Безопасность и аудит

Основные функции системы безопасности

- **Проверка подлинности (аутентификация)** – процедура проверки соответствия некоего лица и его учетной записи в компьютерной системе.
  - Например: Пользовательский идентификатор и пароль = некоторая конфиденциальная информации, знание которой обеспечивает владение определенным ресурсом.
  - Аутентификацию != Идентификация.

- **Авторизация** – это предоставление лицу прав на какие-то действия в системе.

Дополнительные функции безопасности

- Шифрование
- Контекстное переключение
- Олицетворение
- Встроенные средства управления ключами

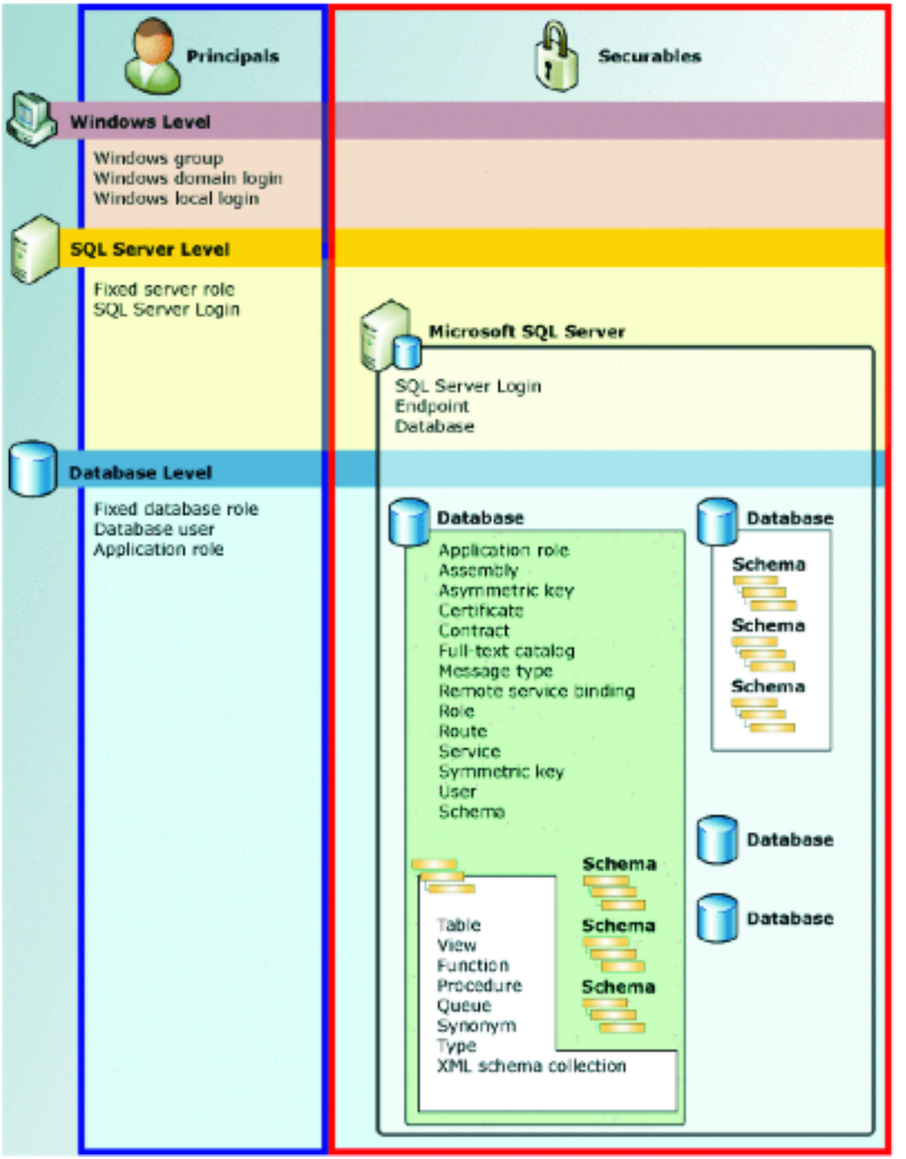
Три ключевых понятия системы безопасности

- Участники системы безопасности (Principals)
- Защищаемые объекты (Securables)
- Система разрешений (Permissions)

Участники системы безопасности

Область влияния принципала зависит:

- от области его определения:
  - Операционная система
  - СУБД
  - База данных
- от количества пользователей области:
  - Индивидуальный (неделимый) участник, например, логин в ОС
  - Коллективный участник - группа пользователей ОС



Защищаемые объекты

Защищаемые объекты – это ресурсы, доступ к которым регулируется системой авторизации.

К областям защищаемых объектов относятся:

- Сервер - управление учетными записями подключения (login)
- База данных - управление:
  - учетными записями пользователей (user)
  - ролями (role)
- Схема - управление разрешениями на объекты БД:
  - функции (function)
  - процедуры (stored procedure)
  - таблицы (table)
  - представления (view)

Пользователи и группы пользователей

Работа с пользователем:

- Создать пользователя
  - CREATE USER Jonathan;
  - CREATE USER davide WITH PASSWORD ‘jw8s0F4’;
  - CREATE USER miriam WITH PASSWORD ‘jw8s0F4' VALID UNTIL ‘2005-01-01’;
  - CREATE USER manuel WITH PASSWORD 'jw8s0F4' CREATEDB;
- Удалить пользователя
  - DROP USER Jonathan

Группы упрощают процедуру назначения прав пользователям. Обычные привилегии должны назначаться каждому пользователю по отдельности.

- Добавить пользователя в группу
  - ALTER GROUP workers DROP USER beth;

Роли

Права можно выдавать для всей группы и у всей группы забирать. Для этого создаётся роль, представляющая группу, а затем членство в этой группе выдаётся ролям индивидуальных пользователей.

Для настройки групповой роли сначала нужно создать саму роль:

- CREATE ROLE ***имя***;
- GRANT групповая_роль TO роль1, ... ;
- REVOKE групповая_роль FROM роль1, ... ;

Обычно групповая роль не имеет атрибута LOGIN, хотя при желании его можно установить.

Членом роли может быть и другая групповая роль. При этом база данных не допускает замыкания членства по кругу. Также не допускается управление членством роли PUBLIC в других ролях.

Члены групповой роли могут использовать её права двумя способами:

- Каждый член группы может явно выполнить SET ROLE, чтобы временно «стать» групповой ролью.
- Роли, имеющие атрибут INHERIT, автоматически используют права всех ролей, членами которых они являются, в том числе и унаследованные этими ролями права.

Роли. Пример

```
CREATE ROLE joe LOGIN INHERIT;
CREATE ROLE admin NOINHERIT;
CREATE ROLE wheel NOINHERIT;
GRANT admin TO joe;
GRANT wheel TO admin;
```

После подключения с ролью joe сеанс базы данных будет использовать права, выданные напрямую joe, и права, выданные роли admin, так как joe «наследует» права admin.

Однако права, выданные wheel, не будут доступны, потому что, хотя joe неявно и является членом wheel, это членство получено через роль admin, которая имеет атрибут NOINHERIT.

После выполнения команды:

```
SET ROLE admin;
```

сеанс будет использовать только права, назначенные admin, а права, назначенные роли joe, не будут доступны.

После выполнения команды:

```
SET ROLE wheel;
```

сеанс будет использовать только права, выданные wheel, а права joe и admin не будут доступны.

Система разрешений

У субъекта системы есть только один путь получения доступа к объектам - иметь назначенные непосредственно или опосредовано разрешения.

- При непосредственном управлении разрешениями они назначаются субъекту явно
- При опосредованном разрешения назначаются через членство в группах, ролях или наследуются от объектов, лежащих выше по цепочке иерархии.

Управление разрешениями производится путем выполнения инструкций языка DCL:

- GRANT (разрешить)
- DENY (запретить)
- REVOKE (отменить)

Модели управления доступом

- **MAC** — мандатная модель управления доступом
- **DAC** — прямое управление доступом
- **RBAC** — ролевая модель управления доступом
- **ABAC** — атрибутивная модель управления доступом

Мандатная модель управления доступом

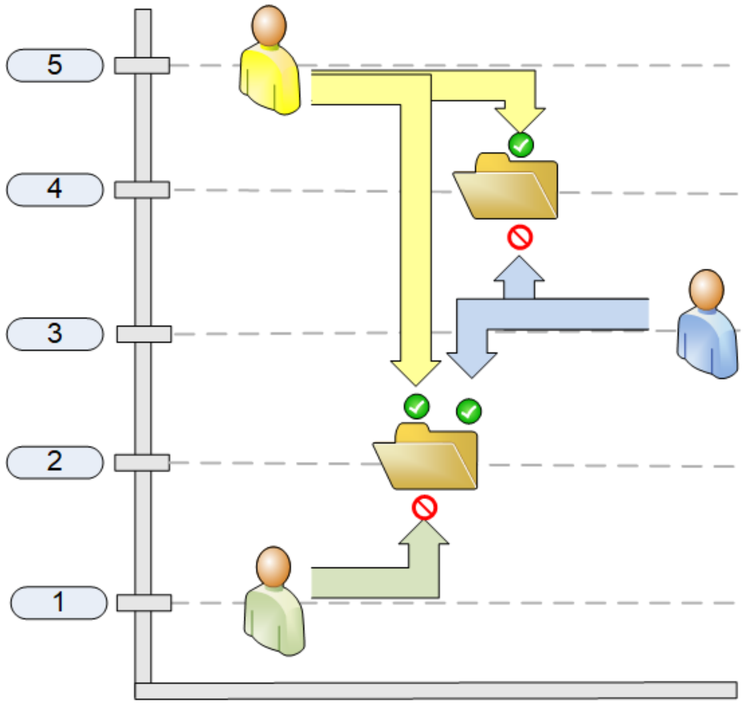
Доступ субъекта к объекту определяется его уровнем доступа: уровень доступа субъекта должен быть не ниже уровня секретности объекта.

Достоинства:

- простата построения общей схемы доступа
- простата администрирования

Недостатки:

- проблема разграничения пользователей одного уровня
- пользователь не может назначать доступ к объекту



Прямое управление доступом

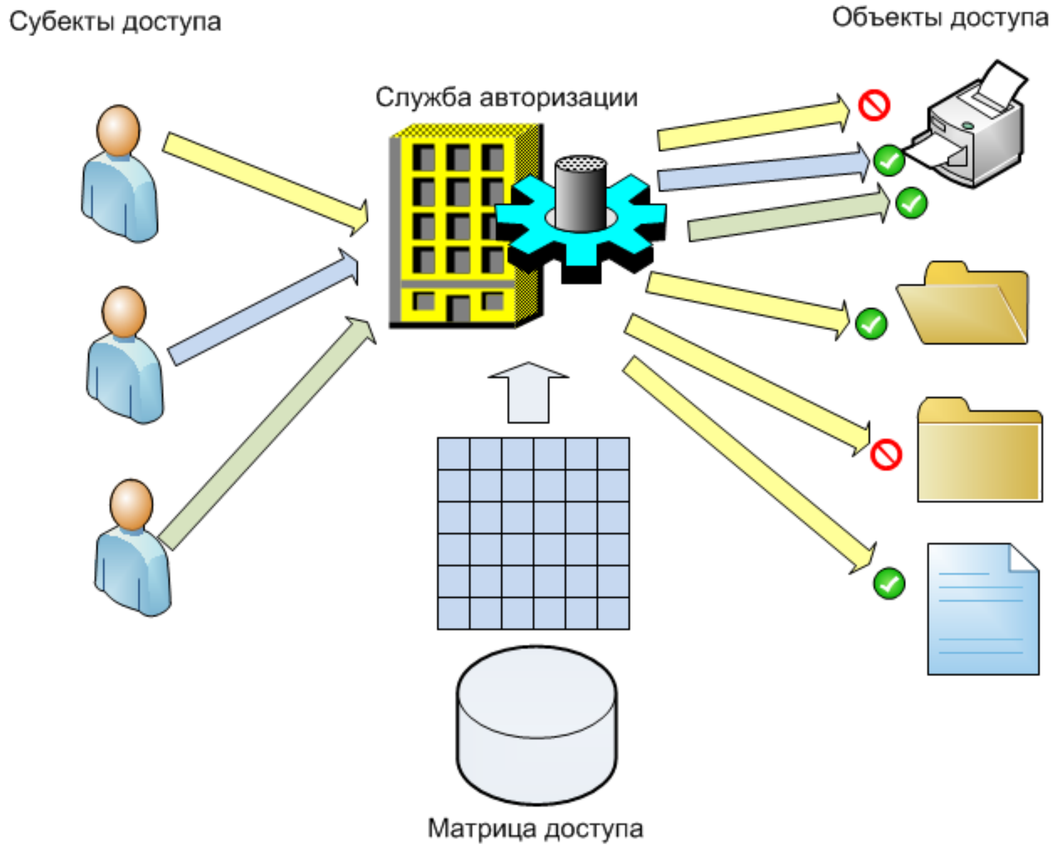
Доступ субъекта к объекту определяется наличием субъекта в списке доступа **(ACL)** объекта.

Достоинства:

- простата реализации
- гибкость (пользователь может описать доступ к своим ресурсам)

Недостатки:

- излишняя детализированность (приводит к запутанности)
- сложность администрирования
- пользователь может допустить ошибку при назначении прав

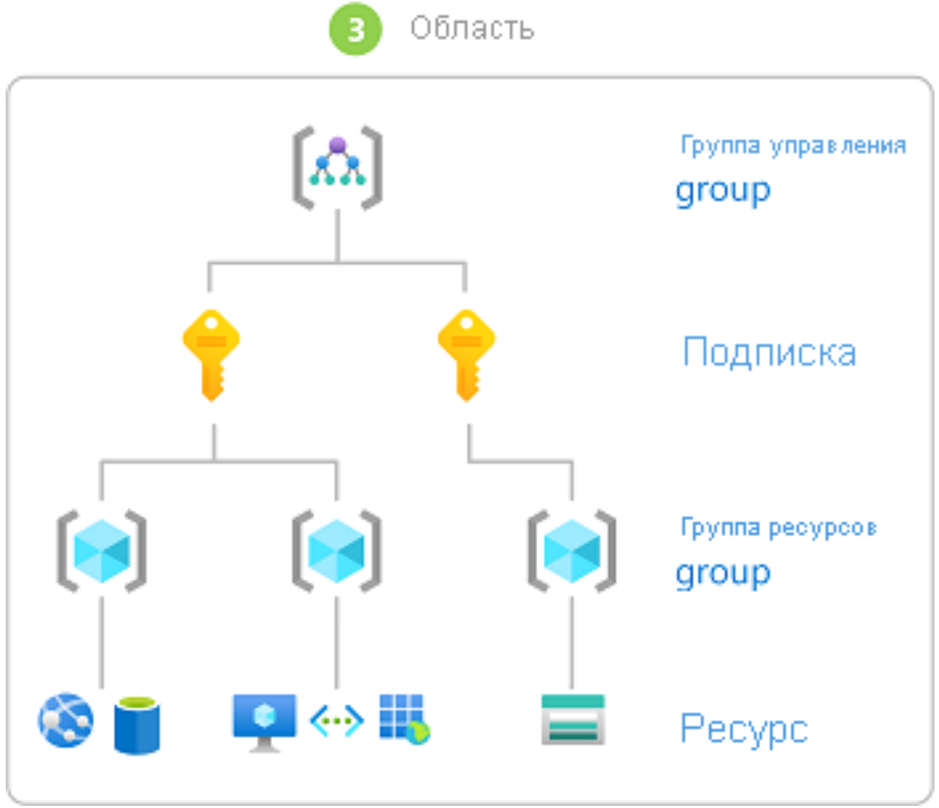


Ролевая модель управления доступом

Доступ субъекта к объекту определяется наличием у субъекта роли, содержащей полномочия, соответствующие запрашиваемому доступу.

Формирование ролей:

- Ролевое разграничение доступа позволяет реализовать гибкие, изменяющиеся динамически в процессе функционирования компьютерной системы правила
- Как правило, данный подход применяется в системах защиты СУБД
- Ролевой подход часто используется в системах, для пользователей которых чётко определён круг их должностных полномочий и обязанностей



Атрибутивная модель управления доступом

Доступ субъекта к объекту определяется динамически на основании анализа политик учитывающих значения атрибутов субъекта, объекта и окружения.

Политика на основе атрибутов:

- Делает управление доступом более эффективным, уменьшая сложность нормативных требований.
- Одна и та же политика может использоваться в разных системах. Это помогает управлять согласованностью доступа к ресурсам в пределах одной компании
- Централизованное управление доступом предполагает единственный авторитетный источник для правил доступа