A01:2021 – Broken Access Control

Алгоритм проверки

1. Сбор информации

- Определить роли в системе (guest, user, admin, API client).
- Зафиксировать все эндпоинты (через Burp/ZAP или документацию API).
- Проверить, какие ресурсы доступны без авторизации.

2. Проверка 'горизонтального повышения привилегий' (Один пользователь получает доступ к данным другого)

- Изменить `id`, `user_id`, `account`, `orderId` в URL/теле запроса.
- Удалить или подменить `session cookie / JWT`.
- Проверить доступ к чужим файлам (например, `/download/12345.pdf` → `/download/12346.pdf`).
- Использовать Burp Plugin **Autorize** – автоматически тестирует доступ к ресурсам других пользователей.

3. Проверка 'вертикального повышения привилегий' (Обычный пользователь получает права администратора)

- Попробовать вызвать админские функции через API (например, `/admin/deleteUser`).
- Подменить роль в токене JWT (`"role":"user"` → `"role":"admin"`) и отправить запрос.
- Проверить скрытые кнопки/ссылки в интерфейсе (часто доступны через прямой URL).
- Использовать Burp Plugin **AuthMatrix** – строит матрицу ролей и проверяет, что доступ ограничен корректно.

4. Проверка обхода контроля доступа

- Изменить HTTP-метод (`GET` → `POST`, `PUT` → `DELETE`).
- Удалить параметры авторизации (cookie, токены) и попробовать доступ.
- Использовать кеширующие прокси (nginx/apache misconfig) → проверить, не отдаются ли приватные данные без авторизации.
- Попробовать **forced browsing** (доступ к `/admin/` без ссылки).

5. Проверка для API

- Запросить API-эндпоинты без токена → доступен ли ресурс.

- Подменить токен одного пользователя → получить доступ к данным другого.
- Проверить CORS-настройки (разрешены ли сторонние домены).
- Использовать Postman/Insomnia для ручных тестов.

6. Проверка хранения сессий и токенов

- Проверить, что токен **привязан к пользователю и роли**.
- Проверить истечение токенов (можно ли использовать старые).
- Убедиться, что при смене пароля/выходе из аккаунта токены инвалидируются.

7. Фиксация результатов

Для каждой проверки:

* **URL / endpoint**
* **Метод и параметры**
* **Ожидаемое поведение** (например, отказ в доступе)
* **Фактическое поведение** (например, доступ разрешен)
* **Уровень риска** (High/Medium/Low)
* **Рекомендации** (RBAC, проверка ACL на сервере, строгая валидация ID и токенов)

Практическая реализация

Вот предложенный алгоритм проверки. Однако за этим (на самом деле стартовым) алгоритмом скрываются следующие вопросы:

- способ хранения и редактирования данных о сервере
- объем требуемых данных
- стэк используемых технологий
- глубина поиска и формат отчета

Разделим на black box и white box.

Например есть некий сервер. Начнем с директории с файлами и посмотрим, насколько это будет удобно.

1. Сбор информации. Определить роли в системе (guest, user, admin, API client).

Revision #2
Created 6 September 2025 12:14:31 by Admin
Updated 7 September 2025 14:32:46 by Admin