PROFESSIONAL Postgres

МОДЕЛЬ МАНДАТНОГО УПРАВЛЕНИЯ ДОСТУПОМ ДЛЯ POSTGRESQL

Валерий Попов
v.popov@postgrespro.ru

Олег Иванов
o.ivanov@postgrespro.ru

План доклада

- Для чего нужно разграничение доступа к информации
- Требования регуляторов
- Методы разграничения доступа
- Особенности реализации МАС
- Модель безопасности
- Реализация модели – СУБД Синергия БД

Термины и определения-РД. Защита от НСД

Информация	сведения (сообщения, данные) независимо от формы их представления
Доступ к информации	ознакомление с информацией, ее обработка (insert, update, delete,..)
Конфиденциальная информация	требующая защиты информация

Конфиденциальная информация

Указ Президента РФ «Об утверждении перечня сведений конфиденциального характера»:

- персональные данные
- тайна следствия и судопроизводства
- служебная тайна
- профессиональная тайна
- коммерческая тайна
- сведения о сущности изобретения, … до официальной публикации информации о них

Термины и определения-2

Субъект доступа	лицо или процесс, действия которого регламентируются ПРД
Объект доступа	контейнер информации в системе, доступ к которой регламентируется ПРД
Правила разграничения доступа (ПРД)	совокупность правил, регламентирующих права субъектов доступа к объектам доступа

Common Criteria -1

Стандарт **ISO/IEC 15408** "Общие критерии оценки безопасности информационных технологий" — его российский эквивалент ГОСТ Р ИСО/МЭК 15408

Иерархия функциональных требований безопасности (ФТБ) – классы, семейства, компоненты.

ФТБ определяют политики функций безопасности (ПФБ).

Для защиты данных ПФБ управления доступом (FDP_ACC) и ПФБ управления информационными потоками (FDP_IFC)

Common Criteria - 2

Область действия политик - 3 множества:

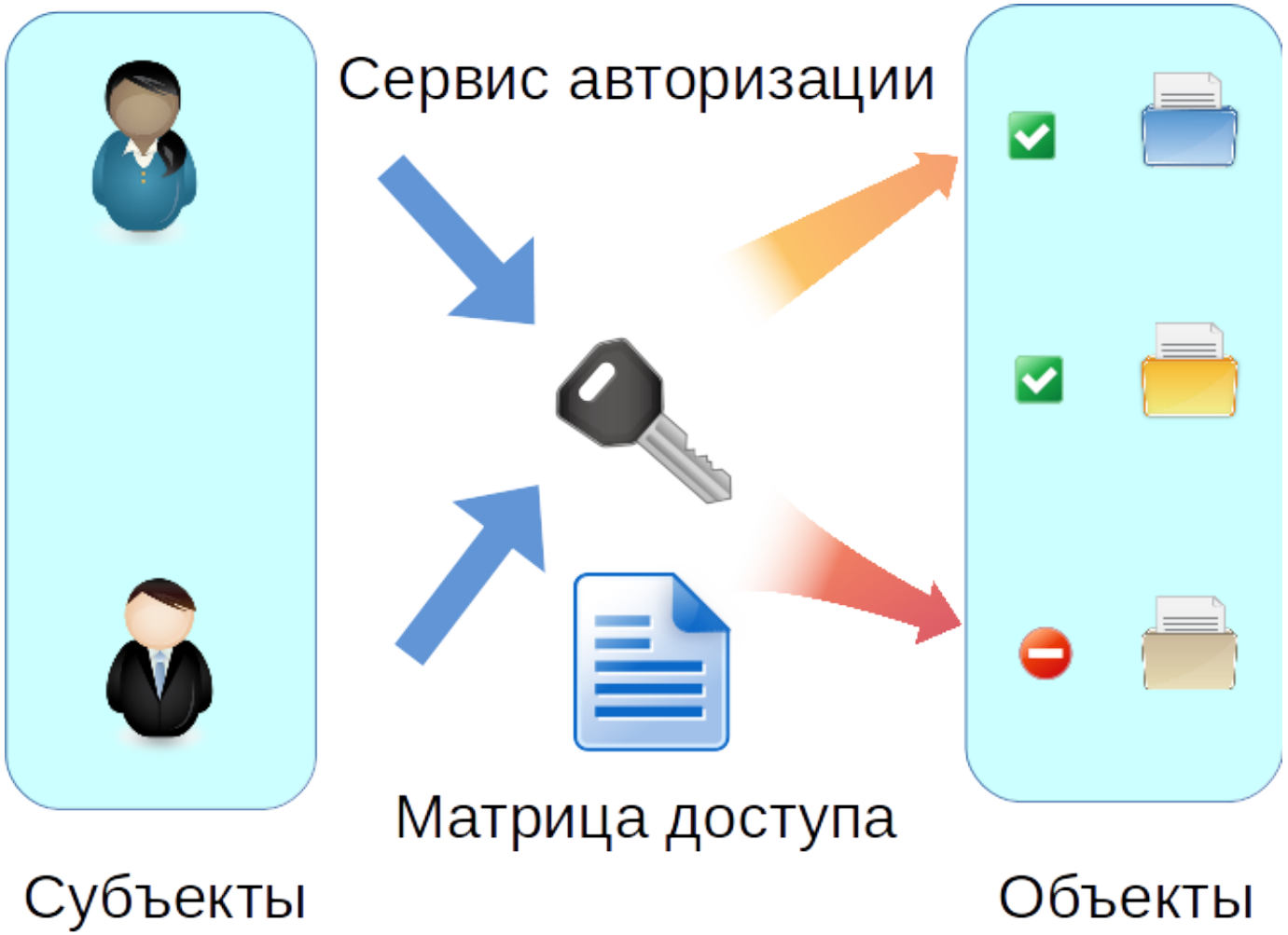
- Субъекты под управлением политики
- Объекты под управлением политики
- Операции управляемых субъектов на управляемых объектах, на которые распространяется политика

Политик может быть несколько –> итерация компонентов семейства по одному разу для каждой политики

Методы управления доступом

DAC	Дискреционный принцип контроля доступа - Discretionary Access Control
MAC	Мандатный принцип контроля доступа - Mandatory Access Control
RBAC	Доступ на основе ролей - Role Based Access Control

Дискреционный принцип контроля доступа - DAC-1



Дискреционный принцип контроля доступа - DAC-2

Основные принципы построения DAC:

1)Все объекты и субъекты идентифицированы
2)Каждый объект системы имеет привязанного к нему субъекта, называемого владельцем. Именно владелец устанавливает права доступа к объекту.
3)Для каждой пары субъект-объект задано явное и недвусмысленное перечисление типов доступа
4)Система может иметь одного выделенного субъекта — суперпользователя, который имеет право устанавливать права владения для всех остальных субъектов системы.
5)Субъект с определенным правом доступа может передать это право любому другому субъекту

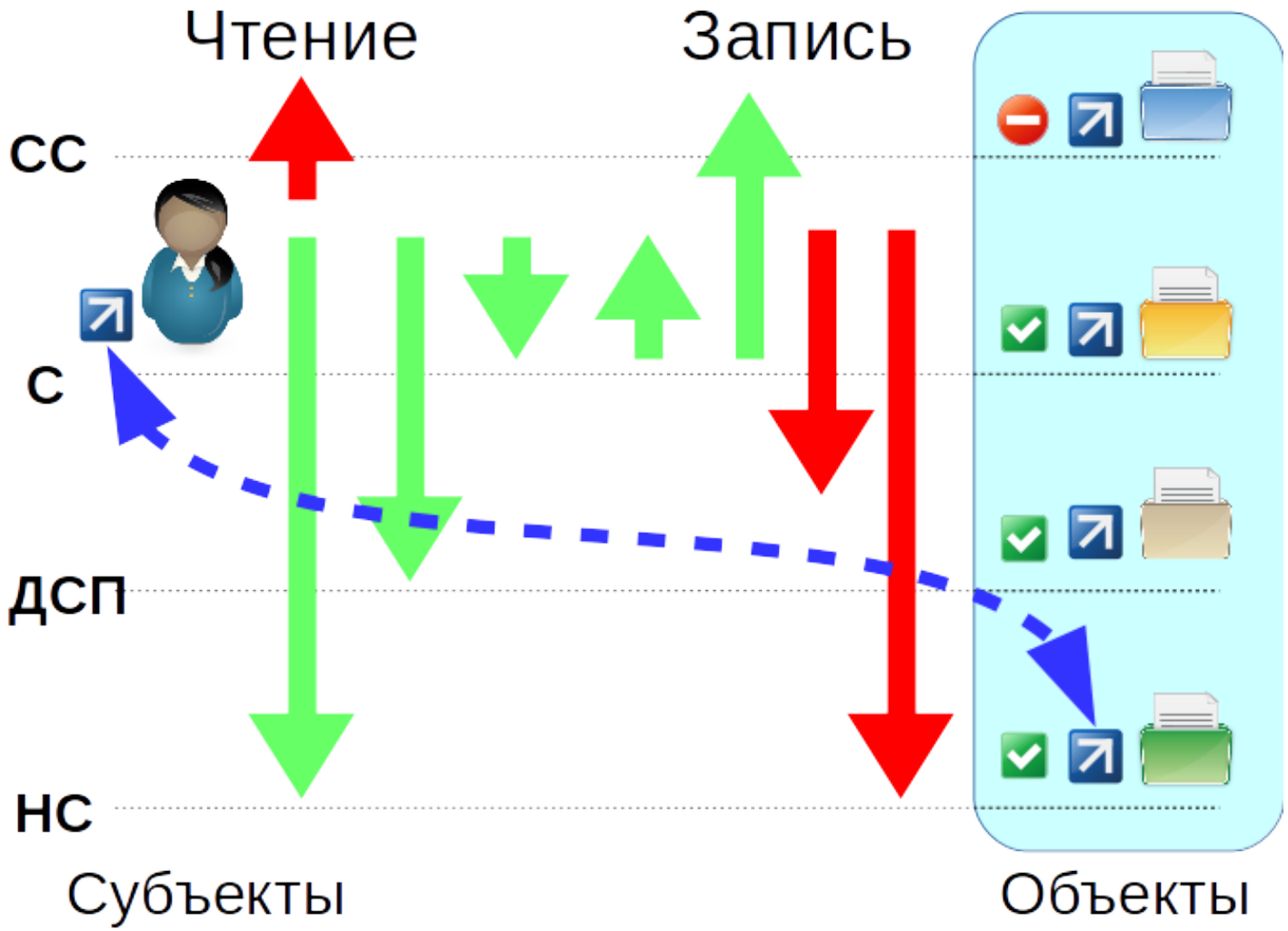
Дискреционный принцип контроля доступа - DAC-4

- Достоинства: гибкость, простота, поддерживается многими системами
- Недостатки: Системы с DAC разрешают пользователям полностью определять доступность их ресурсов, что означает, что они могут случайно или преднамеренно передать доступ неавторизованным пользователям.
- Дискреционный принцип доступа является основной реализацией разграничительной политики доступа к ресурсам при обработке **конфиденциальных** сведений, согласно требованиям к системе защиты информации.

Мандатный принцип контроля доступа (MAC)-1

- Мандатный контроль доступа управляет доступом на основе классификации объектов и субъектов системы. Каждому субъекту и объекту системы назначается некоторый уровень безопасности (УБ) - метки.
- Уровень безопасности объекта - описывает важность этого объекта.
- Уровень безопасности субъекта является уровнем доверия к нему.
- Наряду с УБ можно использовать категории

Мандатный принцип контроля доступа (МАС)-2



MLS/MCS

- Модель Белла-Лападула (Bell–LaPadula) - No read up, no write down. Write up, read down – иерархические уровни
- Неиерархические категории

Мандатный принцип контроля доступа (MAC)-3

- MAC является основой реализации разграничительной политики доступа к ресурсам при защите секретной информации.
- Пользователь не может полностью управлять доступом к ресурсам, которые он создаёт.
- Реализации:

  SELINUX (Red Hat, Cent OS, Fedora, Debian, Alt Linux, ROSA, ОС Синергия)

  Astra Linux Special Edition (Debian 7 based)

  другие

ОС Astra Linux Special Edition - 1

- ДП-модель П.Н.Девянина *)
- Структура метки

Level: Уровень_3 iLevel: Низкий Categories: {3,0xF}

Level — уровень доступа

ilevel - уровень целостности

Categories - Категория - неиерархическое поле, задается 64-битной маской

Пример:

0000000000000000 Нет Категорий

0000000000000001 Категория_1

ОС Astra Linux Special Edition - 2

- PostgeSQL v.9.2 и v.9.4 + расширения SQL для работы с МАС
- Объекты-контейнеры (pg_global, и т.д....)
- Меткой обладает кортеж (tuple)

SELinux+sepgsql

- Контекст безопасности в SELinux состоит из 4 текстовых полей (4-е поле составное): SELinux_User:Role:Type(Domain):Range (Sensitivity:Category).
  system_u : system_r : postgresql_t : s0-s15:c0-c1023
- Набор политик (targeted, mls/mcs, strict,..)
- Sepgsql – имеет ряд ограничений, нет защиты на уровне строк

Ролевая модель контроля за доступом (RBAC) -1

- Контролирует доступ пользователей к информации на основе типов их активностей в системе.
- Роль можно определить как совокупность действий и обязанностей, связанных с определенным видом деятельности.
- Достаточно указать тип доступа к объектам для роли, а пользователям, в свою очередь, указать их роли.
- Является развитием политики DAC, но при этом обладает свойствами MAC: правила определяют порядок предоставления доступа субъектам компьютерной системы в зависимости от имеющихся (или отсутствующих) у него ролей в каждый момент времени.

Ролевая модель контроля за доступом (RBAC) -2

- Достоинства:
  - простота администрирования
  - иерархия ролей
  - принцип наименьшей привилегии
  - разделение обязанностей
- Технология управления доступом на основе ролей достаточно гибка и сильна, чтобы смоделировать как избирательное управление доступом (DAC), так и мандатное управление доступом (MAC).
- Реализации: SELinux, PostgreSQL, FreeBSD, Solaris, Oracle, ...

Области применения различных моделей

- DAC - все ресурсы системы принадлежат пользователям системы, а значит следить за доступом к ресурсу должен его владелец, т.е. пользователь. Небольшое количество пользователей.
- MAC - четкая централизованная система управления доступа, при которой каждый пользователь имеет ровно столько информации сколько ему требуется, и безопасность или надежность данных является основным приоритетом. Гостайна, военные.
- RBAC - большие организации, с большим количеством разделяемых операций.

Показатели защищенности от НСД

Показатель	Конфид. инф.	Гос. тайна
Дискреционный контроль доступа	✓	✓
Мандатный контроль доступа	✗	✓
Очистка памяти	✓	✓
Изоляция модулей	✓	✓
Идентификация и аутентификация	✓	✓
Регистрация событий	✓	✓
Надежное восстановление	✗	✓
Целостность КСЗ	✓	✓
Тестирование КСЗ	✓	✓
Документация	✓	✓

Система защиты информации

- Что защищаем - объект
- От кого защищаем - субъект
- Методы реализации
  - Физическая защита
  - Административно-организационные
  - Программно-технические
- Правила разграничения доступа

Система защиты информации в ОС

- Что защищаем – объект – файлы, директории
- От кого защищаем – субъект – пользователь
- Типы доступа – чтение, запись, выполнение
- Правила разграничения доступа
  - Дискреционные (для каждого субъекта явно задается список объектов, к которым он имеет доступ)
  - Ролевые (для каждой роли явно задается список объектов и других ролей, к которым она имеет доступ)
  - Мандатные (определены и действуют политики безопасности)

Система защиты информации в PostgreSQL

- Что защищаем – объект – таблица, пространство имен, табличное пространство, база данных и т. д.
- От кого защищаем – субъект – сессия в СУБД
- Права доступа – SELECT, INSERT, UPDATE, EXECUTE, REFERENCE, TRIGGER, CREATE, USAGE, CONNECT, EXECUTE, и т. д.
- Правила разграничения доступа
  - Ролевые
  - Мандатные (уровень секретности объекта и допуска субъекта)

Система защиты информации в PostgreSQL

- Требуется соответствие стандарту SQL
- Проверки целостности конфликтуют с безопасностью
- ...

Зачем нужна математическая модель?

- Описание ПРД на языке множеств
- Непротиворечивость и полнота ПРД
  - Доказательство невозможности перехода системы из безопасного состояния в небезопасное с помощью операций, разрешенных ПРД
  - Нет доказательства – ищем контрпримеры – определение уязвимостей системы
  - Формализация и описание предположений, в которых обеспечивается безопасность информации
- Формальная верификация кода
- Соответствие руководящим документам

Мандатная метка

Мандатная метка состоит из иерархической и неиерархической компонент.

- Иерархическая компонента – это неотрицательное целое число от 0 до 255, соответствующее степени секретности.
- Неиерархическая компонента – некоторое подмножество множества из 64 категорий секретных материалов.

Мандатные метки образуют решетку.

Мандатная метка

Метка А меньше или равна метки Б тогда и только тогда, когда уровень секретности А меньше или равен уровню секретности Б, а множество категорий А является подмножеством множества категорий Б.

$$(24, 0110_2) \succ (23, 0110_2)$$

$$(24, 0110_2) \succ (23, 0100_2)$$

$$(23, 0110_2) \succ (23, 0100_2)$$

$$(24, 0110_2) \not\succ (23, 1110_2)$$

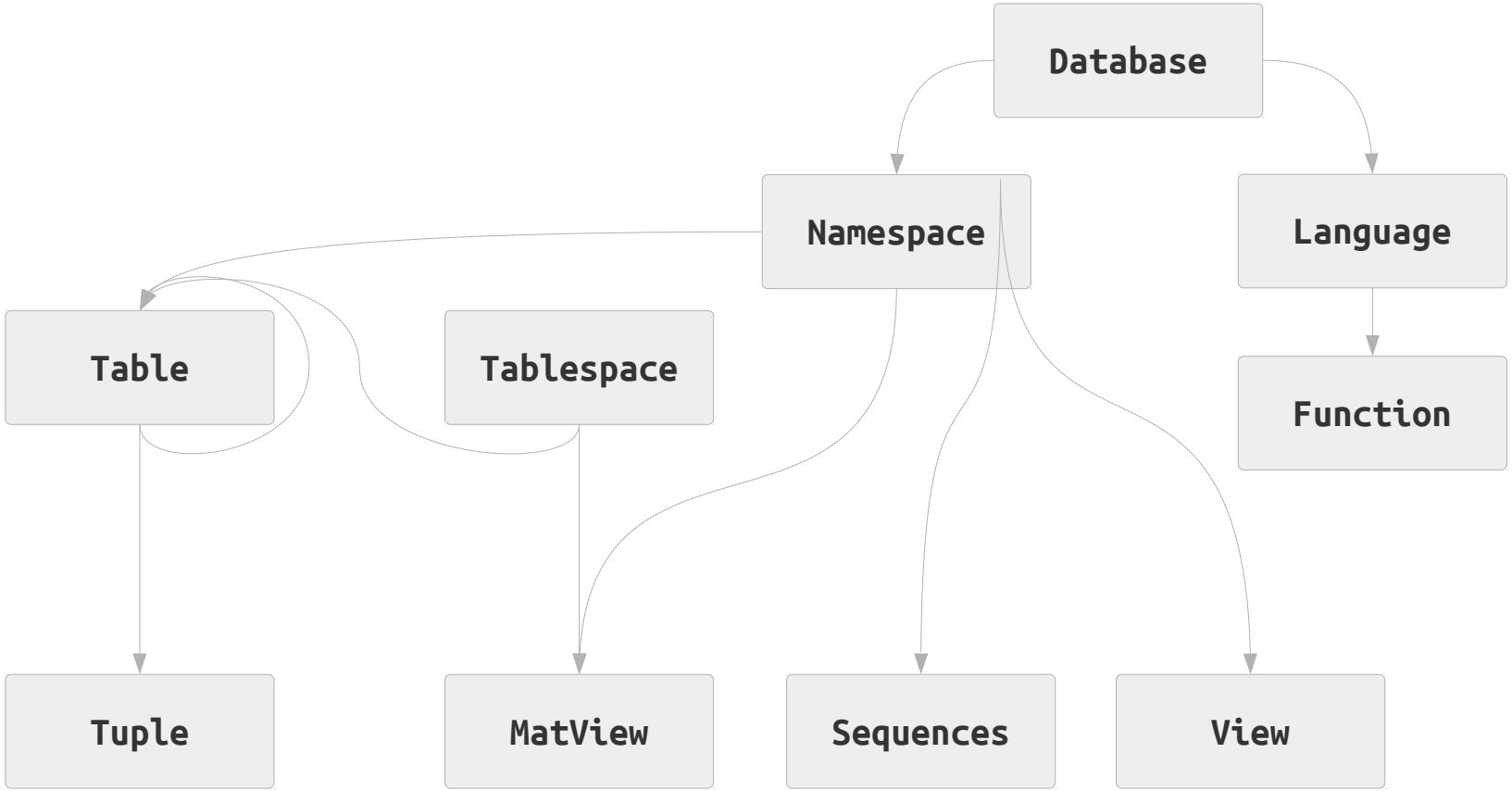
$$(24, 0110_2) \not\prec (23, 1110_2)$$

$$(24, 0110_2) \not\succ (23, 1011_2)$$

$$(24, 0110_2) \not\prec (23, 1011_2)$$

Объекты доступа

Объекты-контейнеры и содержимое контейнеров



Мандатное разграничение доступа

Инварианты:

- Контейнер может содержать объекты только с меньшей или равной мандатной меткой.
- Чтение производится только из объектов, мандатная метка которых меньше или равна мандатной метке сессии.
- Запись производится только в объекты, мандатная метка которых равна мандатной метке сессии.
- Добавление производится только в объекты, мандатная метка которых больше или равна мандатной метке сессии.

Мандатное разграничение доступа

Правило:
Объект называется *видимым* для сессии, если его мандатная метка меньше или равна метке субъект-сессии и все его родители видимы.

Следствие:
Объект не может быть видимым для субъект-сессии, если хотя бы один из его родителей не видим для неё.

Следствие (вырожденность системы):
Видимость объекта определяется видимостью всех соответствующих ему корневых объектов.

Мандатное разграничение доступа

Бит CCR:
В дополнение к мандатной метке каждый из защищаемых контейнеров хранит бит CCR (Container Clearance Required) - является ли его содержимое недоступным для пользователей, не имеющих доступа к этому контейнеру.

Мандатное разграничение доступа

Инварианты:

- Контейнер может содержать объекты только с меньшей или равной мандатной меткой.
- Чтение производится только из объектов, мандатная метка которых меньше или равна мандатной метке сессии **или у которых флаг CRR=false**.
- Запись производится только в объекты, мандатная метка которых равна мандатной метке сессии.
- Добавление производится только в объекты, мандатная метка которых больше или равна мандатной метке сессии.

Мандатное разграничение доступа

Правило:
Объект называется *видимым* для сессии, если его мандатная метка меньше или равна метке субъект-сессии **или CCR=false** и все его родители видимы.

Следствие:
Объект не может быть видимым для субъект-сессии, если хотя бы один из его родителей не видим для неё.

Формальные обозначения

E — множество всех защищаемых объектов доступа.
S — множество субъект–сессий пользователей.
R — множество ролей.
R_r — множество прав доступа к объектам.
R_a — множество видов доступа к объектам.

Формальные обозначения

$role : S \to R$ — функция текущих ролей субъект-сессий.

$P \subseteq E \times R_{p}$ — множество прав доступа к сущностям.

$PA : R \to 2^{P}$ — функция прав доступа к сущностям ролей.

$F \subseteq E \times E$ — множество информационных потоков.

$A \subseteq S \times E \times R_{a}$ — множество доступов субъект-сессий к объектам.

$(LC, \leq)$ — решетка многоуровневой безопасности уровней конфиденциальности.

Формальные обозначения

$f_e : E \to LC$ — функция, задающая уровень конфиденциальности для каждой сущности.

$f_s : S \to LC$ — функция, задающая для каждой субъект-сессии её текущий уровень доступа.

$CCR : E \to \{true, false\}$ — функция, задающая способ доступа к сущностям внутри контейнеров или ролям в иерархии ролей (с учётом их мандатных уровней конфиденциальности).

Постановка задачи

Определим $G=\left(E, S, R, role, PA, f_e, f_s, CCR, A, F\right)$ – состояние системы, обозначим систему как $\sum\left(G*, OP\right)$, где:

$G*$ — множество всех возможных состояний системы.

OP — множество правил преобразования состояний.

$G \vdash_{op} G'$ — переход системы $\sum\left(G*, OP\right)$ из состояния G в состояние G' с использованием правила преобразования состояний $op \in OP$.

В начальном состоянии справедливо $A=F=\varnothing$.

Доказать: невозможность возникновения информационного потока $\left(u,v\right) \in F$, такого что неверно $f_e\left(u\right) \le f_e\left(v\right)$.

Пример правила преобразования состояний

Правило	**Исходное состояние *G***	**Результирующее состояние *G'***
access_update(x, y)	$x \in S, y \in E, lookup(x, y) = true,$ $f_e(y) \leq f_s(x),$ $f_e(z) \geq f_s(x) \forall z \in E : y \in H_E(z),$ $(y, update_r) \in \widetilde{PA}(role(x))$	$A = (A \setminus (S \times \{y\} \times \{read_a\})) \cup$ $\cup \{x, y, write_a\},$ $f_e(y) = f_s(x),$ $F = (F \setminus (\{y\} \times E)) \cup$ $\cup \{(z, y) \vee (x, z, read_a) \in A\}$

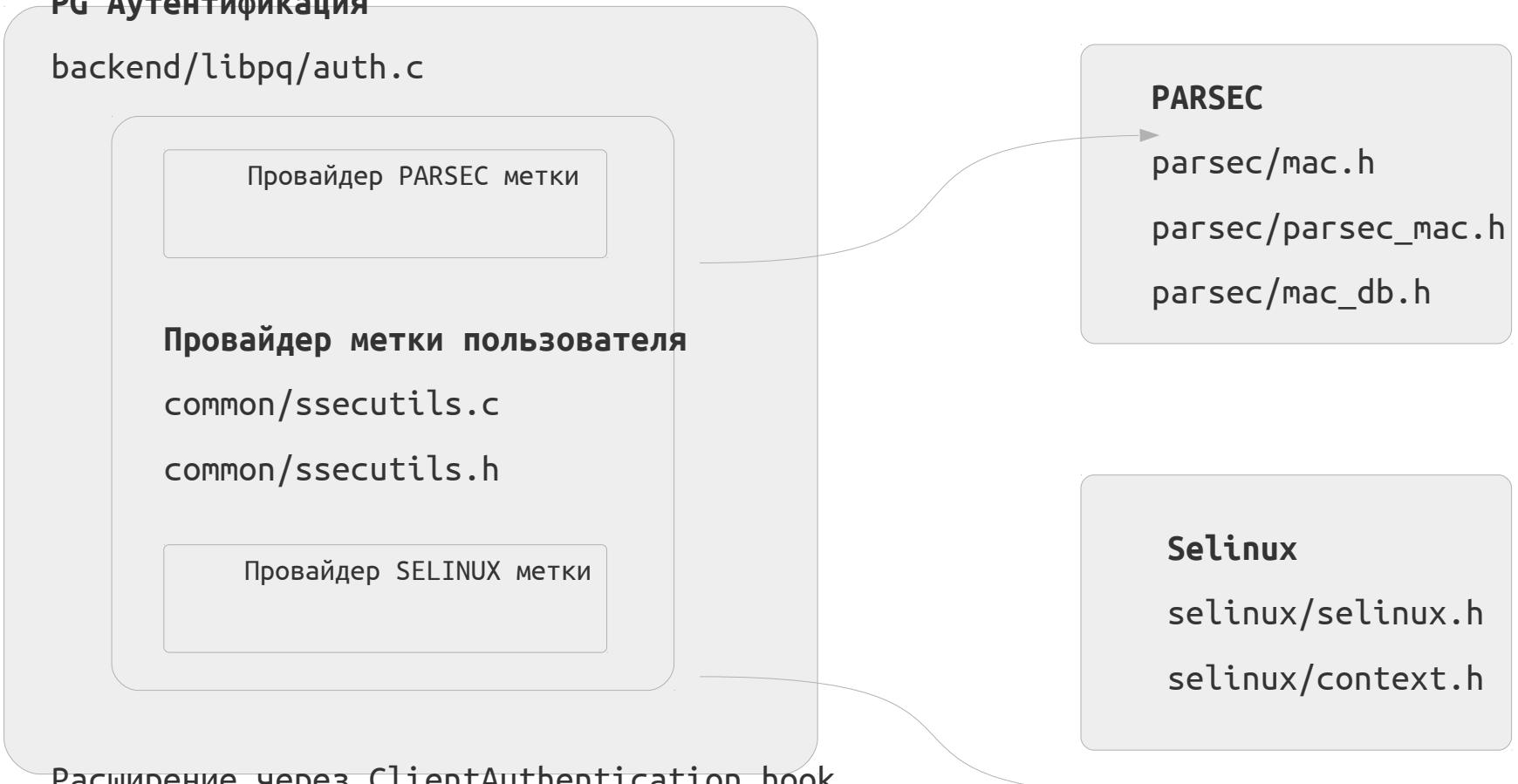
Реализация -СУБД Синергия БД

- Совместимость с Astra Linux
- Соответствует РД СВТ-3
- Работает в ОС Синергия
- Работает в ОС Astra Linux Special Edition

Субъекты доступа

Идентификация и аунтификация

Субъект доступа - пользовательская сессия.

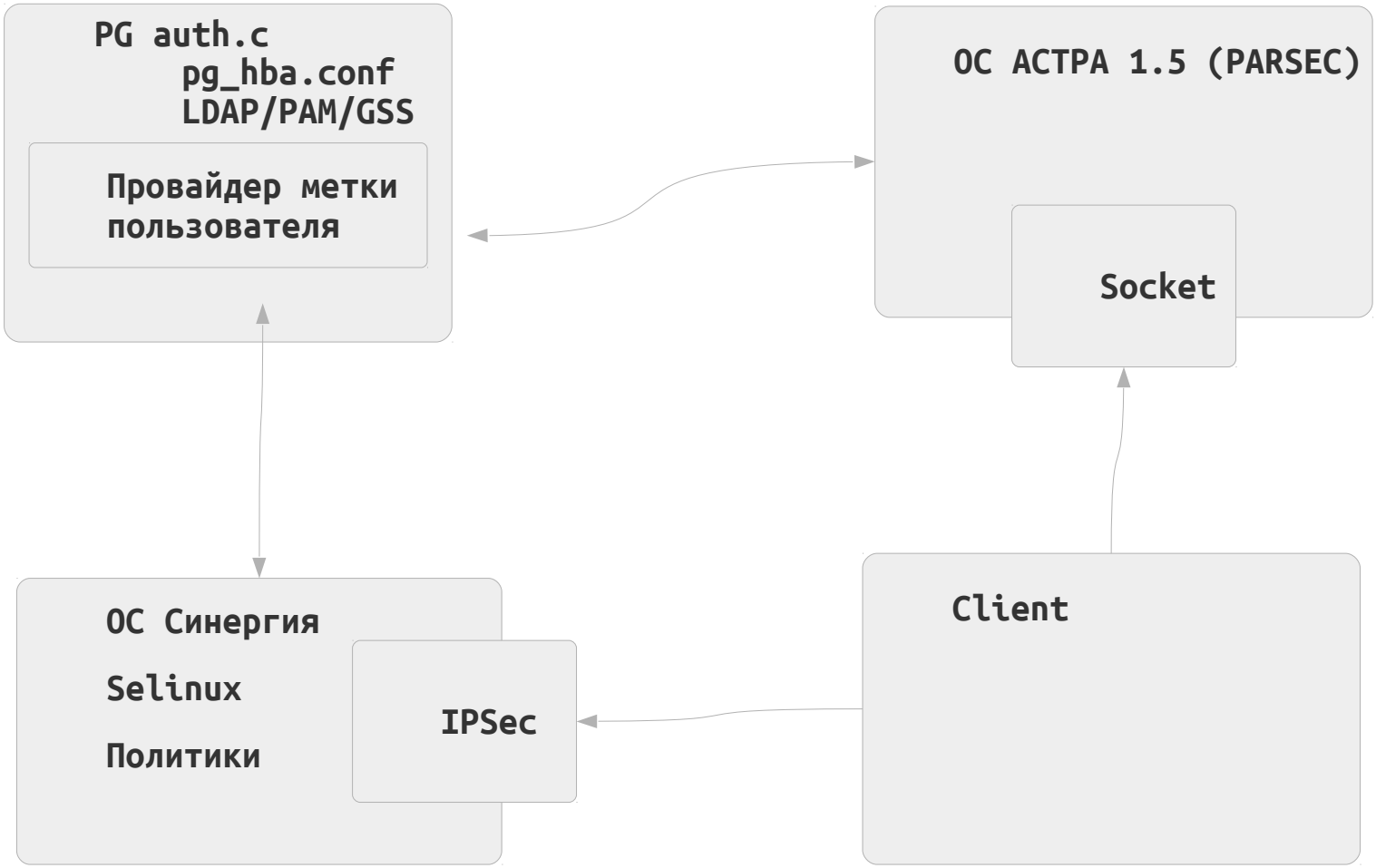


Расширение через ClientAuthentication_hook

Спрашиваем OS права с которыми пришёл пользователь в PG

Субъекты доступа

Идентификация и аунтификация



Политики ОС управляют правами доступа удаленного клиента

Объекты доступа

CREATE: В случае успеха создаваемому объекту присваевается мандатная метка создающей сессии, если объект должен содержать метку, CCR=true если объект является контейнером.

Если для какого-либо из контейнеров, в котором создается объект, мандатная метка не больше или равна мандатной метки пользователя, то создание объекта запрещено.

TABLE, VIEW, MATERIALIZED VIEW, SEQUENCE	Разрешено если родительские схема, таблица (если есть) и табличное пространство видимы для сессии.
TABLESPACE, DATABASE, LANGUAGE, EXTENSION, OPERATOR	Запрещено по умолчанию (дискреционными средствами). Администратор может отменить запрет для некоторых пользователей под свою ответственность. Всегда требуется, чтобы родительские контейнеры были видимы.
RULE, INDEX	Разрешено если таблица видимая.
TRIGGER	Разрешено если функция и таблица видимы.
SCHEMA	Разрешено если родительская база данных видима для сессии.
FUNCTION	Разрешено если язык и схема видимы для сессии.

Объекты доступа

ALTER: все упоминаемые защищаемые объекты должны быть видимыми. При смене структуры контейнеров проверяется невозрастание метки и CCR при переходе от контейнера к его содержимому.

TABLE, VIEW, MATERIALIZED VIEW, SCHEMA	Разрешено если объект видимый и мандатная метка объекта меньше или равняется мандатной метке сессии.
TABLESPACE, DATABASE, LANGUAGE, EXTENSION	Запрещено по умолчанию (дискреционными средствами). Администратор может отменить запрет для некоторых пользователей под свою ответственность.
FUNCTION	Разрешено если объект видимый и его мандатная метка меньше или равна мандатной метке сессии.

REFRESH:

MATERIALIZED VIEW	Разрешено если объект видимый. Мандатная метка объекта (если есть) после выполнения приравнивается мандатной метке сессии.

Примечание: В целях обеспечения безопасности INDEX ONLY SCAN запрещен для защищаемых таблиц, чья мандатная метка отличается от нулевой.

Объекты доступа

Data Manipulation Language	
SELECT	Разрешено для видимых строк видимой таблицы. Мандатные метки результатов совпадают с метками отобранных записей.
INSERT	Разрешено если таблица видима и мандатная метка таблицы больше или равна метки сессии. Мандатной метке вставляемой записи присваивается значение мандатной метке сессии.
UPDATE	Разрешено если таблица видима, ее мандатная метка больше или равна мандатной метки сессии, мандатная метка записи меньше или равна мандатной метке сессии. Мандатной метке обновляемой записи присваивается значение мандатной метке сессии.
DELETE	Разрешено если таблица видима, и мандатная метка записи меньше или равна мандатной метке сессии.
TRUNCATE	Разрешено если таблица видима и мандатная метка таблицы меньше или равна мандатной метке сессии.
Data Control Language	
GRANT	Передавать можно только права, которые одновременно есть у сессии и права передачи которых есть у роли. Нельзя передать права на объект, чья мандатная метка не является меньше или равной мандатной метки передающей сессии.
REVOKE	Разрешено.
Другие действия	
CONNECT DATABASE	Разрешено если база данных видима для субъект-сессии.
EXECUTE FUNCTION	Разрешено если мандатная метка видимой функции меньше или равна мандатной метке сессии.

РЕГИСТРАЦИЯ СОБЫТИЙ

Регистрация событий — реализована с помощью модуля pg_audit, предоставляет возможность детализированного логгирования различных событий. Есть возможность гибко настраивать, какие типы событий регистрировать и на какие объекты необходимо регистрировать эти события:

- Идентификации и аунтификации
- Запросы на доступ к объектам защиты
- Создание/уничтожение объектов защиты
- Действия по изменению ПРД
- Регистрация действий администраторов (их нельзя отключить)
- Механизм интерграции с syslog

Реализация: `/contrib/pgaudit`

Очистка памяти

Принцип работы:
В процессе функционирования СУБД «СИНЕРГИЯ-БД» возвращает операционной системе ранее выделенную память. Это касается как внешней памяти (удаление файлов данных, журналов упреждающей записи), так и оперативной памяти.

Во всех случаях, прежде чем вернуть память операционной системе, СУБД «СИНЕРГИЯ-БД» выполняет очистку памяти, путем заполнения освобождаемого пространства нулевыми байтами.

Особенности:
При удалении данных из таблиц, СУБД «СИНЕРГИЯ-БД» помечает место, которое занимали удаленные строки, как свободное для дальнейшего использования, например для добавления новых строк в таблицу. Эту задачу выполняет процесс очистки (VACUUM), который не только помечает место как свободное, но и заполняет его нулевыми байтами.

Примечание: Если необходимо физически очистить файл после ALTER TABLE DROP COLUMN, используйте команду VACUUM FULL после удаления столбца.

Очистка памяти

Основные положения очистки памяти:
При выполнении ряда операторов происходит удаление файлов, при котором дисковое пространство возвращается операционной системе:

— DROP TABLE
— DROP TEMPORARY TABLE
— DROP MATERIALIZED VIEW
— DROP INDEX
— TRUNCATE
— DROP DATABASE
— DROP SCHEMA
— VACUUM FULL
— REINDEX
— ALTER TABLE ADD COLUMN (с указанием значения по умолчанию)
— ALTER TABLE ALTER COLUMN TYPE
А также и команд:
— DELETE
— UPDATE

Примечание: Если необходимо физически очистить файл после ALTER TABLE DROP COLUMN, используйте команду VACUUM FULL после удаления столбца.


PROFESSIONAL
Postgres
СПАСИБО ЗА ВНИМАНИЕ!
ВОПРОСЫ?
www.postgrespro.ru