

# Теперь о стандартном - безопасность на уровне приложений (бэкенд\фронтенд)

## 1. Валидация и санитизация входных данных

Угрозы:

- SQL-инъекции
- XSS-атаки
- Command injection

Реализация и нюансы:

- Применение библиотек и фреймворков для автоматической валидации входных данных.
- Никогда не доверять входным данным; всегда проверять и очищать их.
- Использование параметризованных запросов для SQL.

## 2. Аутентификация и авторизация

Угрозы:

- Неавторизованный доступ
- Угон сессии
- Внутренние угрозы

Реализация и нюансы:

- Использование стандартных решений и протоколов (OAuth, JWT, SAML).
- Многофакторная аутентификация.
- Принцип наименьших привилегий.

### 3. Управление сессиями

Угрозы:

- Угон сессии
- Перехват данных

Реализация и нюансы:

- Использование уникальных идентификаторов сессии.
- Хранение сессий в безопасном хранилище (не в базе данных, а в специальных инструментах типа Keycloak)
- Ограничение времени жизни сессии.

### 4. Шифрование данных

Угрозы:

- Перехват данных
- Раскрытие конфиденциальной информации

Реализация и нюансы:

- Использование стандартных алгоритмов шифрования (AES, RSA).
- Шифрование данных "в покое" и "в передаче" (TLS/SSL для транспортного уровня).

### 5. API Security

Угрозы:

- Доступ к конфиденциальным данным
- Инъекции кода

Реализация и нюансы:

- Использование API Gateway для управления доступом.
- Валидация типов ожидаемых данных, размера, формата.

**6. Журналирование и мониторинг**

Угрозы:

- Необнаруженные угрозы и атаки
- Отсутствие аудита и прозрачности

Реализация и нюансы:

- Логирование всех подозрительных действий и ошибок.
- Настройка систем мониторинга и алертов (например, ELK Stack, Grafana).

**7. Проверка кода на уязвимости**

Угрозы:

- Уязвимости в коде, приводящие к различным типам атак

Реализация и нюансы:

- Использование автоматических сканеров уязвимостей кода (например, OWASP ZAP, SonarQube).
- Регулярные код-ревью на предмет безопасности.

**8. Обработка ошибок**

Угрозы:

- Раскрытие слишком много информации через сообщения об ошибках

Реализация и нюансы:

- Использование пользовательских страниц ошибок.
- Логирование подробной информации для внутреннего использования, но не для публичного отображения.

**10. Rate Limiting для входящих сетевых запросов**

Угрозы:

- Деградация производительности
- DoS/DDoS

Реализация и нюансы:

- Ограничение числа запросов от одного пользователя или IP-адреса за определенный период времени.