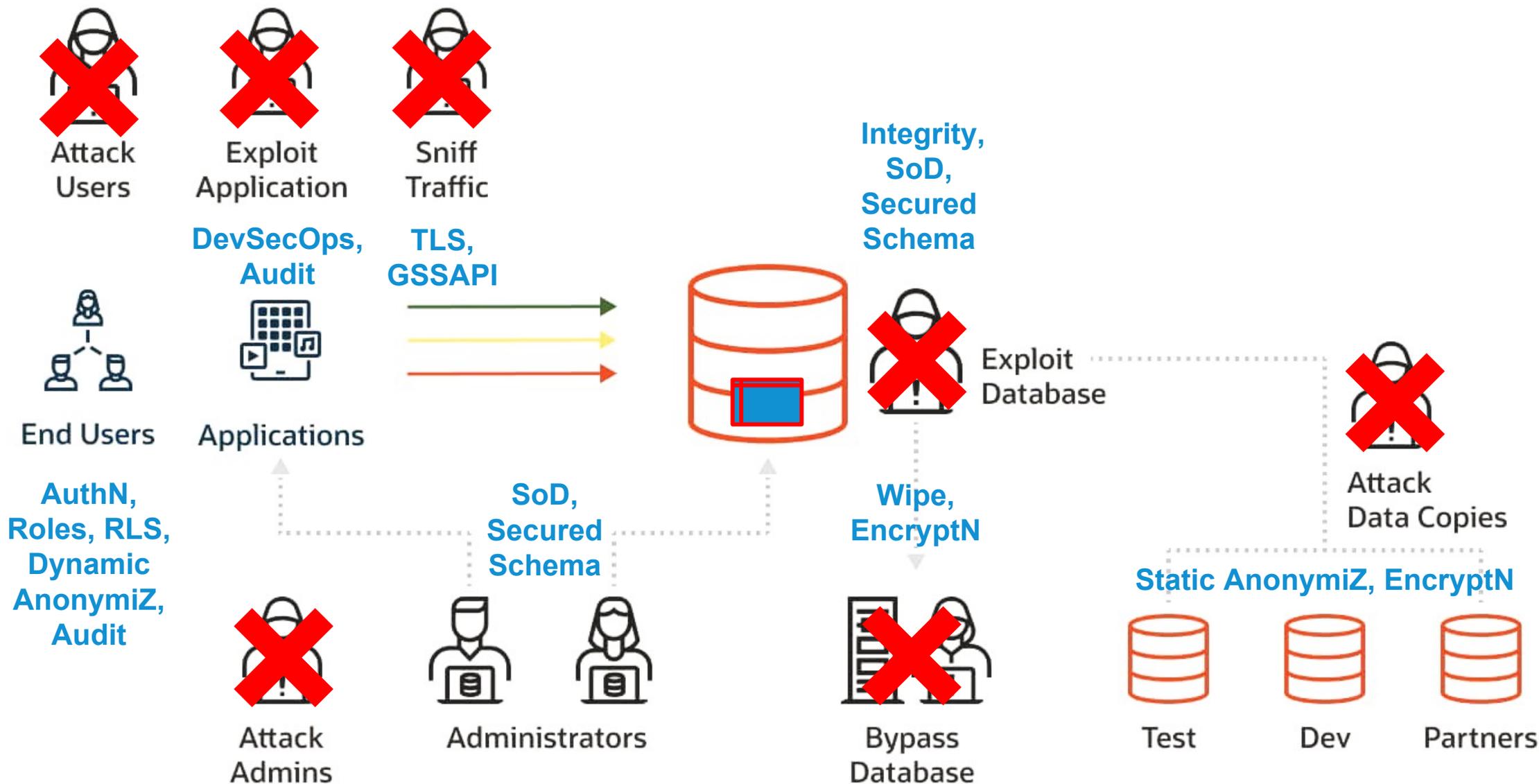
- Удаление баз данных и журналов, используемых СУБД, путём многократной перезаписи уничтожаемых **файлов** – не задача Postgres; в Linux реализуется например командами **wipe** или **shred**
- Postgres может заполнить пустышками (нулевыми байтами) модифицированные **участки файлов** при выполнении команд DELETE и VACUUM (см. postgrespro.ru/docs/enterprise/16/memory-purge-cert)
- **Внимание! Очень ресурсоёмкая операция...**
- Немного разные решения используются для
 - Удаления файлов с дисков
 - Очистки страниц (Multiversion Concurrency Control → VACUUM)
 - Очистки блоков памяти в ОЗУ (MemoryContexts)
 - Очистки файлов WAL (удаление или перезапись сегментов)

Доступная криптозащита данных

- Это не шифрование по определению ФСБ! (см. п.4 в clsz.fsb.ru/clsz/license.htm)
- Шифрование паролей
 - Предпочтительный метод – SCRAM
- Шифрование избранных столбцов
 - Модуль **pgcrypto** позволяет хранить в зашифрованном виде избранные поля. Чтобы прочитать эти поля, клиент передаёт дешифрующий ключ, сервер расшифровывает данные и выдаёт их клиенту. **Уязвимо** в процессе расшифровывания и передачи данных из-за потенциального перехвата системным администратором
- Шифрование раздела данных
 - Шифрование хранилища данных можно реализовать на уровне файловой системы или на уровне блоков. **Уязвимо** при хранении ключа на компьютере, который монтирует ФС
- Шифрование данных при передаче по сети
- Проверка подлинности сервера SSL
- Шифрование на стороне клиента
 - Вся логика д.б. на стороне клиента (приложения) – потенциальный **удар по производительности!**

- **Transparent Data Encryption** для защиты данных в состоянии покоя
- Будут реализован отсутствующий у конкурентов комплекс:
 - Механизмы AES-шифрования разных типов данных – страниц данных, индексов, материализованных представлений, временных таблиц, файлов WAL
 - Механизмы работы с ключами – иерархия, места хранения, ротация, вызов неактивных ключей
 - Авторизация нескольких утилит на использование криптографии в фоне
 - Инструкции по изменениям в работе с кластером СУБД, репликами и резервными копиями
 - Порядок развертывания решения и смены алгоритма шифрования
- Цель – разработка решения с модульной архитектурой, обеспечивающей возможность интеграции с различными СКЗИ для использования альтернативных алгоритмов шифрования
- Также в разработке – **утилита для поиска неиспользуемых привилегий** в разрезе конкретных пользователей и групповых ролей

Нейтрализация угроз средствами защиты СУБД





Сертифицированные редакции СУБД

Standard

Современная СУБД, включает все новые функции PostgreSQL и полезные доработки от компании

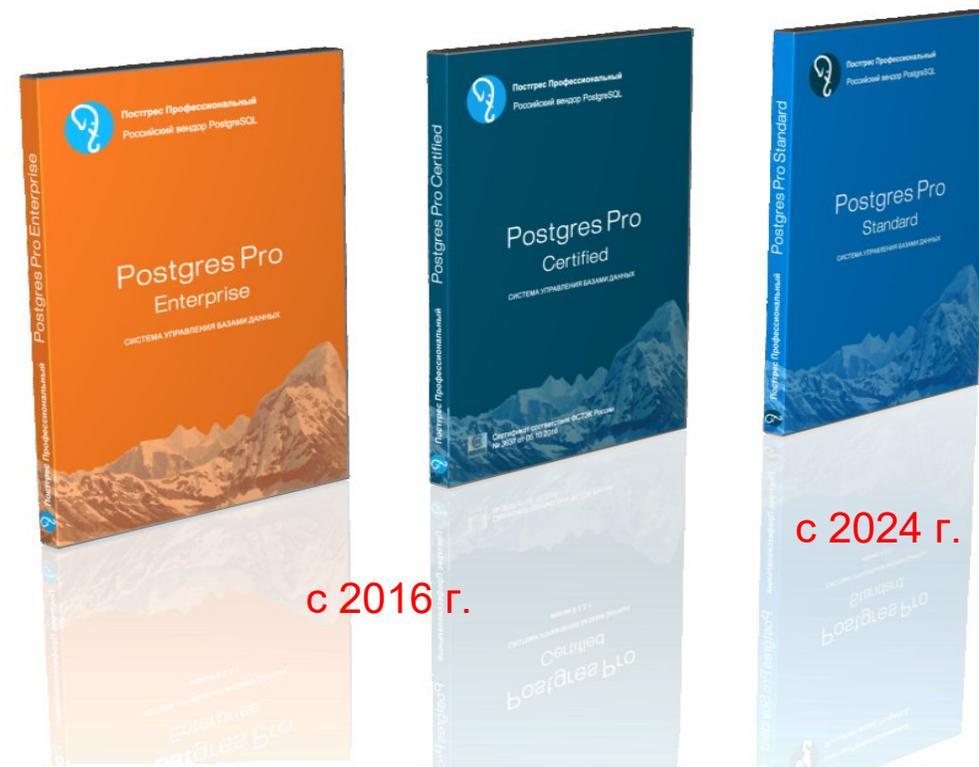
Enterprise

Наиболее полнофункциональная СУБД с высокой производительностью и масштабируемостью

Shardman

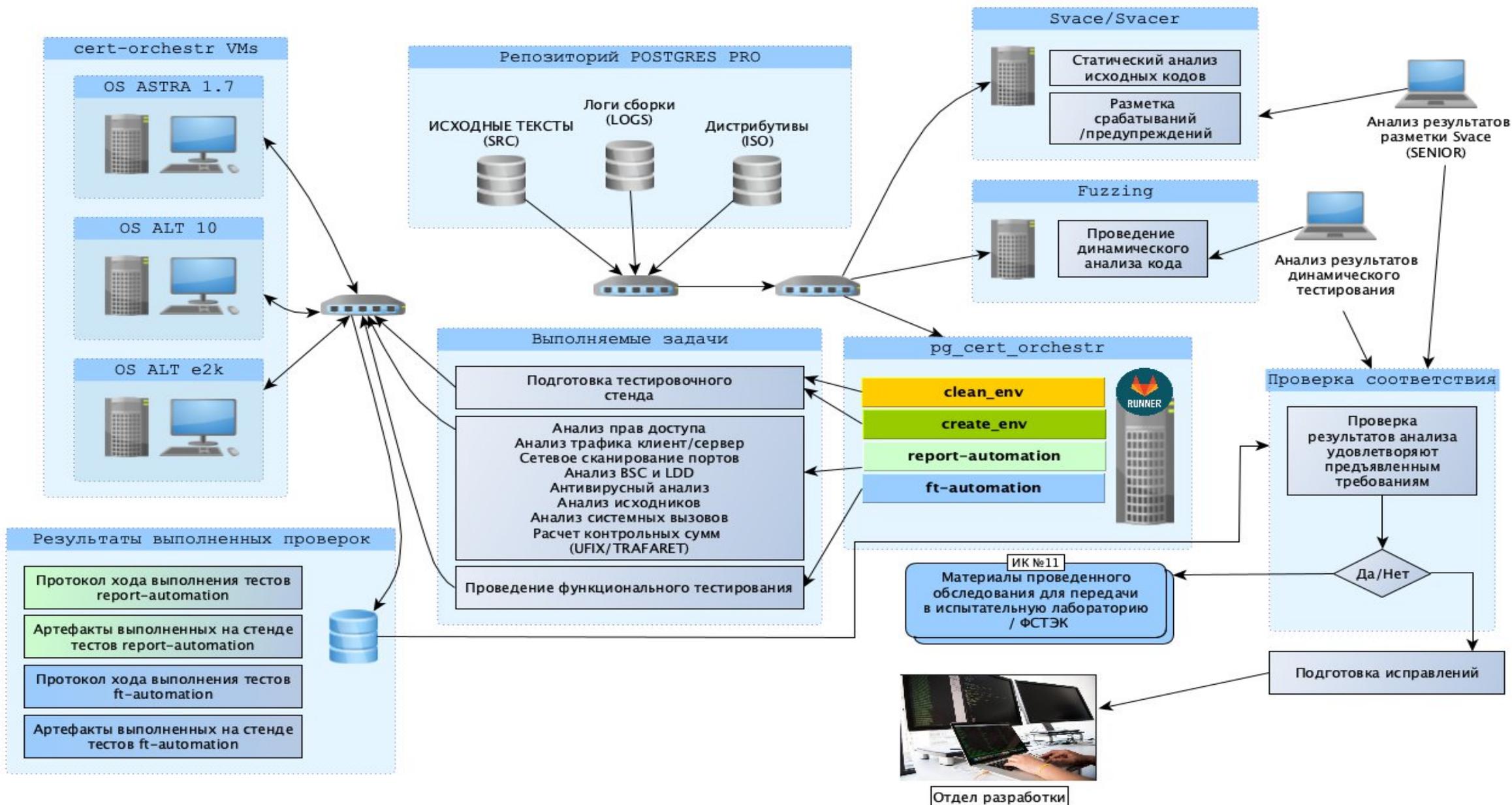
Распределенная СУБД, предоставляющая строгие гарантии целостности данных

соответствуют «Требованиям по безопасности информации к СУБД» ФСТЭК России самого высокого для защиты конфиденциальной информации четвертого класса защиты и уровня доверия



с ежеквартальным обновлением в рамках инспекционного контроля

Процесс выпуска сертифицированных обновлений



Где можно использовать наши сертифицированные версии

- Системы управления базами данных, соответствующие 4 классу защиты, применяются:
 - в значимых объектах критической информационной инфраструктуры 1 категории значимости,
 - в государственных информационных системах 1 класса защищенности,
 - в автоматизированных системах управления производственными и технологическими процессами 1 класса защищенности,
 - в информационных системах персональных данных при необходимости обеспечения 1 уровня защищенности персональных данных,
 - в информационных системах общего пользования II класса

Все это – безопасность инвестиций!

Безопасность разработки

Востребованный функционал, Стабилизация, Вклад в комьюнити-версии

Безопасность эксплуатации

Быстрая доступность обновлений, Скорость и качество поддержки

Юридическая безопасность

Сертифицированные версии, Собственная лаборатория, Инспекционный контроль

Средства безопасности СУБД

Ролевая модель, Маскирование данных, Ограничение прав суперпользователя, Продвинутый аудит, Защита данных

PosgresPro



**Спасибо
за внимание!**

