УДК 004.056:004.415.2

Перминов Никита Ильич, студент, РТУ МИРЭА – Российский Технологический Университет, Россия, г. Москва

Романькова Екатерина Сергеевна, студент, РТУ МИРЭА – Российский Технологический Университет, Россия, г. Москва

БЕЗОПАСНОСТЬ ИНТЕРФЕЙСОВ ПРОГРАММНОГО ПРИЛОЖЕНИЯ

Аннотация

В статье рассматривается безопасность интерфейсов программирования приложений (API) как критический элемент современной цифровой инфраструктуры, уделяя особое внимание российскому контексту. Анализируются основные угрозы и уязвимости API, включая проблемы аутентификации, авторизации, инъекций и утечек данных, а также системные недостатки в управлении жизненным циклом API. Рассматриваются нормативная база и стандарты, такие как ГОСТ Р ИСО/МЭК 27001 и отраслевые рекомендации, включая практики Open Banking и FAPI. Представлены технические и организационные меры защиты, такие как контракт-ориентированный подход, многослойная аутентификация, валидация данных, rate limiting, аудит и мониторинг. На примерах инцидентов в российских компаниях (например, «Билайн», «Спортмастер») выявляются типичные ошибки и предлагаются пути их устранения. Статья предлагает методику внедрения системной защиты API, учитывающую российские регуляторные требования, и подчеркивает важность управления жизненным циклом API, обучения специалистов и интеграции мер безопасности в процессы разработки и эксплуатации.

Abstract

The article examines the security of application programming interfaces (APIs) as a critical element of modern digital infrastructure, with a particular focus on the

Russian context. It analyzes the main threats and vulnerabilities of APIs, including issues with authentication, authorization, injection attacks, and data leaks, as well as systemic shortcomings in API lifecycle management. The regulatory framework and standards, such as GOST R ISO/IEC 27001 and industry recommendations, including Open Banking and FAPI practices, are considered. Technical and organizational protection measures are presented, such as a contract-oriented approach, multi-layered authentication, data validation, rate limiting, auditing, and monitoring. Using examples of incidents in Russian companies (e.g., Beeline, Sportmaster), typical errors are identified and solutions for their elimination are proposed. The article offers a methodology for implementing systematic API protection that considers Russian regulatory requirements and emphasizes the importance of API lifecycle management, training of specialists, and integrating security measures into development and operational processes.

Ключевые слова: безопасность API, угрозы API, контракт-ориентированный подход, аутентификация, авторизация, защита персональных данных, российские практики безопасности.

Keywords: API security, API threats, contract-oriented approach, authentication, authorization, personal data protection, Russian security practices.

Безопасность интерфейсов программирования приложений (API) выступает критическим элементом современной цифровой инфраструктуры, поскольку API обеспечивают обмен данными и функциональностью между распределёнными компонентами информационных систем, мобильными и веб-клиентами, а также сторонними сервисами и партнёрами; при этом рост количества и сложности API, их применение в ключевых отраслях (финансы, телекоммуникации, ритейл) делает API привлекательной целью для злоумышленников и повышает потенциальный ущерб от компрометации, включая утечки персональных данных и нарушение доступности сервисов [1]. Анализ отечественной практики и международных руководств показывает, что уязвимости в API часто связаны не только с недостатками реализации протоколов, но и с организационной непросмотренностью жизненного цикла

API: отсутствием централизованного управления, несогласованностью спецификаций, недостаточной аудируемостью и слабой привязкой политик доступа к модели угроз организации.

Стандартный массив угроз для API включает механизмы перечисления и злоупотребления методами аутентификации и авторизации, эксплойты, направленные на обход валидации входных данных и инъекции, атаки на бизнес-логику, манипулирование параметрами и ресурсами, перехват и повторное использование токенов, а также утечки через некорректную конфигурацию API-каталогов и схем авторизации. OWASP в своём проекте API Security выдвигает ранжированный набор рисков, характерный как для западных, так и для российских реалий, включая «Broken Object Level Authorization», «Excessive Data Exposure», «Lack of Resources & Rate Limiting» и пр.; перевод и комментарии российских специалистов показывают, что большинство критических рисков остаются релевантными для локальных систем и требуют системного подхода к проектированию и эксплуатации API [2].

В части нормативной базы и стандартов, релевантных практикам безопасности API в Российской Федерации, следует различать два уровня: общие стандарты системного менеджмента информационной безопасности и требования регуляторов, задающие основу риско-ориентированного подхода (например, ГОСТ Р ИСО/МЭК 27001—2021, реализующий принципы ISO/IEC 27001), и отраслевые и технические руководства, практики и открытые проекты, а также специализированные профили и требования для финансового сектора (лучшие практики Open Banking и FAPI для защиты финансовых API). ГОСТ/ISO задают структуру СУИБ и требования к управлению рисками, что делает возможным интегрировать специфические механизмы защиты API в общую политику безопасности организации; одновременно практические рекомендации OWASP и документы вендоров помогают конкретизировать меры на уровне разработки и эксплуатации. Российские обзоры внедрения ISO/IEC 27001 и материалы по профилю защиты мобильных приложений

указывают на необходимость адаптации процессов управления доступом и контроля конфигураций в контексте API-ориентированной архитектуры [3].

Технические и организационные практики, зарекомендовавшие себя как лучшие подходы к обеспечению безопасности API, формируются вокруг нескольких основных принципов: минимизация признаков и поверхности атаки (principle of least privilege, минимизация экспонируемых полей в ответах API), строгая спецификация и контракт-ориентированность (OpenAPI/Swagger как основа для автоматической валидации и тестирования), многослойная аутентификация и авторизация (использование протоколов OAuth 2.0, OpenID Connect, JWT с контролем срока жизни и отзыва токенов; в критичных сценариях — mTLS), валидация и санитация входных данных на основе контрактов, защита от перегрузки и DDoS (rate limiting, quota management), подробный аудит и трассировка (логирование запросов/ответов в сочетании с маскированием конфиденциальных полей), автоматизированное тестирование и сканирование (SAST/DAST/IAST) и непрерывный мониторинг инцидентов и аномалий в поведении API [4]. На уровне стандартов и регуляции эти меры соотносятся с требованиями СУИБ по управлению доступом, журналированию и контролю изменений, что даёт основу для включения API-контролей в процессы соответствия и аудита [1].

Более конкретно, контракт-ориентированный подход предполагает формализацию интерфейсов в машинно-читаемой спецификации (OpenAPI), использование схем валидации (JSON Schema), генерацию стендов тестирования и политики безопасности на основе спецификации, а также внедрение API-фаерволов (API Firewall) или WAF-решений с поддержкой проверки соответствия трафика спецификации. В русскоязычных практических материалах подчёркивается, что наличие формального описания API позволяет не только ускорить разработку и тестирование, но и повысить точность детектирования аномалий и автоматически применять контекстно осознанные правила фильтрации трафика [4].

Управление идентификацией и доступом требует сочетания политик аутентификации, авторизации на уровне ресурсов (RBAC/ABAC), контроля жизненного цикла секретов и ключей, а также механизмов ре-выпуска и отзыва полномочий. Для финансовых и критичных к безопасности приложений рекомендуется применять FAPI (Financial-grade API) и высокозащищённые схемы обмена, а также многофакторную и непрерывную аутентификацию при взаимодействии с клиентами и третьими сторонами; в отечественной повестке вопросы open banking и регионального внедрения API сопровождаются дискуссиями о стандартах безопасности и роли регулятора в формировании требований к API-авторизации [5]. Публикации по открытым банковским API в российском контексте акцентируют необходимость применения цифровой подписи, строгой аутентификации и мониторинга соединений между банком и партнёрами.

Аудит и тестирование API должны быть встроены в цикл разработки: обязательные этапы включают модульное тестирование контрактов, статический анализ кода (SAST), динамическое тестирование (DAST) с имитацией атак, тестирование бизнес-логики, проверку на разграничение доступа и CI/CD-интеграцию тестов безопасности. Научные публикации и отчёты российских исследователей подчёркивают, что только комплексное применение инструментов автоматического тестирования и ручной проверки даёт приемлемый уровень обнаружения как традиционных, так и тонких логических ошибок в API. Дополнительно рекомендуется применять процесс управления уязвимостями с приоритизацией на основе риска и влияния на бизнес [6].

Контроль и мониторинг должны включать не только сбор логов и метрик, но и аналитические механизмы обнаружения аномалий, интеграцию с SIEM и SOAR-системами и использование моделей обнаружения атипичного поведения клиентов и сервисов (например, резкие изменения паттернов вызовов, нехарактерные параметры запросов) [7]. В российских аналитических обзорах отмечается возрастающая роль аналитических

платформ и искусственного интеллекта для обнаружения сложных атак, при этом авторы подчёркивают, что для эффективного обнаружения требуется корректная подготовка данных (маскирование, нормализация), а также акторно-ориентированная классификация инцидентов.

Рассмотрение практических кейсов российских компаний позволяет иллюстрировать характерные ошибки и пути их устранения. Пример распространённой инцидентной практики демонстрирует утечки персональных данных через некорректно защищённые внутренние справочники и API у телекоммуникационных операторов; ряд сообщений в российской прессе и отраслевых изданиях указывает на инциденты в работе «ВымпелКом/Билайн», где информация из корпоративных справочников и частично клиентская база оказались доступными в открытом доступе, что выдвигает на первый план вопросы разграничения прав доступа, аудита внутренних интерфейсов и контроля экспорта данных через API [8]. Анализ таких инцидентов показывает, что причиной нередко служит сочетание избыточных прав доступа, отсутствия MFA для внутренних сервисов и неполного логирования операций с конфиденциальными реестрами [9].

Другой реальный кейс — массовая утечка клиентских данных у ритейлера (например, случай «Спортмастер»), когда база пользователей оказалась в свободном доступе; детальный разбор подобных инцидентов указывает на то, что первопричины часто лежат в недостаточном контроле API-эндпоинтов, неправильной маскировке данных в логах и отсутствии сегментации среды разработки и продакшена [10]. В каждом конкретном случае корневыми задачами для предотвращения повторения выступают внедрение контроля доступа на уровне объектов обязательное шифрование в покое и в движении, а также регулярные упражнения по реагированию на инциденты с прогоном сценариев вытекания данных через API.

Для финансового сектора и банковской экосистемы, где API становятся базовым элементом интеграции, российские исследования и публикации подчёркивают необходимость аккуратного внедрения профилей безопасности,

включая FAPI, строгие правила моделирования угроз и более жёсткий контроль третьих сторон. В отечественных аналитических работах по open API для банков подчёркивается, что регуляторный ландшафт и требования к обработке персональных данных требуют от банков не только технических мер, но и архитектурной перестройки — выделения доверенных зон, внедрения унифицированной модели идентификации клиентов и жёсткой политики данных [11].

Реализация технических мер защиты API в российских проектах должна учитывать локальные регуляторные требования по защите персональных данных (ФЗ-152) и соответствие требованиям Роскомнадзора по журналированию и уведомлению о нарушениях; в ряде случаев целесообразно применять дополнительные меры, такие как шифрование полей на уровне приложения, токенизация персональных идентификаторов и применение адаптивной аутентификации для повышения устойчивости к компрометации учётных данных. Практические руководства по внедрению ISO/IEC 27001 в российских компаниях демонстрируют, как связать требования СУИБ с техническими политиками для API — включая классификацию ресурсов, оценку рисков и создание регламента управления доступом [3].

Ниже представлена синтетическая методика этапов внедрения системной защиты API, сбалансированная с учётом российской специфики: на этапе проектирования — обязательное использование формальных спецификаций (OpenAPI), threat modeling и ограничение данных в ответах; на этапе разработки — применение SAST и контрактных тестов, внедрение политики секрета и управление ключами; на этапе развёртывания — конфигурация API-Gateway с политиками аутентификации/авторизации, rate limiting, WAF/API Firewall и TLS с актуальными наборами шифров; на этапе эксплуатации — мониторинг, SIEM-интеграция, управление уязвимостями и процесс реагирования на инциденты; на этапе аудита — регулярные внешние и внутренние pentest-аудиты, проверка исполнения ГОСТ/ISO и документирование соответствия требованиям регуляторов. Данная методика

согласуется с рекомендациями OWASP и отечественными обзорами практик безопасности API [2].

Особое внимание следует уделять управлению жизненным циклом API в организации: инвентаризация доступных интерфейсов, версияция, депрекация старых версий, унификация политик безопасности и централизованный каталог с механизмом обнаружения несанкционированных эндпоинтов. В российских публикациях по управлению API подчёркивается, что отсутствие таковой инвентаризации создаёт «мёртвые зоны» без мониторинга, где уязвимости остаются незамеченными долгие месяцы и используются злоумышленниками для получения доступа к системам. Поэтому реализация API Management платформы с детальной телеметрией и политиками безопасности является ключевым организационным инструментом.

Рассмотрение экономико-организационных аспектов показывает, что киберинциденты, связанные с API, приводят не только к прямому ущербу (штрафы по закону о персональных данных, расходы на расследование и восстановление), но и к репутационным издержкам, утрате клиентов и дополнительным издержкам на усиление процессов контроля; в контексте российских реалий регуляторные воздействия и рост внимания со стороны Роскомнадзора и отраслевых аудиторских организаций усиливают требования к прозрачности и быстрому реагированию [10]. Аналитические обзоры по утечкам в РФ фиксируют множественные прецеденты и рекомендуют страхование киберрисков как часть комплексной стратегии управления рисками.

Синтез рекомендаций для практического внедрения защиты API в российских организациях следует сформулировать как набор обязательных и рекомендуемых мер: обязательными следует считать наличие формализованной политики API (инвентаризация, спецификация, версияция), централизованного управления ключами и секретами (HSM/Secrets Manager), применение TLS и политики минимизации экспонируемых данных, внедрение RBAC/ABAC и механизмов отзыва полномочий, обязательную проверку

соответствия спецификациям на уровне Gateway и интеграцию логирования с SIEM; рекомендованными являются применение mTLS для межсерверных вызовов, использование FAPI в финансовых интеграциях, расширенное поведенческое обнаружение аномалий и периодические внешние pentest-аудиты. Эти положения подтверждаются как международными рекомендациями OWASP и вендоров облачных платформ, так и отечественными публикациями по управлению и тестированию API [12].

Важным организационным аспектом является обучение разработчиков и тестировщиков специфике угроз API: включение в образовательные программы практических кейсов, регулярные code review с фокусом на безопасность, практики threat modeling и red-team упражнения. Российские исследования фиксируют системную проблему нехватки компетенций по безопасной разработке среди инженерных команд, что делает обучение и регулярные упражнения неотъемлемой частью стратегии [6]. Также значимым является включение безопасности API в SLA и контракты с подрядчиками и облачными провайдерами, чтобы обеспечить соответствующие гарантии уровня защиты.

Наконец, институциональная интеграция — обеспечение соответствия требованиям ГОСТ/ISO, взаимодействие с регуляторами и применение отраслевых профилей безопасности — позволяет организациям не только снизить вероятность инцидентов, но и оперативно реагировать и минимизировать ущерб в случае компрометации. Практика сертификации по ISO/IEC 27001 в российских компаниях показывает, что стандартизация процессов информационной безопасности создаёт предпосылки для системного управления API-рисками, если в рамках СУИБ выделены ясные процессы инвентаризации, управления доступом и реагирования на инциденты [3].

В заключение, безопасность API должна рассматриваться как многослойная дисциплина, объединяющая стандарты управления информационной безопасностью, технические протоколы и операционные

процессы. Применение контракт-ориентированного подхода, строгих схем аутентификации/авторизации, интегрированного тестирования безопасности и непрерывного мониторинга существенно снижает риск утечек и злоупотреблений. Российская практика подтверждает универсальность этих подходов, однако подчёркивает необходимость учёта локальных регуляторных требований и адаптации процессов под специфические особенности отраслей (телеком, банки, ритейл). Анализ инцидентов в российских компаниях указывает на типичные ошибки (избыточные привилегии, недостаточное маскирование данных, отсутствие контроля внутренних API), которые могут быть устранены посредством внедрения перечисленных стандартов и практик.

Список литературы

1. Садовая Е.Н. Современные вызовы и угрозы информационной безопасности REST API и способы их предотвращения // Молодой исследователь Дона. 2023. №3(42). С. 3 – 8. [Электронный ресурс] URL: https://cyberleninka.ru/article/n/sovremennye-vyzovy-i-ugrozy-informatsionnoy-bezopasnosti-rest-api-i-sposoby-ih-predotvrascheniya? (дата обращения: 08.12.2025)
2. OWASP (Open Web Application Security Project) [Электронный ресурс] URL: https://habr.com/ru/companies/owasp/articles/547174/ (дата обращения: 08.12.2025)
3. Национальный стандарт российской федерации [Электронный ресурс] URL: https://docs.cntd.ru/document/1200181890 (дата обращения: 09.12.2025)
4. Горбунов Т. Методология построения и защиты API простыми словами [Электронный ресурс] URL: https://www.itsec.ru/articles/metodologiya-postroeniya-i-zashchity-api-prostymi-slovami? (дата обращения: 10.12.2025)
5. Открытый банкинг в России [Электронный ресурс] URL: https://habr.com/ru/articles/668652/ (дата обращения: 11.12.2025)

6. Медаев Марк Казбекович Тестирование API // E-Scio. 2023. №4 (79). С. 8 – 12. [Электронный ресурс] URL: https://cyberleninka.ru/article/n/testirovanie-api? (дата обращения: 11.12.2025)

7. 2025: Тренды в безопасности API - от роста числа утечек до влияния ИИ [Электронный ресурс] URL: https://www.tadviser.ru/index.php/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F:%D0%A2%D1%80%D0%B5%D0%BD%D0%B4%D1%8B_%D0%B2_%D0%B1%D0%B5%D0%B7%D0%BE%D0%BF%D0%B0%D1%81%D0%BD%D0%BE%D1%81%D1%82%D0%B8_API? (дата обращения: 11.12.2025)

8. «Билайн» допустил крупную утечку. В сети данные сотен тысяч работников оператора [Электронный ресурс] URL: https://www.cnews.ru/news/top/2022-12-02_bilajn_dopustil_krupnuyu? (дата обращения: 11.12.2025)

9. "Билайн" начал расследование утечки данных из корпоративного справочника [Электронный ресурс] URL: https://www.comnews.ru/content/223350/2022-12-02/2022-w48/bilayn-nachal-rassledovanie-utechki-dannykh-korporativnogo-spravochnika? (дата обращения: 11.12.2025)

10. «Спортмастер» признал крупномасштабную утечку данных клиентов [Электронный ресурс] URL: https://www.itsec.ru/news/sportmaster-priznal-krupnomasshtabnutu-utechku-dannih-klientov? (дата обращения: 11.12.2025)

11. Татьяна Николаевна Зверькова Open API в региональных банках: вызовы и направления внедрения // Учет. Анализ. Аудит. 2025. №3, т. 12. С. 4 – 12. [Электронный ресурс] URL: https://cyberleninka.ru/article/n/open-api-v-regionalnyh-bankah-vyzovy-i-napravleniya-vnedreniya? (дата обращения: 12.12.2025)

12. Рекомендации по устранению десятки угроз безопасности API OWASP с помощью управления API [Электронный ресурс] URL:

https://learn.microsoft.com/ru-ru/azure/api-management/mitigate-owasp-api-threats (дата обращения: 13.12.2025)

References

1. Sadovaya E.N. Modern Challenges and Threats to REST API Information Security and Methods for Their Prevention // Young Researcher of the Don. 2023. No. 3 (42). pp. 3–8. [Electronic resource]. URL: https://cyberleninka.ru/article/n/sovremennye-vyzovy-i-ugrozy-informatsionnoy-bezopasnosti-rest-api-i-sposoby-ih-predotvrascheniya

2. OWASP (Open Web Application Security Project) [Electronic resource]. URL: https://habr.com/ru/companies/owasp/articles/547174/

3. National Standard of the Russian Federation [Electronic resource]. URL: https://docs.cntd.ru/document/1200181890

4. Gorbunov T. Methodology for Building and Protecting APIs in Simple Terms [Electronic resource]. URL: https://www.itsec.ru/articles/metodologiya-postroeniya-i-zashchity-api-prostymi-slovami

5. Open Banking in Russia [Electronic resource]. URL: https://habr.com/ru/articles/668652/

6. Medaev Mark Kazbekovich. API Testing // E-Scio. 2023. No. 4 (79). pp. 8–12. [Electronic resource]. URL: https://cyberleninka.ru/article/n/testirovanie-api

7. 2025: Trends in API Security — From the Growth in Data Breaches to the Impact of AI [Electronic resource]. URL: https://www.tadviser.ru/index.php/Статья:Тренды_в_безопасности_API

8. Beeline Allowed a Major Data Leak. Data of Hundreds of Thousands of the Operator’s Employees Appeared Online [Electronic resource]. URL: https://www.cnews.ru/news/top/2022-12-02_bilajn_dopustil_krupnuyu

9. Beeline Launched an Investigation into a Data Leak from the Corporate Directory [Electronic resource]. URL: https://www.comnews.ru/content/223350/2022-12-02/2022-w48/bilayn-nachal-rassledovanie-utechki-dannykh-korporativnogo-spravochnika

10. Sportmaster Admitted a Large-Scale Leak of Customer Data [Electronic resource]. URL: https://www.itsec.ru/news/sportmaster-priznal-krupnomasshtabnutu-utechku-dannih-klientov

11. Zverkova Tatyana Nikolaevna. Open API in Regional Banks: Challenges and Directions of Implementation // Accounting. Analysis. Audit. 2025. No. 3, Vol. 12. pp. 4–12. [Electronic resource]. URL: https://cyberleninka.ru/article/n/open-api-v-regionalnyh-bankah-vyzovy-i-napravleniya-vnedreniya

12. Recommendations for Mitigating the OWASP API Security Top 10 Using API Management [Electronic resource]. URL: https://learn.microsoft.com/ru-ru/azure/api-management/mitigate-owasp-api-threats