УДК 004.056.57 Вестник УрФО № 2(56) / 2025, с. 30–39

Серебряков Д. С.

DOI: 10.14529/secur250203

ПРОАКТИВНАЯ ЗАЩИТА КОРПОРАТИВНЫХ ВЕБ-ПРИЛОЖЕНИЙ ОТ УГРОЗ, СВЯЗАННЫХ С КОМПРОМЕТАЦИЕЙ JAVASCRIPT-БИБЛИОТЕК

В статье анализируются угрозы безопасности веб-приложений, возникающие в результате использования сторонних JavaScript-библиотек, загружаемых в браузер с внешних ресурсов. Предлагается архитектура информационной системы, предназначенной для обеспечения интеллектуальной проактивной безопасности на уровне клиентской части корпоративных систем. В качестве основы для реализации системы рассматривается комбинация серверного и браузерного компонентов, обеспечивающих контроль целостности кода, мониторинг поведения исполняемых скриптов и фильтрацию вредоносных зависимостей. Подчеркивается важность централизованного управления безопасностью зависимостей и непрерывного мониторинга, позволяющего своевременно выявлять аномалии, недоступные для традиционных средств защиты. Полученные результаты могут быть использованы при разработке защищенных корпоративных систем на основе архитектуры тонкого клиента.

Ключевые слова: *информационная безопасность, веб-приложения, JavaScript, open source, проактивная защита, клиентская часть, зависимость*

Serebryakov D. S.

PROACTIVE SECURITY OF CORPORATE WEB APPLICATIONS AGAINST JAVASCRIPT LIBRARY COMPROMISE

The article analyzes security threats to web applications caused by the use of third-party JavaScript libraries loaded into the browser from external sources. A system architecture is proposed to ensure intelligent proactive client-side protection of corporate systems. The solution is based on a combination of server-side and browser-side components, which provide code integrity validation, behavior monitoring of executable scripts, and real-time filtering of malicious dependencies. The study emphasizes the importance of centralized dependency management and continuous monitoring for detecting anomalies that bypass traditional protection tools. The proposed results can be applied to the development of secure corporate systems built on thin-client architecture.

Keywords: *information security, web applications, JavaScript, open source, proactive protection, client-side, dependencies*

Современные корпоративные информационные системы используют архитектуру тонкого клиента для создания веб-приложений, в рамках которой обработка данных и взаимодействие с пользователем происходят непосредственно в браузере. Данный подход упрощает управление логикой АРМ пользователей и снижает нагрузку на серверы, однако открывает новые типы угроз, связанные с возможностью манипуляций клиентскими данными и подменой загружаемого контента [1, 2].

Одной из наиболее серьезных угроз является использование скомпрометированных, но формально легитимных JavaScript-библиотек, загружаемых со сторонних ресурсов (сторонние JS-компоненты). Такие компоненты могут выполнять скрытые действия по подмене данных, манипуляции контентом или взаимодействовать с несанкционированными ресурсами. Эти угрозы зачастую проходят сигнатурные проверки, не обнаруживаются антивирусными решениями и отсутствуют в известных базах уязвимостей [3, 4].

Современные веб-приложения используют тысячи сторонних JavaScript-библиотек, которые динамически управляются системой управления зависимостями, такой как npm. Этот механизм автоматической загрузки, обновления и интеграции компонентов из внешних репозиториев значительно упрощает процесс разработки, но в то же время создает потенциально неконтролируемый вектор атак. Разработчики и администраторы зачастую не отслеживают весь спектр загружаемых зависимостей, полагаясь на механизмы пакетных менеджеров, которые автоматически устанавливают и обновляют пакеты из множества источников. В результате уязвимости или преднамеренно внедренные вредоносные изменения в одном из многочисленных пакетов могут незаметно распространиться по всей экосистеме приложения.

Кроме того, масштабное использование автоматизированного управления зависимостями приводит к тому, что в проектах присутствуют сотни и тысячи зависимостей, поступающих из различных источников. В этих условиях традиционные механизмы статического анализа кода и мониторинга целостности компонентов оказываются недостаточно эффективными. Разнообразие поставщиков и версионных цепочек делает практически невозможным выявление скрытых угроз в огромных объемах данных, что усложняет задачу обеспечения безопасности и требует разработки специализированных механизмов защиты (см. Табл. 1).

Таблица 1

Основные угрозы, связанные с компрометацией сторонних JavaScript-библиотек

№	Тип угрозы	Механизм атаки	Последствия для веб-приложений
1	Подмена данных [5, 6]	Модификация DOM через популярные библиотеки (Vue.js, jQuery.js, WPForms.js)	Искажение информации (politware)
2	Перехват запросов [7]	Вставка обработчиков сетевых взаимодействий	Кража данных, перенаправление на фишинговые сайты
3	Изменение интерфейсов [8]	Скрытие или подмена форм, элементов управления	Социальная инженерия, хищение информации
4	Скрытая активность [9]	Фоновые сетевые соединения с внешними серверами	Утечка данных, заражение приложений
5	Компрометация репозиториев [10, 11]	Недостаточный контроль npm, CDN, GitHub	Распространение вредоносных зависимостей

Анализ Банка данных угроз ФСТЭК России подтверждает актуальность обсуждаемой проблемы. УБИ «Доступ к локальным файлам сервера при нарушении целостности скриптов» описывает сценарий, при котором вредоносные или модифицированные JavaScript-библиотеки могут инициировать несанкционированные обращения к внутренним ресурсам, что особенно критично для веб-приложений с архитектурой тонкого клиента [12].

Цель, задачи и методы исследования

Целью работы является определение технических требований к системе защиты веб-приложений корпоративных информационных систем от угроз, связанных с использованием сторонних JavaScript-библиотек. Для достижения цели поставлены задачи:

- выделить требования к архитектуре серверного компонента системы, включая организацию базы знаний, контроль версий зависимостей и механизмов уведомления;
- рассмотреть направления интеграции предложенного подхода с действующими средствами информационной безопасности корпоративной инфраструктуры.

Методологической основой исследования выступают:

- анализ типовых сценариев использования сторонних JavaScript-библиотек в современных веб-приложениях;
- изучение особенностей динамического поведения исполняемого кода в браузере в условиях внедрения внешних компонентов;
- сопоставление методов статического и поведенческого контроля в контексте обеспечения целостности клиентской части веб-приложений.

При прикладной реализации предложенных принципов проактивной защиты, исключающей загрузку и исполнение нежелательных компонентов до момента их воздействия на защищаемую корпоративную информационную систему.

Распространенность угроз и их последствия

Современные веб-приложения широко используют компоненты с открытым исходным кодом. Исследования показывают, что в 96% приложений присутствует открытый код [13], что подтверждает широкое распространение сторонних JS-компонентов [14], в частности было проанализировано более 12 миллионов случаев использования свободного и открытого программного обеспечения (FOSS) в более чем 10 тысячах компаний. Рассмотрим основные угрозы, характерные для атак, связанных с использованием уязвимостей в сторонних JavaScript-библиотеках. Злоумышленники могут загружать вредоносные версии библиотек в открытые репозитории (npm, GitHub), что может приводить к компрометации данных и утечкам информации [14]. Такая атака может привести к подмене данных, внедрению вредоносного кода и утечке конфиденциальной информации.

Системы доставки контента (CDN, Content Delivery Network) применяются в веб-разработке для обеспечения высокой скорости загрузки сторонних JavaScript-библиотек

и снижения нагрузки на основные серверы. Однако использование внешних CDN-источников влечёт за собой значительные риски. В случае компрометации такого узла злоумышленник получает возможность внедрения поддельного или модифицированного кода, который будет исполняться непосредственно в браузере конечного пользователя. Это может привести к перехвату данных, несанкционированному доступу к пользовательским сессиям или подмене интерфейсных элементов [3, 4].

Дополнительную угрозу представляют библиотеки, изначально содержащие вредоносные элементы, реализованные посредством легитимных возможностей языка JavaScript. Такие компоненты способны незаметно изменять структуру DOM (Document Object Model), в частности — скрывать элементы формы, подменять действия по нажатию кнопок или внедрять скрытые переходы. Подобные методы активно используются при атаках социальной инженерии и остаются слабо детектируемыми средствами статического анализа [5].

В ряде случаев вредоносный код встраивается на этапе разработки библиотеки — в pull-запросах, новых релизах или дополнительных зависимостях. Такие модификации могут длительное время не вызывать подозрений у участников сообщества или администраторов репозиториев, и, как следствие, распространяются через популярные пакеты среди тысяч других проектов [6]. Ярким примером таких атак стали инциденты 2021–2022 годов, когда изменения в цепочке поставок ПО позволили злоумышленникам внедрять вредоносный код в широко используемые библиотеки, не прибегая к взлому конечных приложений [7].

Особую известность получил случай в январе 2025 года, когда было выявлено заражение более 500 правительственных и университетских сайтов по всему миру через внедрение вредоносных JavaScript-компонентов. Они создавали скрытые элементы в DOM, обеспечивавшие автоматическую переадресацию пользователей на сторонние ресурсы. Атака не была зафиксирована средствами традиционного мониторинга, что продемонстрировало слабость существующих систем обнаружения и необходимость их совершенствования [15].

В 2025 году была зафиксирована волна атак на сайты, работающие на платформе «1С-Битрикс». Злоумышленники использовали уязвимости в сторонних модулях, таких как esol.importexcel и currency_export_excel, для внедрения вредоносного JavaScript-кода. Это привело к перенаправлению пользователей на фишинговые ресурсы и установке вредоносного ПО. [16]. В 2021 году популярная библиотека UAParser.js, используемая во многих веб-приложениях, была скомпрометирована. Злоумышленники внедрили в неё вредоносный код, который распространялся через официальный репозиторий npm. Это привело к заражению множества систем, включая российские компании, использующие эту библиотеку [17].

Учитывая изложенное, можно заключить, что широкое распространение и сложность отслеживания уязвимых или скомпрометированных JS-библиотек делают их одним из наиболее критичных элементов угроз информационной безопасности веб-приложений.

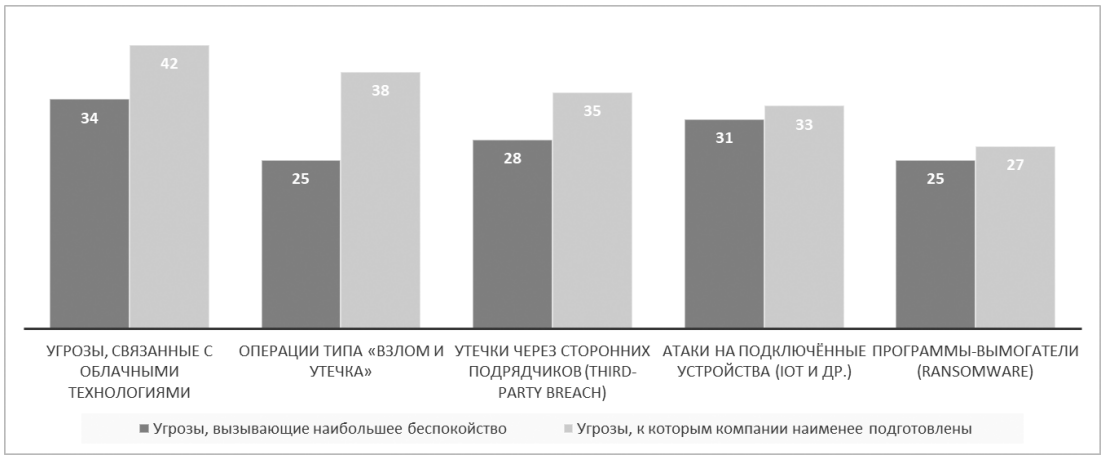

Рис. 1. Разрыв между обеспокоенностью и готовностью к отражению киберугроз - доля респондентов, выбравших угрозу в топ-3 [18]

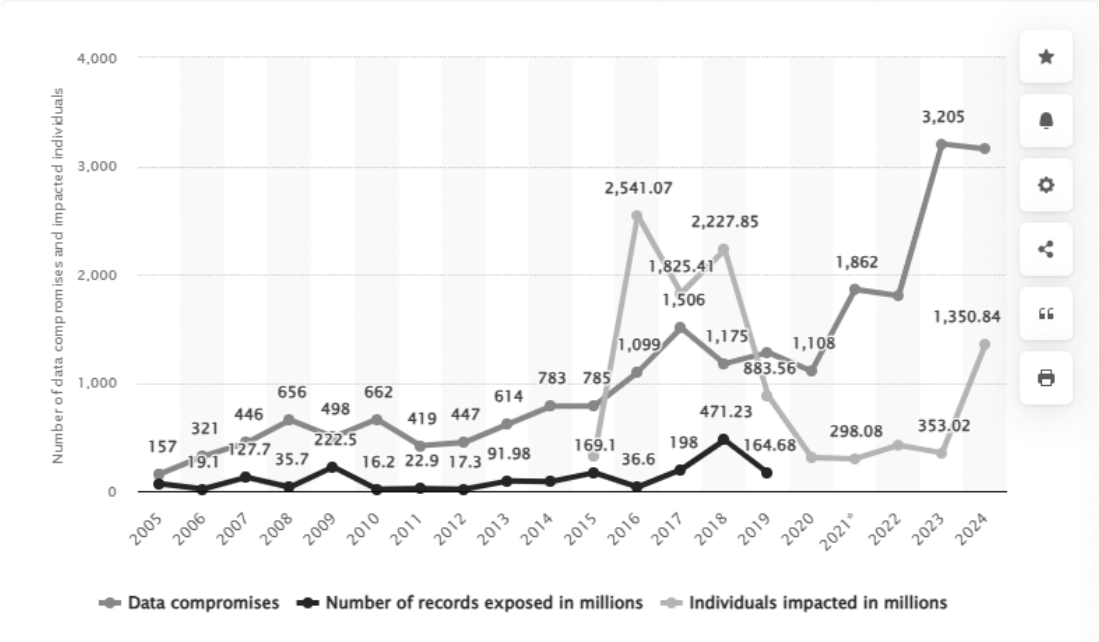

Рис. 2. Ежегодное количество случаев компрометации данных и пострадавших лиц в США с 2005 по 2024 г. [20]

Такое распространение уязвимостей подтверждается результатами международных исследований. Угрозы, связанные с третьесторонними утечками и атаками через внешние зависимости, входят в число наиболее тревожных факторов для организаций, при этом уровень готовности к их отражению остаётся недостаточным (см. Рис. 1).

Также исследование 2020 г. показало, что 37% из 133 тыс. проанализированных вебсайтов использовали по крайней мере одну JavaScript-библиотеку с известной уязвимостью, что свидетельствует о недостаточной защищенности существующих систем перед подобными угрозами [19]. Повышение уровня защищённости требует регулярной верификации используемых зависимостей, внедрения систем контроля целостности и развития механизмов защиты на клиентской стороне.

Последствия от киберинцидентов продолжают увеличиваться (см. Рис 2).

По данным аналитиков, совокупные расходы на обеспечение информационной безопасности в 2024 году достигнут $183,87 млрд, что на 13,4% больше по сравнению с предыдущим годом [21]. Эти затраты сопоставимы с годовыми бюджетами отдельных государств: например, ВВП Новой Зеландии или Греции находится на аналогичном уровне. Такой масштаб подчёркивает не только экономическую значимость вопроса, но и необходимость системного подхода к минимизации уязвимостей в инфраструктуре современных веб-приложений.

Предлагаемые технические требования к технологии

Современные методы обеспечения информационной безопасности (ИБ) часто оказываются недостаточно эффективными против новых векторов атак, связанных с использованием сторонних JavaScript-библиотек [22]. Традиционные системы, такие как файрволы веб-приложений (WAF), в основном нацелены на предотвращение известных угроз, но могут быть не в состоянии обнаружить и предотвратить атаки, возникающие из-за компрометации сторонних библиотек.

Комплексный подход защиты должен включать динамический анализ и мониторинг изменений в репозиториях, а также механизмы безопасности на уровне браузера [8], что отражено в авторском подходе к усилению защиты веб-приложений от уязвимостей в сторонних JavaScript-библиотеках [23, 24]. Основные различия между традиционными подходами и предлагаемым усилением представлены ниже (см. Табл. 2).

Такой подход подразумевает создание информационной системы, предназначенной для проактивного мониторинга действий в браузере, контроля и валидации исполняемого кода и интегрируемой с веб-

Таблица 2

Структура предлагаемого комплексного подхода

№	Ключевой элемент	Ограничения традиционных решений	Элементы предлагаемого подхода
1	Управление загрузкой JavaScript-зависимостей	Отсутствие динамического контроля, ручная проверка	Динамическое профилирование и мониторинг загрузки скриптов в браузере
2	Проверка целостности и изменений кода	Редкий аудит, отсутствие автоматизированного отслеживания	Автоматический мониторинг изменений в open-source библиотеках
3	Контроль сетевых взаимодействий	Защита только на периметре сети (WAF)	Анализ исходящих HTTP-запросов и загружаемого контента в браузере
4	Реагирование на выявленные угрозы	Реакция только после инцидента	Проактивное уведомление о подозрительной активности и динамическая коррекция политик безопасности (CSP)

приложением корпоративной информационной системы, что позволит существенно снизить риски, связанные с упомянутыми ранее векторами атак. Архитектурно система, обеспечивающая интеллектуальную проактивную безопасность веб-приложений, должна сочетать централизованный серверный компонент и комплексный браузерный модуль.

Ключевым элементом предлагаемой архитектуры является расширенный браузерный модуль, обеспечивающий реализацию концепции проактивной клиентской защиты [25]. Одной из его важнейших функций выступает динамический мониторинг загружаемых в браузере скриптов, сопровождающийся проверкой их соответствия политике безопасности, заданной на сервере. Это позволяет в момент загрузки выявлять подозрительные компоненты и блокировать их до начала исполнения.

Контроль целостности реализуется посредством верификации контрольных хешсумм библиотек, что позволяет обнаруживать даже минимальные несанкционированные изменения кода, в том числе незаметные визуально, но опасные по воздействию. Для минимизации последствий потенциального заражения применяется механизм ограничения полномочий сторонних библиотек. Принцип изоляции по минимуму прав предотвращает выполнение операций с чувствительными объектами DOM, утечку данных и вмешательство в ключевые функции веб-приложения.

Браузерный компонент также обеспечивает обратную связь с сервером, передавая информацию о выявленных отклонениях и нестандартных действиях библиотек. Это позволяет обеспечить синхронизацию между клиентской и серверной частями системы, а также оперативное обновление защитных правил. Особое внимание уделено автоматизации процесса обновления: браузерный модуль инициирует применение патчей и загрузку актуальных версий библиотек, в случае если ранее установленные версии были признаны уязвимыми.

Центральный серверный компонент разработанной системы обеспечивает хранение данных и выполняет функции координации, анализа и политики управления безопасностью [26]. В частности, в его задачи входит хранение информации об используемых open-source библиотеках, их версиях и известных уязвимостях. Интеграция с общедоступными базами, такими как CVE, обеспечивает своевременное обновление данных о потенциальных угрозах, организация оперативного реагирования на инциденты. Система уведомлений реализует механизм автоматической передачи сообщений ответственным специалистам при обнаружении признаков угроз — будь то выявленные уязвимости в используемых JavaScript-компонентах или отклонения от их стандартного поведения. Такие уведомления направлены на снижение времени реакции и минимизацию последствий атак.

Важным элементом является настройка и контроль политик безопасности, определяющих перечень разрешённых к использованию библиотек, их версий и источников распространения. Фильтрация доступа к несанкционированным компонентам позволяет на раннем этапе предотвращать загрузку потенциально опасного кода, в том числе скомпрометированных пакетов из внешних CDN или репозиториев. Дополнительно предусмотрена система аналитики, обеспечивающая регулярное формирование отчетов, содержащих информацию о текущем уровне защищенности, динамике обновлений зависимостей и зафиксированных инцидентах. Эти данные используются для оценки уязвимостей и выработки корректирующих мер на уровне организации.

Практическая реализация предлагаемой архитектуры потенциально имеет высокую эффективность в предотвращении атак, связанных с использованием сторонних JavaScript-компонентов. Ключевым преимуществом системы является проактивный характер защиты: аномалии выявляются до начала их эксплуатации, что принципиально отличает подход от реактивных средств контроля. Благодаря модульной структуре и открытым интерфейсам, система легко интегрируется в существующую ИТ-инфраструктуру компании, включая конвейеры DevSecOps, корпоративные SIEM-системы и средства WAF. Это обеспечивает непрерывный мониторинг клиентской стороны без необходимости масштабной перестройки архитектуры.

Гибкость архитектуры позволяет адаптировать её к изменениям в экосистеме, основанной на open-source: обновления библиотек, изменения политик безопасности и появление новых угроз оперативно обрабатываются системой за счёт централизованного управления и синхронизации с базой знаний (уязвимостей и политик). В совокупности предложенный подход позволяет рассматривать браузер не как уязвимую зону, а как полнофункциональную и контролируемую среду исполнения, интегрированную в архитектуру защиты корпоративной информационной системы.

Таким образом, предложенный подход к защите клиентской части веб-приложений от угроз, связанных с компрометацией сторонних JavaScript-библиотек, позволяет выстроить проактивную архитектуру информационной безопасности, минимизирующую риски внедрения вредоносного кода. Разработанные технические меры применимы к корпоративным ИТ-системам с архитектурой тонкого клиента и легко интегрируются в существующие процессы SIEM-мониторинга. Основные ожидаемые эффекты представлены в таблице 3. Следующий этап научного исследования предполагает разработку прототипа и интеграции предложенной системы в реальную ИБ-архитектуру предприятия.

Таблица 3

Ожидаемые эффекты от внедрения проактивной системы защиты

№	Вектор угроз	Эффект от внедрения предлагаемого подхода
1	Подмена данных через модификацию DOM	Блокировка исполнения скриптов, нарушающих контрольные хеши и политику CSP
2	Загрузка модифицированных JS-зависимостей	Автоматическая верификация источника и контрольных сумм; уведомление об отклонениях
3	Перехват и переадресация пользовательских данных	Анализ исходящих HTTP-запросов и блокировка несанкционированных направлений
4	Внедрение вредоносного кода через open-source	Мониторинг репозиториев и сигнатурный анализ изменений в библиотеках
5	Атаки через CDN и внешние ресурсы	Ограничение по источникам, фильтрация недоверенных URL и проверка сетевой активности

Литература

1. CVEfixes: Automated Collection of Vulnerabilities and Their Fixes from Open-Source Software. arXiv. URL: https://arxiv.org/abs/2107.08760 (дата обращения 28.04.2025 г.)

2. Automated Vulnerability Detection in Source Code Using Deep Representation Learning. arXiv. URL: https://arxiv.org/abs/1807.04320 (дата обращения 28.04.2025 г.)

3. 3Proactive Vulnerability Management is a No Brainer for Security, but... JFrog. URL: https://jfrog.com/blog/proactive-vulnerability-management/ (дата обращения 28.04.2025 г.)

4. 13 Open Source Software Security Risks. SentinelOne. URL: https://www.sentinelone.com/cybersecurity-101/cybersecurity/open-source-software-security-risks/ (дата обращения 28.04.2025 г.)

5. MISP Open Source Threat Intelligence Platform & Open Standards. MISP Project. URL: https://www.misp-project.org/ (дата обращения 28.04.2025 г.)

6. Strengthening Open Source Software: Best Practices for Enhanced Security. OpenSSF. URL: https://openssf.org/blog/2023/09/06/strengthening-open-source-software-best-practices-for-enhanced-security/ (дата обращения 28.04.2025 г.)

7. What Are Open Source Vulnerabilities. Sonatype. URL: https://www.sonatype.com/resources/articles/what-are-open-source-vulnerabilities (дата обращения 28.04.2025 г.)

8. Open Source Security and Risk Analysis Report Trends. Black Duck. URL: https://www.blackduck.com/blog/open-source-trends-ossra-report.html (дата обращения 28.04.2025 г.)

9. 10 Free and Open Source Cybersecurity Tools to Know. Lumifi Cyber. URL: https://www.lumificyber.com/blog/free-open-source-software-cybersecurity/ (дата обращения 28.04.2025 г.)

10. Information Security and Protection of Information. MSU. URL: https://hsmi.msu.ru/sites/hsmi.msu.ru/files/program_common_files/po_vyb._-_informacionnaya_bezopasnost_i_zashchita_informacii_0.pdf (дата обращения 28.04.2025 г.)

11. Theoretical Foundations of Information Security. MSU. URL: https://cs.msu.ru/sites/cmc/files/docs/_26_08_teoreticheskie_osnovy_informacionnoy_bezopasnosti.pdf (дата обращения 28.04.2025 г.)

12. Банк данных угроз безопасности информации, URL https://bdu.fstec.ru/threat (дата обращения 28.05.2025 г.)

13. Исследование показало, что 96% современных приложений используют open-source. CNews. URL: https://www.cnews.ru/news/top/2024-12-10_issledovanie_pokazalochto (дата обращения 28.04.2025 г.)

14. В открытом ПО обнаружили 12 млн уязвимостей: компании используют библиотеки, не проверяя их. Habr. URL: https://habr.com/ru/news/865290/ (дата обращения 28.04.2025 г.)

15. На более чем 500 сайтах по всему миру обнаружена новая атака с использованием JavaScript. ITSec.ru. URL: https://www.itsec.ru/news/na-bolee-chem-500-saytah-po-vsemu-miru-obnaruzhena-novaya-ataka-s-ispolzovaniyem-javascript (дата обращения 28.04.2025 г.)

16. Kaspercky Daily. В популярный Javascript-пакет UAParser.js внедрили зловреда URL: https://www.kaspersky.ru/blog/uaparser-js-infected-versions/31787/ (дата обращения 28.05.2025 г.)

17. RushRadio Новая волна вирусов на Битрикс в 2025 году URL: https://rushstudio.by/blog/razrabotchiku/novaya-volna-virusov-na-bitriks-v-2025-godu/ (дата обращения 28.05.2025 г.)

18. Отчёт PwC Global Digital Trust Insights 2025, URL https://www.pwc.com/us/en/services/consulting/cybersecurity-risk-regulatory/library/global-digital-trust-insights.html (дата обращения 28.04.2025 г.)

19. JavaScript Security: A Survey of Attacks and Defenses. arXiv. URL: https://arxiv.org/abs/1811.00918 (дата обращения 28.04.2025 г.)

20. Статистические данные сайта Statista.com URL: https://www.statista.com/statistics/273550/data-breaches-recorded-in-the-united-states-by-number-of-breaches-and-records-exposed/ (дата обращения 28.04.2025 г.)

21. Информационная безопасность (мировой рынок). TAdviser. URL: https://www.tadviser.ru/index.php/Статья:Информационная_безопасность_%28мировой_рынок%29 (дата обращения 28.04.2025 г.)

22. Обзор угроз информационной безопасности за I квартал 2024 года. Positive Technologies. URL: https://www.ptsecurity.com/ru ru/research/analytics/cybersecurity-threatscape-2024-q1/ (дата обращения 28.04.2025 г.)

23. Vasilakis N., Staicu C.-A., Ntousakis G., Kallas K., Karel B., DeHon A., Pradel M. Mir: Automated Quantifiable Privilege Reduction Against Dynamic Library Compromise in JavaScript. arXiv preprint arXiv:2011.00253. URL: https://arxiv.org/abs/2011.00253

24. Nakhaei K., Ansari E., Ansari F. JSSignature: Eliminating Third-Party-Hosted JavaScript Infection Threats Using Digital Signatures. arXiv preprint arXiv:1812.03939. URL: https://arxiv.org/abs/1812.03939

25. Государев И.Б. Основы разработки веб-приложений на платформах Node.js и Deno. СПб: Университет ИТМО, 2023.

26. Хоффман Э. Безопасность веб-приложений. Разведка, защита, нападение. СПб: Питер, 2021.

References

1. CVEfixes. Automated Collection of Vulnerabilities and Their Fixes from Open-Source Software. arXiv. URL: https://arxiv.org/abs/2107.08760 (accessed: 28.04.2025)

2. Automated Vulnerability Detection in Source Code Using Deep Representation Learning. arXiv. URL: https://arxiv.org/abs/1807.04320 (accessed: 28.04.2025)

3. JFrog. Proactive Vulnerability Management is a No Brainer for Security. URL:https://jfrog.com/blog/proactive-vulnerability-management/ (accessed: 28.04.2025)

4. SentinelOne. 13 Open Source Software Security Risks. URL: https://www.sentinelone.com/cybersecurity-101/cybersecurity/open-source-software-security-risks/ (accessed: 28.04.2025)

5. MISP Project. MISP Open Source Threat Intelligence Platform & Open Standards. URL: https://www.misp-project.org/ (accessed: 28.04.2025)

6. OpenSSF. Strengthening Open Source Software: Best Practices for Enhanced Security. URL: https://openssf.org/blog/2023/09/06/strengthening-open-source-software-best-practices-for-enhanced-security/ (accessed: 28.04.2025)

7. Sonatype. What Are Open Source Vulnerabilities. URL: https://www.sonatype.com/resources/articles/what-are-open-source-vulnerabilities (accessed: 28.04.2025)

8. Black Duck. Open Source Security and Risk Analysis Report Trends. URL: https://www.blackduck.com/blog/open-source-trends-ossra-report.html (accessed: 28.04.2025)

9. Lumifi Cyber. 10 Free and Open Source Cybersecurity Tools to Know. URL: https://www.lumificyber.com/blog/free-open-source-software-cybersecurity/ (accessed: 28.04.2025)

10. Moscow State University. Information Security and Protection of Information. URL: https://hsmi.msu.ru/sites/hsmi.msu.ru/files/program_common_files/po_vyb._-_informacionnaya_bezopasnost_i_zashchita_informacii_0.pdf (accessed: 28.04.2025)

11. Moscow State University. Theoretical Foundations of Information Security. URL: https://cs.msu.ru/sites/cmc/files/docs/_26_08_teoreticheskie_osnovy_informacionnoy_bezopasnosti.pdf (accessed: 28.04.2025)

12. Bank dannykh ugroz bezopasnosti informatsii, URL https://bdu.fstec.ru/threat (data obrashcheniya 28.05.2025 g.)

13. Issledovaniye pokazalo, chto 96% sovremennykh prilozheniy ispol'zuyut open-source. CNews. URL: https://www.cnews.ru/news/top/2024-12-10_issledovanie_pokazalochto (data obrashcheniya 28.04.2025 g.)

14. V otkrytom PO obnaruzhili 12 mln uyazvimostey: kompanii ispol'zuyut biblioteki, ne proveryaya ikh. Habr. URL: https://habr.com/ru/news/865290/ (data obrashcheniya 28.04.2025 g.)

15. Na boleye chem 500 saytakh po vsemu miru obnaruzhena novaya ataka s ispol'zovaniyem JavaScript. ITSec.ru. URL: https://www.itsec.ru/news/na-bolee-chem-500-saytah-po-vsemu-miru-obnaruzhena-novaya-ataka-s-ispolzovaniyem-javascript (data obrashcheniya 28.04.2025 g.)

16. Kaspercky Daily. V populyarnyy Javascript-paket UAParser.js vnedrili zlovreda URL: https://www.kaspersky.ru/blog/uaparser-js-infected-versions/31787/ (data obrashcheniya 28.05.2025 g.)

17. RushRadio Novaya volna virusov na Bitriks v 2025 godu URL: https://rushstudio.by/blog/razrabotchiku/novaya-volna-virusov-na-bitriks-v-2025-godu/ (data obrashcheniya 28.05.2025 g.)

18. PwC. Global Digital Trust Insights 2025. URL: https://www.pwc.com/us/en/services/consulting/cybersecurity-risk-regulatory/library/global-digital-trust-insights.html (accessed: 28.04.2025)

19. JavaScript Security: A Survey of Attacks and Defenses. arXiv. URL: https://arxiv.org/abs/1811.00918 (accessed: 28.04.2025)

20. Statisticheskiye dannyye sayta Statista.com URL: https://www.statista.com/statistics/273550/data-breaches-recorded-in-the-united-states-by-number-of-breaches-and-records-exposed/ (data obrashcheniya 28.04.2025 g.)

21. Informatsionnaya bezopasnost' (mirovoy rynok). TAdviser. URL: https://www.tadviser.ru/index.php/Stat'ya:Informatsionnaya_bezopasnost'_%28mirovoy_rynok%29 (data obrashcheniya 28.04.2025 g.)

22. Obzor ugroz informatsionnoy bezopasnosti za I kvartal 2024 goda. Positive Technologies. URL: https://www.ptsecurity.com/ru ru/research/analytics/cybersecurity-threatscape-2024-q1/ (data obrashcheniya 28.04.2025 g.)

23. Vasilakis N., Staicu C.-A., Ntousakis G., Kallas K., Karel B., DeHon A., Pradel M. Mir: Automated Quantifiable Privilege Reduction Against Dynamic Library Compromise in JavaScript. arXiv, 2020. URL: https://arxiv.org/abs/2011.00253

24. Nakhaei K., Ansari E., Ansari F. JSSignature: Eliminating Third-Party-Hosted JavaScript Infection Threats Using Digital Signatures. arXiv, 2018. URL: https://arxiv.org/abs/1812.03939 (accessed: 28.04.2025)

25. Gosudarev I.B. Osnovy razrabotki veb-prilozheniy na platformakh Node.js i Deno. SPb: Universitet ITMO, 2023.

26. Khoffman E. Bezopasnost' veb-prilozheniy. Razvedka, zashchita, napadeniye. SPb: Piter, 2021.

СЕРЕБРЯКОВ Дмитрий Сергеевич, генеральный директор ООО «БИТ Автоматизация». 115487, г. Москва, ул, Нагатинская д. 16, пом. 1/21в/10. E-mail: sm.house.2016@gmail.com

SEREBRYAKOV Dmitry Sergeevich, Chief Executive Officer, BIT Automation LLC. 115487, Moscow, st. Nagatinskaya, bldg. 16, office 1/21v/10. E-mail: sm.house.2016@gmail.com