УДК 004.056.52, 004.658

О СОВМЕСТНОМ ПРИМЕНЕНИИ МЕТОДОВ АВТОМАТИЗИРОВАННОГО АУДИТА И МЕХАНИЗМОВ ВРЕМЕННОГО ДЕЛЕГИРОВАНИЯ ПОЛНОМОЧИЙ ДЛЯ ОПТИМИЗАЦИИ КОНТРОЛЯ ДОСТУПА В СУБД

Тимофеенко А.А., студент гр. ИИби-231, III курс
Научный руководитель: Матисов А.В., к.т.н.
Кузбасский государственный технический университет им. Т. Ф. Горбачева

1. Введение

Разработан программный комплекс для автоматизации аудита и управления привилегиями в системах управления базами данных (СУБД). Система объединяет модуль мониторинга текущих прав, аналитический блок сравнения с эталонными моделями и механизм управления временными доступами. В отличие от традиционных методов ручного администрирования, требующих регулярного анализа системных таблиц СУБД, предложенное решение формирует динамическую картину прав доступа, позволяет мгновенно выявлять «дрейф полномочий» (privilege drift) и устранять несоответствия без ручного написания SQL-запросов.

Безопасность данных в корпоративных СУБД напрямую зависит от актуальности прав пользователей. Наличие избыточных или устаревших привилегий увеличивает риски несанкционированного доступа. Классический подход к аудиту через чтение логов и ручную сверку ролей является трудоемким процессом. Предложенный подход позволяет автоматизировать контроль, заменить рутинные операции быстрой визуализацией отклонений и реализовать безопасный механизм предоставления краткосрочных прав.

2. Постановка задачи

Ключевым объектом проектирования является программный модуль (Identity Management), решающий задачи по обеспечению информационной гигиены в базах данных. Система реализует следующий функционал:

- **Консолидация данных о субъектах:** агрегация списка активных учетных записей и сопоставление их с ролевой моделью.
- **Автоматическая верификация:** выявление расхождений между фактическим набором прав и утвержденным профилем с визуальной индикацией критических отклонений.
- **Принудительная синхронизация:** механизм мгновенного отзыва неавторизованных привилегий для приведения системы к безопасному состоянию.
- **Динамическое делегирование:** функционал выделения временных доступов с автоматическим ограничением по времени (TTL), что

реализует принцип минимально необходимых полномочий.

- **Протоколирование:** ведение событийного журнала для последующего анализа безопасности и разбора инцидентов.

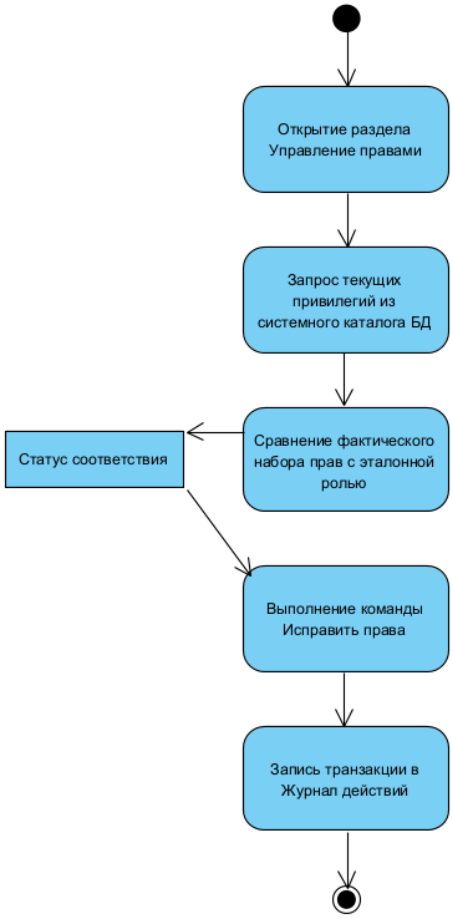
3. Архитектура программного комплекса

Архитектура решения базируется на взаимодействии управляющего интерфейса с сервисным слоем, который транслирует команды непосредственно в движок СУБД. Работа с программным комплексом разделена на два сценария:

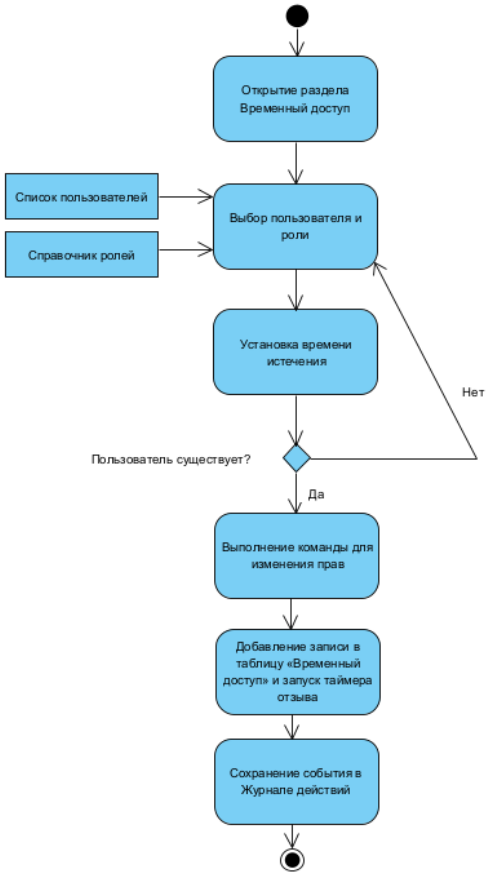
Режим «Аудит и Сверка» (Реактивный контроль) На этом этапе система производит сканирование системных таблиц прав, сравнивает полученный вектор привилегий с «золотым образом» роли и выводит уведомление о несоответствии. В случае обнаружения лишних или отсутствующих прав администратор инициирует автоматическую процедуру исправления (fix-up).

Режим «Временные привилегии» (Проактивное управление) Этот сценарий предназначен для разовых задач (например, внеплановых работ разработчика в PROD-контуре). Система генерирует запись в реестре временных доступов, активирует необходимые права в БД и запускает таймер обратного отсчета. По истечении срока действия система автоматически деактивирует выданные полномочия, исключая «забытые» открытые доступы.

Детальная последовательность операций и взаимодействие компонентов системы визуализированы в следующем разделе на диаграммах деятельности.



Аудит и сверка



Временные привилегии

3.1. Логика работы модуля управления (IDM-движок)

Процесс функционирования системы автоматизации прав строится на циклической обработке состояний учетных записей. Алгоритм включает следующие последовательные этапы:

1. **Сбор данных:** идентификация активных сессий и выгрузка текущих привилегий из системного каталога СУБД;
2. **Верификация:** сопоставление полученного вектора прав с шаблоном «эталонной роли»;
3. **Детекция:** присвоение статуса безопасности (визуализация несоответствий через флаг WARNING);
4. **Инициация изменений:** формирование очереди команд на добавление или отзыв прав (Grant/Revoke);
5. **Контроль времени:** мониторинг жизненного цикла временных доступов по метке времени (TTL);
6. **Деактивация:** автоматическое аннулирование прав по истечении заданного интервала;
7. **Аудит:** сохранение итогового состояния в журнале транзакций.

Структурные компоненты логической модели:

- **DB_CONNECTOR** — сервисный модуль с высоким уровнем привилегий для прямого взаимодействия с СУБД;

- **ROLE_METADATA** — хранилище описаний «золотых образов» ролей (эталонов);
- **TTL_SCHEDULER** — планировщик задач, отвечающий за проверку и очистку просроченных доступов;
- **AUDIT_SINK** — подсистема сбора и хранения логов административных действий.

Параметризация системы включает:

- **User_Set** — массив контролируемых учетных записей;
- **Reference_Map** — матрица соответствия между пользователями и их целевыми ролями;
- **Interval_Value** — настраиваемая длительность временного окна доступа (по умолчанию — 5 минут);
- **Status_Flags** — булевы маркеры состояния целостности прав (OK/WARNING).

Особенностью реализации является работа системы в режиме реального времени: при каждом обновлении интерфейса IDM-движок инициирует процедуру сверки, обеспечивая актуальность данных. Логика обработки исключений реализована через механизмы проверки успешности выполнения SQL-команд, что предотвращает рассинхронизацию между интерфейсом управления и фактическим состоянием базы данных.

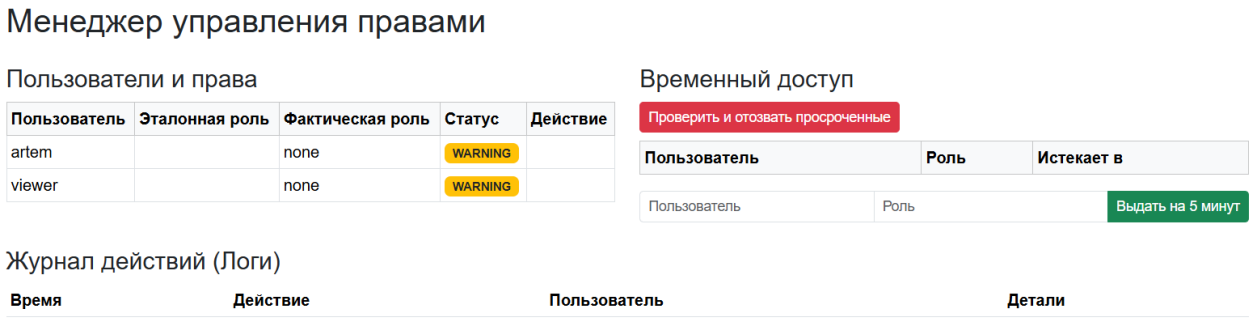


Рисунок 1 – Матрица соответствия прав и ролей пользователей

3.2. Сервисный уровень взаимодействия на C# (Blazor)

Центральным звеном приложения является компонент UserService, реализованный на языке C#. Он выступает в роли оркестратора между веб-интерфейсом и движком базы данных. Основные функции модуля:

- **Агрегация состояний:** выполнение асинхронных запросов к системным таблицам СУБД для получения актуального среза прав;
- **Управление транзакциями:** инициация команд GRANT и REVOKE через объект подключения IDbConnection;
- **Персистентность метаданных:** работа с локальными таблицами IDM для хранения эталонных ролей и логов.

Для обмена данными между интерфейсом и базой используется микро-ORM Dapper, что позволяет избежать избыточных накладных расходов. Пример формирования команды на изменение привилегий представлен ниже:

```
public async Task FixRightsAsync(string username, string expectedRole) { using var connection =
_db.CreateConnection();
await connection.ExecuteAsync($"GRANT {expectedRole} TO {username
await LogActionAsync(connection, "Fix Rights", username, $"Granted {expectedRole}"); }
```

3.3. Аналитический модуль и контроль временного доступа

В отличие от классических систем мониторинга, данный модуль выполняет интеллектуальный аудит отклонений и управляет временными полномочиями. Алгоритм работы модуля разделен на четыре этапа:

1. **Детекция отклонений:** Анализ связей в системных представлениях pg_user, pg_auth_members и pg_roles. С помощью LEFT JOIN с таблицей ожидаемых ролей (idm_expected_roles) система выявляет пользователей, чей статус не соответствует «золотому образу».
2. **Валидация TTL (Time-To-Live):** Проверка реестра временных доступов на предмет истечения срока действия привилегий.
3. **Автоматическая ревокация:** Массовый отзыв просроченных прав для минимизации поверхности атаки.
4. **Логирование инцидентов:** Каждое изменение прав фиксируется в таблице idm_logs для обеспечения прослеживаемости (accountability).

Ключевая логика контроля временного доступа реализована через фильтрацию активных сессий по временной метке:

```
public async Task RevokeExpiredAccessAsync()
{
using var connection = _db.CreateConnection();
var expired = await connection.QueryAsync<TempAccess>(
"SELECT * FROM idm_temp_access WHERE is_active = TRUE AND expires_at <= CURRENT_TIMESTAMP");

foreach (var access in expired)
{
await connection.ExecuteAsync($"REVOKE {access.RoleName} FROM {access.Username};");
await connection.ExecuteAsync("UPDATE idm_temp_access SET is_active = FALSE WHERE id = @Id", new {
access.Id });
await LogActionAsync(connection, "Revoke Temp", access.Username, $"Revoked {access.RoleName}");
}
}
```

В качестве оптимизатора безопасности выступает процедура сопоставления фактической роли с эталоном. Если значение в системном каталоге возвращает none (через COALESCE), система автоматически генерирует событие WARNING, сигнализируя администратору о необходимости вмешательства. Результат работы модуля визуализируется в веб-интерфейсе, предоставляя администратору инструменты для мгновенного исправления конфигурации.

4. Методы и алгоритмы управления

4.1. Формирование базы аудита

Каждая сессия сканирования СУБД фиксирует:

- текущий перечень активных ролей пользователей;
- наличие привилегий на выполнение DDL/DML операций;
- статус временных разрешений в реестре `idm_temp_access`. Данные консолидируются в оперативной памяти приложения и логируются в таблицу `idm_logs`.

4.2. Алгоритм верификации прав

Используется конвейер обработки данных:

- **Идентификация:** получение `usename` и `rolname` из системных каталогов `pg_user` и `pg_roles`.
- **Сопоставление:** выполнение `LEFT JOIN` с таблицей эталонов `idm_expected_roles`.
- **Классификация:** присвоение статуса безопасности.

Система автоматически подсвечивает записи, где фактическая роль не совпадает с целевой конфигурацией.

4.3. Оптимизация параметров трассы

Оптимизируются только параметры, которые пользователь не вводил, включая:

- параметры экскаватора;
- параметры автосамосвалов;
- технологические характеристики;
- итоговую производительность.

Оптимизация выполняется по функции:

$$F(\mathrm{x}) = -\hat{Y}(\mathrm{x}),$$

где $\hat{Y}$– прогнозируемая производительность.

5. Результаты

В результате работы программного комплекса:

- сформирована модель автоматизированного выявления «дрейфа прав» без прямого написания SQL-запросов администратором;
- достигнуто ускорение реакции на инциденты избыточных прав за счет визуальной индикации (WARNING);
- реализован механизм «безопасного делегирования», исключающий человеческий фактор (забытый отзыв прав).

Пример результата работы:

- автоматически сформированная команда `REVOKE` для просроченных доступов;
- синхронизация текущего состояния БД с эталонной политикой в один клик;
- детализированный журнал действий для внутреннего аудита.

6. Обсуждение

Предложенный подход позволяет:

- отказаться от трудоемкого ручного мониторинга системных таблиц;
- решать задачу оперативного предоставления доступа без ущерба для безопасности;
- исследовать динамику изменений прав доступа внутри организации;
- интегрировать IDM-модуль в общую систему мониторинга ИТ-инфраструктуры;
- минимизировать риски, связанные с привилегированными учетными записями.

7. Заключение

Разработан программный комплекс, объединяющий функционал мониторинга, автоматического аудита и динамического управления доступом. Система позволяет поддерживать контур безопасности СУБД в актуальном состоянии и исключать накопление избыточных прав пользователей за счет автоматизации циклов выдачи и отзыва полномочий.

8. Список литературы

1. **Уильямс, Д.** «Управление доступом и идентификацией: принципы и практика». — М.: Вильямс, 2020. — 432 с. (Фундаментальное руководство по принципам IAM/IDM систем).
2. **Петров, А. С.** «Автоматизация аудита прав доступа в реляционных СУБД» // Сборник трудов конференции «Информационная безопасность в цифровой экономике». — 2024. — С. 112–118. (Источник по теме автоматизации проверок привилегий).
3. **Рид, К., Клеменс, Д.** «PostgreSQL 15. Изнутри. Архитектура, администрирование, безопасность». — СПб.: Питер, 2023. — 512 с. (Базовый источник по системным каталогам и ролевой модели PostgreSQL).
4. **Кузнецов, В. В.** «Методы обеспечения принципа наименьших привилегий в корпоративных информационных системах» //

Диссертация, Московский государственный технический университет, 2023. (Актуальная работа по теме Zero Trust и временных доступов).
5. **Фримен, А.** «Blazor и .NET 8: Создание современных веб-приложений на C#». — М.: ДМК Пресс, 2024. — 600 с. (Техническая база для реализации интерфейса и сервисного слоя приложения).