

# РБПО – Отчёт ДЗ 1-5

Выполнил: Васюков Александр, БПИ235  
Проект: "Media Catalog"

## 1. Нефункциональные требования безопасности (NFR)

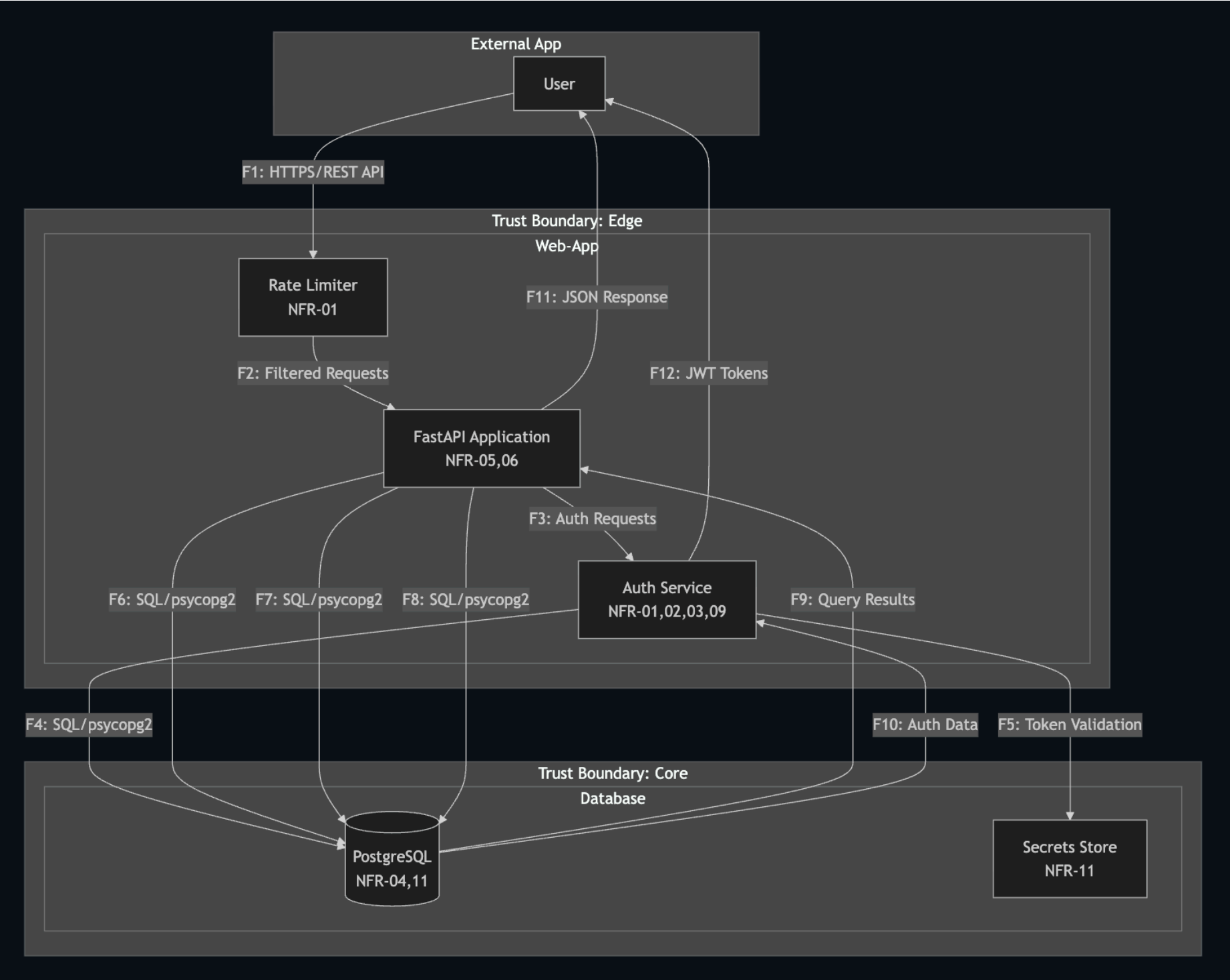
Было сформулировано 11 измеримых требований безопасности с чёткими критериями приемки:

ID	Название	Описание	Метрика/Порог	Проверка (чем/где)	Компонент	Приоритет
NFR-01	Защита от brute-force атак	Ограничение попыток входа/регистрации	$\leq 5$ попыток/мин с IP, блокировка на 15 мин	WAF/Rate limiting middleware	auth	High
NFR-02	Безопасность паролей	Хэширование паролей современными алгоритмами	Argon2id: t=3, m=64MB, p=2	Unit-тесты + конфиг хэширования	auth	High
NFR-03	Время жизни JWT токенов	Ограничение времени действия токенов	Access: 15 мин, Refresh: 7 дней	Конфиг JWT + тесты	auth	High
NFR-04	Защита данных в БД	Шифрование чувствительных данных	PII в зашифрованном виде	Миграции БД + тесты	database	High
NFR-05	Безопасность API эндпоинтов	Валидация и санитизация входных данных	100% эндпоинтов с валидацией	Pydantic схемы + тесты	api	High
NFR-06	Производительность под нагрузкой	Отклик API при пиковой нагрузке	$p95 \leq 500ms$ при 30 RPS	Нагрузочные тесты k6	api	Medium
NFR-07	Уязвимости зависимостей	Контроль уязвимостей в зависимостях	Critical/High $\leq 7$ дней	CI: SCA сканирование	build	High
NFR-08	Безопасность контейнеров	Запуск без root прав	runAsNonRoot: true	Dockerfile + securityContext	deployment	Medium
NFR-09	Логирование безопасности	Аудит критических событий	100% auth событий логируются	Structured logging + ELK	monitoring	Medium
NFR-10	HTTPS enforcement	Принудительное шифрование трафика	100% redirect HTTP→HTTPS	Reverse proxy config	network	High
NFR-11	Безопасная ротация секретов	Секреты (.env, токены) обновляются регулярно	каждые $\leq 30$ дней	Vault/CI политика + журнал	ci/cd	Low

**Комментарий:** Требования покрывают все ключевые аспекты безопасности приложения - от аутентификации до инфраструктуры. Особенно важно, что все требования измеримы, что позволяет объективно оценивать их выполнение.

## 2. Диаграммы потоков данных (DFD) с границами доверия

Было разработано 3 уровня DFD. Контекстная диаграмма показывает систему с двумя границами доверия:



Границы доверия:

- **Trust Boundary: Edge** - веб-приложение (FastAPI + аутентификация)
- **Trust Boundary: Core** - база данных и хранилище секретов

Ключевые потоки данных:

- **F1:** Пользователь → API (HTTPS/REST) - основной вектор атак
- **F2-F5:** API → БД (SQL/psycopg2) - защита от инъекций
- **F6-F8:** БД → API - защита конфиденциальных данных
- **F12:** Аутентификация → Пользователь (JWT) - безопасная передача токенов

**Комментарий:** Архитектура спроектирована с учётом принципа минимальных привилегий - каждая компонента имеет только необходимые доступы.

3. Анализ угроз STRIDE

Проанализировано 14 угроз по методике STRIDE:

Поток/Элемент	Угроза (STRIDE)	Контроль	Ссылка на NFR	Обоснование/Проверка
F1 (User→API)	<b>Spoofing:</b> Подмена пользователя	JWT токены с TTL, валидация	NFR-03	Access: 15 мин, Refresh: 7 дней
F1 (User→API)	<b>Tampering:</b> Изменение данных в транзите	HTTPS enforcement	NFR-10	100% redirect HTTP→HTTPS
F1 (User→API)	<b>Repudiation:</b> Отказ от операций	Логирование auth событий	NFR-09	100% auth событий логируются
F1 (User→API)	<b>Information Disclosure:</b> Перехват данных	Шифрование TLS, валидация данных	NFR-05, NFR-10	Pydantic схемы + HTTPS
F1 (User→API)	<b>DoS:</b> Отказ в обслуживании	Rate limiting middleware	NFR-01, NFR-06	≤5 попыток/мин, p95 ≤500ms при 30 RPS
F1 (User→API)	<b>Elevation of Privilege:</b> Неавторизованный доступ	Валидация входных данных	NFR-05	100% эндпоинтов с валидацией

Поток/Элемент	Угроза (STRIDE)	Контроль	Ссылка на NFR	Обоснование/Проверка
F2-F5 (API→DB)	Tampering: SQL Injection	Параметризованные запросы SQLAlchemy	NFR-05	Pydantic валидация + ORM
F2-F5 (API→DB)	Information Disclosure: Утечка PII данных	Шифрование чувствительных данных	NFR-04	PII в зашифрованном виде в БД
F6 (DB→API)	Information Disclosure: Чтение чужих данных	Проверка прав доступа	NFR-05	Валидация ownership данных
API (Application)	Tampering: Уязвимости в зависимостях	SCA сканирование в CI/CD	NFR-07	Critical/High ≤7 дней для исправления
API (Application)	DoS: Resource exhaustion	Лимиты запросов, мониторинг	NFR-01, NFR-06	Rate limiting + нагрузочное тестирование
DB (Database)	Spoofing: Неавторизованный доступ	Защита учетных данных БД	NFR-11	Ротация секретов каждые ≤30 дней
DB (Database)	Information Disclosure: Кража данных БД	Шифрование PII, безопасные бэкапы	NFR-04	Argon2id для паролей, шифрование PII
Container (Runtime)	Elevation of Privilege: Привилегии контейнера	Запуск без root прав	NFR-08	runAsNonRoot: true в Dockerfile

**Комментарий:** Анализ покрывает все категории STRIDE и показывает системный подход к безопасности - от кода до инфраструктуры.

## 4. Меры защиты с трассировкой к угрозам и историям

Реализованные меры напрямую связаны с выявленными угрозами:

Мера (ADR/PR)	Угрозы	NFR	Пользовательские истории	Реализация
ADR-001: RFC 7807 Error Handling	R10 (Недостаточное логирование)	NFR-09	MON-01	Correlation ID + структурированные логи
ADR-002: URL Validation	R2 (SQL Injection), XSS	NFR-05	API-01	Валидация схем, запрет внутренних адресов
ADR-003: Resource Limits	R4 (DoS атака)	NFR-01, NFR-06	PERF-01	Лимиты 1MB, таймауты 30с
Pydantic валидация	R2, R12	NFR-05	API-01	100% эндпоинтов с валидацией
Rate Limiting	R1, R4	NFR-01	AUTH-01	5 запросов/мин на аутентификацию

**Комментарий:** Каждая реализованная мера защиты закрывает конкретные уязвимости и связана с измеримыми требованиями безопасности.

## 5. Приоритизация рисков

RiskID	Описание	Связь (Flow/NFR)	L	I	Risk	Стратегия	Владелец	Срок	Критерий закрытия
R1	Брутфорс аутентификации	F1, NFR-01	3	4	12	Снизить	@backend-dev	2025.10	Rate limiting 5 req/min на /auth (AUTH-01)
R2	SQL Injection через API	F2-F5, NFR-05	2	5	10	Снизить	@backend-dev	2025.10	100% Pydantic валидация (API-01)
R3	Утечка PII данных	F6, NFR-04	2	5	10	Снизить	@backend-dev	2025.11	Шифрование PII в БД (DB-01)
R4	DoS атака на API	F1, NFR-01, NFR-06	3	3	9	Снизить	@devops	2025.12	Rate limiting + тесты 30 RPS (PERF-01)
R5	Компрометация JWT токенов	F1, NFR-03	3	3	9	Снизить	@backend-dev	2025.10	JWT TTL 15мин/7дней (AUTH-03)
R6	Уязвимости зависимостей	API, NFR-07	3	3	9	Снизить	@security	2025.10	SCA scanning в CI (DEV-01)
R7	Небезопасные контейнеры	Container, NFR-08	2	4	8	Снизить	@devops	2025.10	runAsNonRoot: true (DEPLOY-01)
R8	HTTP трафик	F1, NFR-10	2	4	8	Снизить	@devops	2025.10	100% redirect HTTP→HTTPS (INFRA-01)

RiskID	Описание	Связь (Flow/NFR)	L	I	Risk	Стратегия	Владелец	Срок	Критерий закрытия
R9	Просроченные секреты	DB, NFR-11	2	3	6	Снизить	@devops	2025.10	Ротация секретов ≤ 30 дней
R10	Недостаточное логирование	F1, NFR-09	2	3	6	Снизить	@backend-dev	2025.12	100% auth событий логируются (MON-01)
R11	Weak password hashing	DB, NFR-02	1	4	4	Снизить	@backend-dev	2025.10	Argon2id хэширование (AUTH-02)
R12	Невалидные данные медиа	F1, NFR-05	4	1	4	Принять	@backend-dev	2025.10	Pydantic validation покрывает основные кейсы

На основе матрицы рисков (L×I) выделены три уровня приоритета:

### Высокий приоритет (Risk ≥ 10)

- **R1:** Брутфорс аутентификации (12) - реализован rate limiting
- **R2:** SQL Injection (10) - реализована Pydantic валидация
- **R3:** Утечка PII данных (10) - запланировано шифрование

### Средний приоритет (Risk 6-9)

- **R4:** DoS атака (9) - реализованы лимиты запросов
- **R5:** Компрометация JWT (9) - настроены TTL токенов
- **R6:** Уязвимости зависимостей (9) - SCA в CI/CD

### Низкий приоритет (Risk ≤ 5)

- **R11:** Weak password hashing (4) - запланирован Argon2id
- **R12:** Невалидные данные медиа (4) - принят риск

## 6. Реализация DevOps и безопасного кодирования

### Безопасное кодирование

- **Валидация:** Pydantic схемы с кастомными валидаторами (URL, границы данных)
- **Защита от инъекций:** SQLAlchemy ORM, параметризованные запросы
- **Обработка ошибок:** RFC 7807 без раскрытия внутренней информации

### DevOps-процесс

- **Git:** Ветвление через PR с обязательными ревью
- **Docker:** Многостадийные сборки, минимальные образы
- **CI/CD:** GitHub Actions с автотестами и проверками безопасности
- **Pre-commit:** Ruff, Black, Isort для качества кода

### DevSecOps

- **SAST:** Статический анализ в pre-commit хуках
- **Dependency scanning:** Контроль уязвимостей в requirements
- **Тестирование:** Pytest с покрытием модулей, включая security-тесты

## 7. Наблюдаемость и мониторинг

В процессе реализации ключевых механизмом наблюдаемости:

- **Структурированные логи:** Correlation ID для трассировки запросов
- **Мониторинг ошибок:** RFC 7807 с детализацией для отладки
- **Метрики производительности:** Готовность к нагрузочному тестированию

## Заключение

Проделанная работа демонстрирует полноценный цикл внедрения безопасности в процесс разработки:

1. **Подход:** Требования безопасности сформулированы до реализации
2. **Системный анализ:** DFD и STRIDE выявили угрозы на всех уровнях
3. **Измеримость:** Все требования имеют чёткие критерии проверки

- 4. **Трассируемость:** Меры защиты напрямую связаны с конкретными угрозами
- 5. **Реализация:** Меры внедрены в код и инфраструктуру