

1. Введение

1.1 Назначение проекта

Разработка контейнера для обеспечения безопасности рекомендательной системы в финансовом секторе при угрозе неправомерных действий в каналах связи.

Проект реализует рекомендательную систему с защитой от DDoS, SQL-инъекций и XSS-атак через контейнеризацию и логирование в JSON-формате

1.2 Основание для разработки

Задание в рамках курса «Основы DevOps» на реализацию рекомендательной системы с защитой от сетевых угроз.

Цель: создать масштабируемую систему, соответствующую современным стандартам безопасности и DevOps-подходам

2. Назначение и область применения

2.1 Функциональное назначение

Контейнер предназначен для:

- Прокси-сервера (Nginx) с логированием в JSON для анализа трафика.
- Рекомендательного сервиса (Flask) с HTML-интерфейсом.
- Хранения данных в MySQL.
- Блокировки IP через Fail2ban.

2.2 Область применения

Система может использоваться в банковских и финансовых организациях для:

- Персонализированных рекомендаций по продуктам (кредиты, инвестиции, страхование).
- Защиты от несанкционированного доступа к данным.
- Интеграции с WAF (ModSecurity), мониторингом (Prometheus) и визуализацией (Grafana)

3. Технические характеристики

3.1 Архитектура системы

Пользователь

|



[proxy:443 (Nginx)] → [fail2ban: (IP-блокировка)]

|



[recommendation_system:5000 (Flask)] ← [db:3306 (MySQL)]

3.2 Описание компонентов

Nginx - Прокси, логирование в JSON, TLS 1.2+

Flask - Бэкенд рекомендаций, безопасный поиск через ORM

MySQL - Хранение данных рекомендаций, инициализация через

Fail2ban - Планируется: блокировка IP при частых запросах (не настроен)

3.3 Логирование

Формат: JSON

Поля:

- @timestamp: время запроса
- remote_addr: IP-адрес клиента
- request: тип и путь запроса
- status: HTTP-код ответа

3.4 Безопасность

- ORM (SQLAlchemy): защита от SQL-инъекций.
- Fail2ban (не настроен): блокировка IP после 50+ запросов за 60 секунд.
- HTTPS: шифрование данных через TLS 1.2+.
- Docker-сети: изолированная сеть для контейнеров

4. Ожидаемые технико-экономические показатели

4.1 Эффективность

- Быстродействие: обработка до 1000 RPS.
- Безопасность: предотвращение DDoS, SQLi, XSS через контейнеризацию.
- Масштабируемость: добавление Redis, WAF, мониторинга — без изменения основной логики.

4.2 Стоимость

- Аппаратные требования:
 - RAM: 4 GB+
 - CPU: 2 ядра+
 - Диск: 20 GB свободного места
- Программное обеспечение:
 - Docker Engine 20+, Docker Compose 2.0+
 - Linux kernel 5.10+
 - Python 3.11+, MySQL 8.0+

4.3 Ожидаемые выгоды

- Снижение рисков утечки данных через логирование и блокировку подозрительного трафика.
- Упрощение масштабирования за счет контейнеризации.

- Снижение времени на сопровождение через автоматизацию (Docker Compose, JSON-логи).

5. Источники, использованные при разработке

1. ГОСТ 19.404-79 — ЕСПД. Пояснительная записка. Требования к содержанию
2. Docker Documentation — <https://docs.docker.com>
3. Nginx Documentation — <https://nginx.org/en/docs/>
4. Flask Documentation — <https://flask.palletsprojects.com>
5. MySQL Documentation — <https://dev.mysql.com/doc/>
6. Fail2ban Documentation — <https://fail2ban.org>