Документация по обеспечению безопасности системы ISET

---

1. Требования по безопасности, предъявляемые к решению

1.1 Защита секретов

- Конфиденциальные данные (логины, пароли, токены) хранятся в `.env`
- Использование Docker secrets для критичных параметров
- JWT-токены с ограниченным сроком действия и refresh-токенами

1.2 Нетранспортная безопасность

- Все внешние соединения защищены TLS (NGINX Proxy Manager)
- Изолированная Docker-сеть `iset-local`
- Цифровая подпись JWT-токенов

1.3 Механизмы защиты от атак

- Валидация входных данных на API
- Ограничение логин-атак в PGAdmin (`MAX_LOGIN_ATTEMPTS`)
- Redis для хранения токенов с таймаутом
- Регулярное обновление компонентов

1.4 Изоляция компонентов

- Каждый сервис в отдельном контейнере
- Ограничение взаимодействия через Docker network
- Минимальный сетевой доступ к PostgreSQL/Redis

1.5 Состав продукта

- **Собственные сервисы**:
  SPA Frontend, API Main, API Report, API File, Analytics

- **Внешние сервисы**:
  PostgreSQL, MinIO, Redis, NGINX

- **Вспомогательные**:
  pgAdmin, Portainer, Prometheus+Grafana

---

2. Элементы конфигурации, реализующие функции безопасности

2.1 Docker-сети

```yaml
networks:
  iset-local:
    name: iset-local
    driver: bridge
    internal: true
```

2.2 Защита базы данных

- Разделение БД на схемы
- Раздельные пользователи для сервисов
- Резервное копирование:

```
environment:
  - BACKUP_KEEP_DAYS=28
  - BACKUP_KEEP_WEEKS=4
  - BACKUP_KEEP_MONTHS=12
```

2.3 Защита Redis

```
environment:
  - ALLOW_EMPTY_PASSWORD=no
  - REDIS_AOF_ENABLED=no
```

2.4 NGINX Proxy Manager

- HTTPS с Let's Encrypt
- Ограничение доступа:

```
limit_req zone=one burst=10 nodelay;
limit_conn zone=addr burst=5;
```

---

3. Регистрация событий ИБ

3.1 Логирование

- Вывод логов в stdout/stderr
- Централизованное логирование (ELK/Loki)
- Ключевые события: входы, изменения конфигурации, ошибки аутентификации

3.2 Изменения конфигурации

- Регистрация изменений `.env`
- Контроль версий через Git
- Перезапуск сервисов при изменении конфигурации

---

4. Типовые инциденты ИБ и реагирование

Инцидент	Симптомы	Действия по устранению	Обращение к вендору
Утечка токенов	Увеличение неправомерных запросов	Отзыв токенов, изменение схемы авторизации	Для анализа механизма генерации токенов

Инцидент	Симптомы	Действия по устранению	Обращение к вендору
DDoS-атака	Повышенная нагрузка на API	Ограничения в NGINX, временное отключение доступа	Для оптимизации обработки запросов
Уязвимость в зависимостях	Уведомление об уязвимости	Обновление образов, пересборка контейнеров	Для патчинга уязвимостей
Нарушение целостности	Ошибки при запуске сервисов	Проверка контрольных сумм, восстановление из бэкапов	Для проверки механизмов целостности
Компрометация учетной записи	Несанкционированные изменения	Блокировка учетной записи, смена паролей, аудит	Для внедрения MFA
RCE-уязвимость	Выполнение произвольных команд	Изоляция контейнеров, обновление образов	Для срочного патчинга

Процедура обращения к вендору:
1. Сбор доказательств (логи, конфигурации, дампы памяти)
2. Открытие тикета в системе поддержки с пометкой `SECURITY`
3. Предоставление временного решения (workaround)
4. Координация внедрения фикса
5. Ведение журнала обращений (дата, проблема, решение, сроки)

---

5. Необходимые сетевые параметры

Компонент	Протокол	Внутренний порт	Внешний порт	Назначение	Доступ
NGINX Proxy Manager	HTTPS	443	443	Терминация TLS	Интернет
NGINX Proxy Manager	HTTP	80	80	Перенаправление на HTTPS	Интернет
NGINX Proxy Manager	HTTP	81	81	Админ-панель	Доверенные IP

Компонент	Протокол	Внутренний порт	Внешний порт	Назначение	Доступ
Frontend	HTTP	80	8181	Клиентское SPA	Через NGINX
API Main Service	HTTP	80	8282	Бизнес-логика	Внутренняя сеть
API File Service	HTTP	80	8383	Управление файлами	Внутренняя сеть
API Report Service	HTTP	80	8484	Генерация отчетов	Внутренняя сеть
PostgreSQL	TCP	5432	-	База данных	Внутренняя сеть
Redis	TCP	6379	6379	Кеш/брокер сообщений	Внутренняя сеть
MinIO	HTTP	9000	9001	Хранилище файлов	Внутренняя сеть
pgAdmin	HTTP	80	8585	Администрирование БД	Через NGINX
pgBackup	HTTP	8080	-	Мониторинг бэкапов	Внутренняя сеть
Celery	-	-	-	Асинхронные задачи	-

---

6. Механизмы проверки целостности

6.1 Проверка образов

- Использование signed Docker images
- Проверка контрольных сумм
- Сравнение хэшей конфигураций (SHA-256)

6.2 Проверка .env файла

- Хранение в зашифрованном виде
- Шифрование GPG
- Проверка контрольной суммы при старте

---

7. Резервное копирование

7.1 Параметры

- Регулярность: ежедневно/еженедельно/ежемесячно
- Хранение: 28 дней/4 недели/12 месяцев
- Формат: сжатый SQL (`.sql.gz`)

7.2 Объекты копирования

- Базы данных (PostgreSQL)
- Конфигурационные файлы
- Пользовательские файлы (MinIO)
- Зашифрованные секреты (.env)

7.3 Технические меры

- PostgreSQL: `pgbackup`
- MinIO:

```
docker exec -it minio mc mb local/mybackup
docker exec -it minio mc mirror /data local/mybackup
```

7.4 Восстановление

- PostgreSQL:

```
pg_restore --clean -h db -U postgres -d iset /backups/backup.sql.gz
```

- MinIO:

```
docker exec -it minio mc mirror local/mybackup /data
```

---

8. Рекомендации по hardening

8.1 Обновление зависимостей

- Регулярный аудит версий Docker
- Мониторинг уязвимостей (Docker Security Scanning)
- Обновление Angular/Python-библиотек

8.2 Безопасность контейнеров

- Минимальные образы (alpine)
- Отключение root-доступа
- SELinux/AppArmor для ограничения привилегий

8.3 Безопасность приложения

- Content Security Policy (CSP) в Angular
- Строгая настройка CORS
- Регулярный аудит кода

---

9. Дополнительные меры безопасности

9.1 Защита от DDoS

- Настройка NGINX:

```
limit_req zone=one burst=100;
limit_conn zone=addr burst=50;
```

- Использование WAF

9.2 Защита от SQL-инъекций

- Использование ORM
- Параметризованные запросы
- Регулярные проверки уязвимостей

9.3 Защита файлов

- Минимальные права доступа в MinIO
- Уникальные имена файлов
- Ограничение размеров загрузки

---

10. Рекомендации по эксплуатации

1. Регулярное обновление Docker образов
2. Мониторинг (Prometheus + Grafana)
3. Настройка алертов
4. Периодические penetration-тесты
5. Обновление Angular/Python-библиотек
6. Проверка целостности .env
7. Квартальный аудит конфигурации
8. Тестирование процедур восстановления

---

11. Требования к развертыванию

11.1 Управление учетными данными

- **Обязательная смена паролей** после первого запуска:
  - `POSTGRES_PASSWORD`
  - `MINIO_ROOT_PASSWORD`
  - `BACKEND_SECRET_KEY`
  - `JWT_SECRET_KEY`
- Ротация каждые 90 дней
- Требования к сложности: 16+ символов, верхний/нижний регистр, цифры, спецсимволы
- Использование временных токенов для Docker Registry

11.2 TLS-шифрование

- **Клиент-API**:
  - TLS 1.3+ с шифрами `TLS_AES_256_GCM_SHA384`
  - Обязательный HSTS

  - Авто-продление сертификатов Let's Encrypt
- **Межсервисное взаимодействие**:
  - mTLS для аутентификации сервисов

```
ssl_verify_client on;
ssl_client_certificate /etc/nginx/client_certs/ca.crt;
```

  - Генерация сертификатов через HashiCorp Vault
  - Полный запрет HTTP-трафика

11.3 Дополнительные меры

- Сетевая изоляция: VLAN для prod/stage
- Шифрование данных:
  - MinIO: SSE-S3
  - PostgreSQL: pgcrypto
- Firewall: запрет прямого доступа к БД/Redis из интернета

11.4 Настройки фаервола

- Разрешенные порты:
  - 22 (SSH)
  - 80 (HTTP → HTTPS)
  - 443 (HTTPS)
  - 81 (NGINX Admin, только для доверенных IP)
- Блокировка всех остальных портов

11.5 Взаимодействие с вендором

- Назначение ответственного за связь
- Контакты поддержки 24/7 для критичных инцидентов
- SLA на установку обновлений безопасности: 72 часа
- Процедура запроса доработок:
  1. Формирование технического задания
  2. Согласование сроков и стоимости
  3. Тестирование в staging-среде
  4. Внедрение в production

Приложения

Приложение А. Пример .env файла

```
JWT_SECRET_KEY=supersecretkey
POSTGRES_USER=iset_user
POSTGRES_PASSWORD=Chang3Me!2023
MINIO_ROOT_USER=admin
MINIO_ROOT_PASSWORD=Str0ngP@ss!2023
```

Приложение B. Резервное копирование

```
# PostgreSQL
docker exec pgbackup pg_dump -h db -U postgres iset_main | gzip >
backup.sql.gz

# MinIO
docker exec minio mc mirror /data local/backup
```

Приложение C. Восстановление

```
# PostgreSQL
gunzip < backup.sql.gz | docker exec -i db psql -U postgres

# MinIO
docker exec minio mc mirror local/backup /data
```

Приложение D. Контакты вендора

- **Экстренная поддержка**: security@iset-soft.ru (телефон: +7-XXX-XXX-XXXX)
- **Тикет-система**: https://support.iset-soft.ru
- **SLA**: Ответ в течение 1 часа для критичных инцидентов
- **Часы работы**: 24/7 для проблем безопасности

Заключение

Документация соответствует требованиям ГОСТ Р 57580, ISO 27001 и содержит: 1. Полный состав системы с портами сервисов 2. Процедуры обработки инцидентов ИБ с обращением к вендору 3. Требования к развертыванию и управлению секретами 4. Настройки TLS для всех типов взаимодействия 5. Механизмы резервного копирования и восстановления 6. Процедуры взаимодействия с вендором

Все рекомендации должны быть реализованы перед промышленной эксплуатацией. Обязательно: - Тестирование TLS/mTLS - Проверка процедур восстановления - Тренировка реагирования на инциденты - Подписание SLA с вендором