DOI: 10.12731/2227-930X-2024-14-3-297
УДК 004.056

Научная статья |
Системный анализ, управление и обработка информации, статистика

АНАЛИЗ И МИНИМИЗАЦИЯ РИСКОВ ВРЕДОНОСНЫХ ЗАВИСИМОСТЕЙ В ПРОЦЕССЕ НЕПРЕРЫВНОЙ ИНТЕГРАЦИИ И ВНЕДРЕНИЯ ПРОГРАММНОГО КОДА

Д.Н. Алексеев, Р.М. Хамитов

Процессы непрерывной интеграции и непрерывного развертывания (CI/CD) стали важнейшими элементами современной разработки программного обеспечения, обеспечивая автоматизацию и оптимизацию рабочих процессов. Однако эти процессы сопровождаются рисками, связанными с уязвимостями в цепочке зависимостей, что может привести к серьезным последствиям, таким как несанкционированный доступ и утечка данных. В статье рассматривается необходимость внедрения надежных механизмов обнаружения и смягчения рисков зависимостей для повышения надежности CI/CD.

Основной риск в процессах CI/CD связан с эксплуатацией вредоносных зависимостей, используемых в процессе сборки и развертывания программного обеспечения. Основные виды атак включают путаницу зависимостей, перехват зависимостей и кибersквоттинг опечаток. Для предотвращения этих угроз предлагаются различные методы защиты, такие как контроль доступа к приватным пакетам, использование автоматизированных инструментов для мониторинга и проверки зависимостей, а также внедрение систем машинного обучения для обнаружения подозрительных пакетов. Эти меры направлены на обеспечение целостности и безопасности программных продуктов, минимизируя риски, связанные с зависимостями в CI/CD.

Цель *– проанализировать атаки на цепочку зависимостей и определить эффективные методы решения рисков для обеспечения высокой безопасности процессов непрерывной интеграции и развертывания для улучшения практики разработки программного обеспечения путем выявления и устранения потенциальных уязвимостей и проблем стабильности, что обеспечивает более безопасные и надежные конвейеры доставки программного обеспечения, снижая вероятность сбоев и сбоев в производственных средах.*

Метод и методология проведения работы. *Данная работа включает в себя результаты как международных, так и местных научных исследований. Для выявления взаимосвязей и получения оригинальных выводов автор использует теоретические методы исследования, уделяя особое внимание поиску и анализу информации. Авторами применяются теоретические методы исследования, связанные с поиском и анализом информации для выявления связей и получения уникальных выводов.*

Результаты. *Проведен анализ рисков вредоносных зависимостей в процессе непрерывной интеграции и внедрения программного кода. Определены методы минимизации рисков злоупотребления зависимостями, необходимость внедрять многоуровневые меры безопасности, включая автоматизированные инструменты для мониторинга и анализа, строгий контроль доступа к репозиториям и использование криптографических методов для проверки целостности пакетов. Кроме того, регулярные аудиты и обучение сотрудников помогают поддерживать высокий уровень безопасности и осведомленности о потенциальных угрозах.*

Область применения результатов. *Полученные результаты целесообразно применять в области DevOps разработки с целью оптимизации процесса разработки и выпуска приложений путем устранения известного узкого места: минимизация рисков вредоносных зависимостей в процессе непрерывной интеграции.*

Ключевые слова: *DevOps; управление данными; безопасность; DevSecOps; контроль версий зависимостей; управление зависимо-*

стями; PyPI; минимизация рисков; безопасная разработка; паттерны

Для цитирования. *Алексеев Д.Н., Хамитов Р.М. Анализ и минимизация рисков вредоносных зависимостей в процессе непрерывной интеграции и внедрения программного кода // International Journal of Advanced Studies. 2024. Т. 14, № 3. С. 100-116. DOI: 10.12731/2227-930X-2024-14-3-297*

Original article |
System Analysis, Management and Information Processing, Statistics

ANALYSIS AND MINIMIZATION OF THE RISKS OF HARMFUL DEPENDENCIES IN THE PROCESS OF CONTINUOUS INTEGRATION AND IMPLEMENTATION OF SOFTWARE CODE

D.N. Alekseev, R.M. Khamitov

Continuous integration and continuous deployment (CI/CD) processes have become essential elements of modern software development, enabling automation and optimization of work processes. However, these processes come with risks associated with vulnerabilities in the dependency chain, which can lead to serious consequences such as unauthorized access and data leakage. The article discusses the need to implement reliable mechanisms for detecting and mitigating dependency risks to improve the reliability of CI/CD.

The main risk in CI/CD processes is the exploitation of malicious dependencies used during the software build and deployment process. The main types of attacks include dependency confusion, dependency hijacking, and typo cybersquatting. To prevent these threats, various protection methods are proposed, such as controlling access to private packages, using automated tools for monitoring and checking dependencies, and implementing machine learning systems to detect suspi-

cious packages. These measures are aimed at ensuring the integrity and security of software products, minimizing the risks associated with dependencies in CI/CD.

Purpose. *Analyze dependency chain attacks and identify effective risk management methods to ensure high security of continuous integration and deployment processes to improve software development practices by identifying and eliminating potential vulnerabilities and stability issues, which provides safer and more reliable software delivery pipelines, reducing the likelihood of failures and disruptions in production environments.*

Methodology. *This work includes the results of both international and local scientific research. To identify the relationships and obtain original conclusions, the author uses theoretical research methods, paying special attention to the search and analysis of information. The authors apply theoretical research methods related to the search and analysis of information to identify connections and obtain unique conclusions.*

Results. *The analysis of the risks of malicious dependencies in the process of continuous integration and implementation of the program code is carried out. Methods have been identified to minimize the risks of dependency abuse, the need to implement multi-level security measures, including automated monitoring and analysis tools, strict access control to repositories and the use of cryptographic methods to verify the integrity of packages. In addition, regular audits and employee training help maintain a high level of security and awareness of potential threats.*

Practical implications. *It is advisable to apply the results obtained in the field of DevOps development in order to optimize the application development and release process by eliminating a known bottleneck: minimizing the risks of malicious dependencies in the process of continuous integration.*

Keywords: *DevOps; data management; safety; DevSecOps; dependency version control; dependency management; PyPI; risk minimization; safe development; patterns*

For citation. *Alekseev D.N., Khamitov R.M. Analysis and Minimization of the Risks of Harmful Dependencies in the Process Of Continuous Integration and Implementation of Software Code. International Journal of Advanced Studies, 2024, vol. 14, no. 3, pp. 100-116. DOI: 10.12731/2227-930X-2024-14-3-297*

Введение

Процессы непрерывной интеграции и непрерывного развертывания (Continuous Integration/Continuous Delivery) стали неотъемлемыми компонентами современной разработки программного обеспечения. Наряду с преимуществами оптимизированных рабочих процессов разработки возникают риски, особенно связанные с зависимостями, используемыми в экосистеме программного обеспечения [1].

Один из основных рисков в процессах CI/CD связан с эксплуатацией вредоносных зависимостей во время извлечения и использования библиотек и внешних пакетов. По данным экспертного сообщества OWASP [9] проблема злоупотребления цепочкой зависимостей может привести к серьезным последствиям, включая несанкционированный доступ, утечки данных и компрометацию системы.

В данной статье рассматривается необходимость реализации надежных механизмов обнаружения и эффективного смягчения рисков зависимостей [14]. Для начала мы выясним основные векторы атак, а после перечислим наиболее эффективные методы защиты зависимостей проектов.

Цель работы

Проанализировать атаки на цепочку зависимостей и определить эффективные методы решения рисков для обеспечения высокой безопасности процессов непрерывной интеграции и развертывания для улучшения практики разработки программного обеспечения путем выявления и устранения потенциальных уязвимостей и проблем стабильности, что обеспечивает более безопасные и надежные конвейеры доставки программного обеспечения, снижая вероятность сбоев и сбоев в производственных средах.

Материалы и методы исследования

Данная работа включает в себя результаты как международных, так и местных научных исследований. Для выявления взаимосвязей и получения оригинальных выводов автор использует теоретические методы исследования, уделяя особое внимание поиску и анализу информации.

Результаты исследования и их обсуждение

Прежде чем углубляться в детали того, как добиться высокой надежности CI/CD процессов, давайте убедимся, что у нас есть общее понимание нескольких концепций.

Надежность CI/CD – это последовательная и надежная работа конвейеров непрерывной интеграции и доставки, гарантируя, что изменения кода автоматически создаются, тестируются и развертываются с минимальными ошибками и временем простоя [2].

Зависимости – это внешние компоненты или библиотеки, от которых зависит правильная работа программного приложения, например определенные пакеты программного обеспечения, модули или системные ресурсы.

Злоупотребление цепочкой зависимостей – это процесс, когда злоумышленники используют уязвимости в способах управления и извлечения библиотек и внешних пакетов. Это может привести к случайному выполнению вредоносного кода в системе.

Пример этапов конвейера CI/CD показан на рисунке 1.

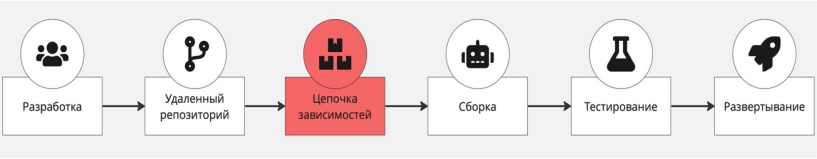

Рис. 1. Этапы конвейера CI/CD

Каждый автоматизированный блок, выполняемый в конвейере непрерывной интеграции и развертывания (CI/CD), имеет реша-

ющее значение, причем каждый последующий блок зависит от предыдущего [13]. На этапе разработки программисты пишут код для создания продукта. Затем этот код передается в удаленный репозиторий, что запускает конвейер CI/CD. Процесс начинается с этапа цепочки зависимостей и подготовки к сборке. После этого проект собирается, код проверяется на ошибки и конечный результат разворачивается на сервере.

Этап цепочки зависимостей особенно важен, поскольку он является основной целью злоумышленников. Далее мы перечислим основные векторы атак в контексте цепочек поставок и методы их устранения, для минимизации рисков на этапе сборок продуктов.

Начнем с атаки *Путаница зависимостей (Dependency confusion)* или *Атака на замещение (Substitution attack)*. В данной атаке производится публикация вредоносных пакетов в общедоступных репозиториях с тем же именем, что и внутренние имена пакетов, в попытке обманом заставить клиентов загружать вредоносный пакет, а не частный.

Эксперимент, проведенный экспертом по информационной безопасности Алексом Бирсаном, продемонстрировал критическую уязвимость в способе обработки пакетов зависимостей при разработке программного обеспечения [6]. Бирсан обнаружил, что если пакет зависимостей, используемый приложением, существует как в общедоступном репозитории с открытым исходным кодом, так и в частном репозитории, общедоступный пакет в конечном итоге будет иметь приоритет, даже без каких-либо действий со стороны разработчика (рисунок 2).

Воспользовавшись этой уязвимостью, Бирсан успешно осуществил атаки на крупные компании, включая Microsoft, Apple, PayPal, Shopify, Netflix, Tesla, Yelp, Uber и другие.

Во избежания данной атаки, можно прибегнуть к рекомендации инженеров Microsoft [4, 15]. Необходимо защищать приватные пакеты с помощью контролируемых областей в публичных репозиториях, а также использовать верификацию на стороне клиента (закрепление версий, проверка целостности).

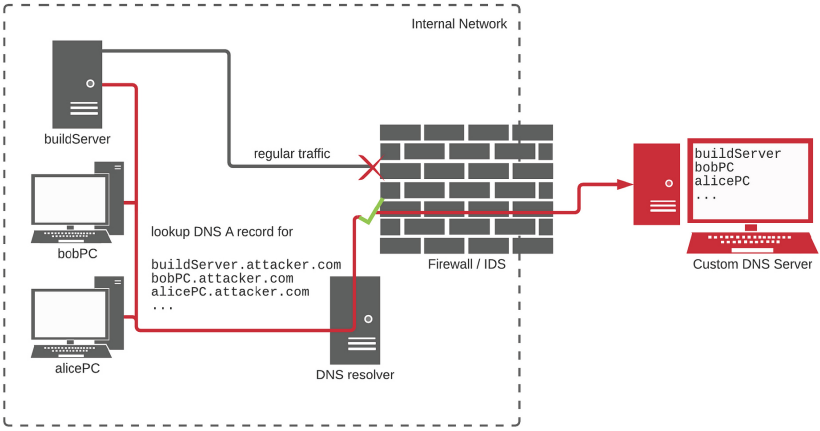

Рис. 2. Схема атаки на замещение зависимостей при помощи общедоступных репозиториев

Перехват зависимостей (Dependency hijacking) – получение контроля над учетной записью сопровождающего пакета в общедоступном репозитории с целью загрузки новой вредоносной версии широко используемого пакета с целью поставить под угрозу ничего не подозревающих клиентов, которые получают последнюю версию пакета.

Данная уязвимость особенно опасна для пакетов npm, где пакет с более высоким номером версии всегда имеет приоритет, независимо от его источника (рисунок 3).

Одно из эффективных решений данной проблемы предполагает обеспечение надежных процессов проверки для всех зависимостей. Этого можно достичь путем внедрения строгих мер контроля доступа и аутентификации для внутренних репозиториев, гарантируя, что только доверенные источники могут публиковать и обновлять пакеты. Кроме того, включение автоматизированных инструментов для сканирования и мониторинга зависимостей может помочь быстро обнаружить аномалии или несанкционированные изменения. Эти инструменты могут сравнивать исходный код и метаданные зависимостей с известными, надежными вер-

сиями, отмечая любые несоответствия для дальнейшей проверки. Поддерживая строгий процесс проверки, организации могут значительно снизить риск перехвата зависимостей.

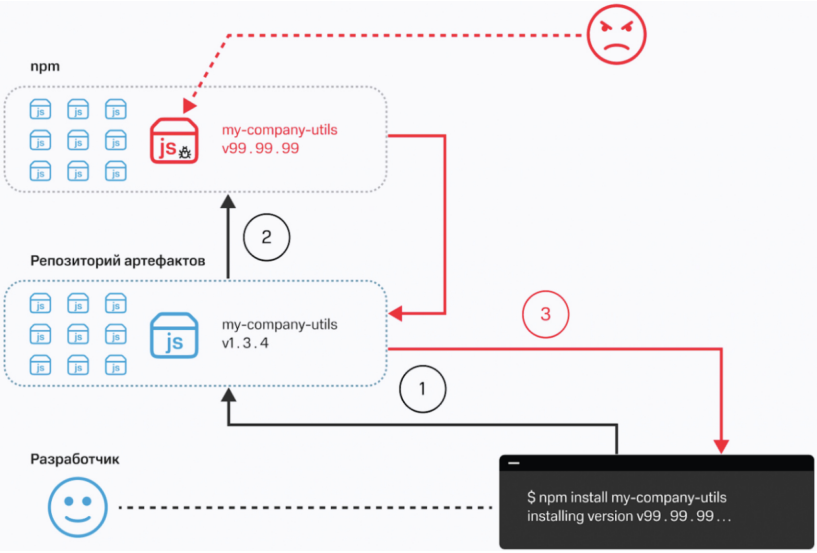

Рис. 3. Перехват зависимости при помощи увеличения версии пакета

Кибersквоттинг опечаток (Typosquatting) – публикация вредоносных пакетов с именами, похожими на названия популярных пакетов, в надежде, что разработчик допустит ошибку в написании имени пакета и случайно получит опечатанный пакет.

Данная атака может нанести обширный вред компаниям и разработчикам. Важно разберемся с некоторыми проблемами безопасности, возникающими в результате атак с использованием кибersквоттинга:

- Утечка данных. Вредоносное программное обеспечение может содержать код, предназначенный для кражи конфиденциальной информации – из личных записей или проприетарного исходного кода, – что приводит к утечке данных с разрушительными последствиями.

- Компрометация системы. Вредоносные пакеты могут включать в себя бэкдоры в изолированные среды, которые открывают двери для дальнейших эксплойтов и компрометаций.
- Репутационный ущерб. Организации, пострадавшие от опечаток пакетов, могут понести непоправимый репутационный ущерб, поскольку это подрывает доверие к их способности предоставлять безопасные программные среды.

Для решения проблемы кибersквоттинг опечаток зависимостей необходим многогранный подход, включающий расширенное управление пакетами, протоколы проверки и обучение пользователей.

Во-первых, внедрение автоматизированных систем обнаружения в репозиториях пакетов может снизить риск опечаток [8]. Эти системы используют алгоритмы машинного обучения и методы нечеткого сопоставления для идентификации и пометки пакетов с именами, похожими на популярные зависимости, что позволяет специалистам по обслуживанию репозитория просматривать и, при необходимости, удалять подозрительные пакеты. Кроме того, интеграция криптографической подписи пакетов гарантирует, что только проверенные и надежные источники смогут публиковать обновления. Разработчикам рекомендуется использовать функции менеджера пакетов, такие как блокировка зависимостей и контрольные суммы, чтобы гарантировать установку только нужных пакетов и минимизировать риск непреднамеренного включения вредоносных зависимостей.

Во-вторых, решающее значение имеет постоянный мониторинг и аудит зависимостей. Организации могут использовать инструменты управления зависимостями, которые регулярно сканируют уязвимости и проверяют целостность установленных пакетов на предмет известных инцидентов с опечатками. Эти инструменты можно настроить так, чтобы они предупреждали разработчиков при обнаружении несоответствий или потенциальных угроз. Более того, продвижение лучших практик, таких

как регулярные обновления и управление исправлениями, может значительно сократить окно возможностей злоумышленников для использования опечатываемых зависимостей.

Бренджекинг (Brandjacking) — публикация вредоносных пакетов способом, соответствующим соглашению об именах или другим характеристикам пакета конкретного бренда, в попытке заставить ничего не подозревающих разработчиков получить эти пакеты из-за ложной ассоциации их с доверенным брендом.

Данный метод атак похож на предыдущий за исключением, что здесь подменяются не имена пакетов, а используется замена имен авторов на имена популярных компаний и упор атаки идет на невнимательность или незнание разработчика. Все же это серьезная угроза кибербезопасности, может привести к широко распространенным нарушениям безопасности, поскольку ничего не подозревающие разработчики интегрируют «испорченные» пакеты в свои проекты.

Для борьбы с этим родом атак необходимо следующую стратегию: внедрение надежных процессов проверки пакетов (OSA, SCA) [5]. Объединив технические и образовательные меры, сообщество разработчиков программного обеспечения может значительно снизить количество случаев бренджекинга и защитить целостность своих проектов.

Исследование Mend показывает, что с января по сентябрь 2022 года наблюдается устойчивый ежеквартальный рост количества вредоносных пакетов, опубликованных в 2022 году, причем скачок со второго квартала на третий составил более 79 процентов [3, 7]. Каждый день на npm и Rubygems публиковалось не менее десяти вредоносных пакетов. В таблице 1 показано количество вредоносных пакетов, опубликованных за месяц, январь-октябрь 2022 г.

Таблица 1.

Количество вредоносных пакетов, опубликованных за месяц, январь-октябрь 2022 г.

Месяц публикации	Количество опубликованных вредоносных пакетов
Январь	525
Февраль	610
Март	1,605
Апрель	495
Май	500
Июнь	2,010
Июль	2,915
Август	1,660
Сентябрь	795
Октябрь	635

Также в таблице 2 можно увидеть интенсивность злоумышленников на атаки цепочек зависимостей в период с 2017 года по 2022 год. Интенсивность указывается в диапазоне от 1 до 5, где 1 малая активность, а 5 сильные удары.

Таблица 2.

Интенсивность атак в период с 2017 г. по 2022 г.

Год атаки	2017	2018	2019	2020	2021	2022
Typosquatting	1	2	2	6	5	5
Brandjacking	1	2	2	4	3	5
Dependency hijacking			1	2	2	2
Dependency confusion					2	2

Мы убедились, что подготовка, планирование и последовательное соблюдение передовых методов обеспечения безопасности методов CI/CD помогут организациям создать прочную основу кибербезопасности. Но поскольку активность угроз продолжает увеличиваться в объеме и инновациях, предприятиям необходимо выйти за рамки сегодняшнего статус-кво, чтобы выжить. Приложения – это источник жизненной силы глобальной экономики, и субъекты угроз знают об этом.

К счастью, мы наблюдаем растущую глобальную приверженность обеспечению кибербезопасности со стороны государственного сектора [12]. Правительства многих стран, в том числе России, США, Великобритании и Японии, ужесточают правила и стандарты для повышения безопасности во всей цепочке поставок программного обеспечения. Однако это всего лишь один шаг. Поскольку долг по безопасности для большинства продолжает расти, важно соблюдать методы безопасности и правильно расставлять приоритеты для уязвимостей, которые представляют наибольший риск. Организациям необходимо использовать инструменты определения приоритетов и устранения уязвимостей, которые больше всего повлияют на их системы и бизнес, если они хотят разумно управлять своим долгом по обеспечению безопасности.

Заключение

Процессы непрерывной интеграции и непрерывного развертывания (CI/CD) играют ключевую роль в современной разработке программного обеспечения [10, 11], предоставляя возможности для автоматизации и ускорения вывода продуктов на рынок. Однако, как показано в статье, с ростом популярности этих процессов увеличиваются и риски, связанные с эксплуатацией зависимостей. Уязвимости в цепочках поставок могут привести к серьезным последствиям, таким как компрометация системы, утечка данных и репутационные потери. Разработчикам и организациям необходимо уделять особое внимание безопасности на каждом этапе CI/CD, внедряя передовые методы защиты и контроля.

Мы определили методы минимизации рисков злоупотребления зависимостями, что важно внедрять многоуровневые меры безопасности, включая автоматизированные инструменты для мониторинга и анализа, строгий контроль доступа к репозиториям и использование криптографических методов для проверки целостности пакетов. Кроме того, регулярные аудиты и обучение

сотрудников помогают поддерживать высокий уровень безопасности и осведомленности о потенциальных угрозах.

В конечном итоге интеграция подходов, рассмотренных в статье, позволит не только обеспечить безопасность CI/CD процессов, но и повысить доверие к продуктам, создаваемым с их помощью, что в свою очередь укрепит позиции компании на рынке и обеспечит стабильность ее программных решений.

Список литературы

1. Будзко В. И. Развитие систем высокой доступности с применением технологии «большие данные» // Системы высокой доступности. 2013. Т. 9, № 4. С. 003-011.
2. Хамитов Р. М. Цифровизация образования и ее аспекты // Современные проблемы науки и образования. 2021. № 3. С. 8. https://doi.org/10.17513/spno.30771
3. Data Attack Surface Report Steve Morgan, Editor-in-Chief Northport, N.Y. June 8, 2020. URL: https://cybersecurityventures.com/wp-content/uploads/2020/12/ArcserveDataReport2020.pdf (дата обращения: 19.05.2024).
4. Introducing Package Source Mapping. URL: https://devblogs.microsoft.com/nuget/introducing-package-source-mapping/ (дата обращения: 20.05.2023).
5. Software composition analysis. URL: https://en.wikipedia.org/wiki/Software_composition_analysis (дата обращения: 20.05.2023).
6. Dependency Confusion: How I Hacked Into Apple, Microsoft and Dozens of Other Companies. URL: https://medium.com/@alex.birsan/dependency-confusion-4a5d60fec610 (дата обращения: 20.05.2024).
7. Mend Research Snapshot: Malicious Packages. URL: https://www.mend.io/malicious-package-research/ (дата обращения: 21.05.2023).
8. Learning Data Visualization in Assessing Linguistic Competence in the International Baccalaureate / O. M. Shevchenko, Yu. V. Torkunova, A. E. Upshinskaya, T. V. Shorina // European Proceedings of Social and

Behavioural Sciences: Conference proceedings, Moscow, April 23-25, 2020. London: European Publisher, 2020. P. 1155-1164. https://doi.org/10.15405/epsbs.2020.11.03.122
9. The OWASP Foundation - OWASP Top 10 CI/CD Security Risks. URL: https://owasp.org/www-project-top-10-ci-cd-security-risks/ (дата обращения: 16.05.2024).
10. Дэвис Д. Эффективный DevOps: искусство управления IT / Д. Дэвис, К. Дэниелс. СПб : O'Reilly Media, 2016. 118 с.
11. Sharma S. The DevOps Adoption Playbook: A Guide to Adopting DevOps in a Multi-Speed IT Enterprise. Wiley, 2017. 416 p.
12. Wilson G. DevSecOps: A leader's guide to producing secure software without compromising flow, feedback and continuous improvement. London: Rethink Press, 2020. 278 p.
13. Calvin S. P. Jenkins administrator's guide: Install, Manage and Scale a Ci/Cd Build and Re-lease System to Accelerate Your Product Lifecycle / S. P. Calvin, J. Humble, P. Debois. Boston, UK: IT Revolution Press, 2021. 436 p.
14. Безопасность разработки в Agile-проектах / Л. Белл, М. Брантон-Сполл, Р. Смит, Д. Бэрд. пер. с англ. А. А. Слинкин. М.: ДМК Пресс, 2018. 448 с.
15. Джозеф Д. Microsoft Windows Server / Д. Джозеф, Л. Дэвис. Вашингтон: Эком, 2018. 303 с.

References

1. Budzko V. I. Development of high availability systems using "big data" technology. *High Availability Systems,* 2013, vol. 9, no. 4, pp. 003-011.
2. Khamitov R. M. Digitalization of education and its aspects. *Modern problems of science and education*, 2021, no. 3, p. 8. https://doi.org/10.17513/spno.30771
3. Data Attack Surface Report Steve Morgan, Editor-in-Chief Northport, N.Y. June 8, 2020. URL: https://cybersecurityventures.com/wp-content/uploads/2020/12/ArcserveDataReport2020.pdf (accessed 19.05.2024).

4. Introducing Package Source Mapping. URL: https://devblogs.microsoft.com/nuget/introducing-package-source-mapping/ (accessed 20.05.2023).
5. Software composition analysis. URL: https://en.wikipedia.org/wiki/Software_composition_analysis (accessed 20.05.2023).
6. Dependency Confusion: How I Hacked Into Apple, Microsoft and Dozens of Other Companies. URL: https://medium.com/@alex.birsan/dependency-confusion-4a5d60fec610 (accessed 20.05.2024).
7. Mend Research Snapshot: Malicious Packages. URL: https://www.mend.io/malicious-package-research/ (accessed 21.05.2023).
8. Learning Data Visualization in Assessing Linguistic Competence in the International Baccalaureate / O. M. Shevchenko, Yu. V. Torkunova, A. E. Upshinskaya, T. V. Shorina. *European Proceedings of Social and Behavioral Sciences: Conference proceedings, Moscow, April 23-25, 2020*. London: European Publisher, 2020, pp. 1155-1164. https://doi.org/10.15405/epsbs.2020.11.03.122
9. The OWASP Foundation - OWASP Top 10 CI/CD Security Risks. URL: https://owasp.org/www-project-top-10-ci-cd-security-risks/ (accessed 16.05.2024).
10. Davis D. *Effective DevOps: the art of IT management* / D. Davis, K. Daniels. SPb : O'Reilly Media, 2016, 118 p.
11. Sharma S. *The DevOps Adoption Playbook: A Guide to Adopting DevOps in a Multi-Speed IT Enterprise*. Wiley, 2017, 416 p.
12. Wilson G. *DevSecOps: A leader's guide to producing secure software without compromising flow, feedback and continuous improvement*. London: Rethink Press, 2020, 278 p.
13. Calvin S. P. *Jenkins administrator's guide: Install, Manage and Scale a Ci/Cd Build and Re-lease System to Accelerate Your Product Lifecycle* / S. P. Calvin, J. Humble, P. Debois. Boston, UK: IT Revolution Press, 2021, 436 p.
14. *Development security in Agile projects* / L. Bell, M. Brunton-Spoll, R. Smith, D. Baird. Moscow: DMK Press, 2018, 448 p.
15. Joseph D. *Microsoft Windows Server* / D. Joseph, L. Davis. Washington, DC: Ecom, 2018, 303 p.

ДАННЫЕ ОБ АВТОРЕ

Алексеев Данил Николаевич, студент кафедры «Информационные технологии и интеллектуальные системы»
ФГБОУ ВО «Казанский государственный энергетический университет»
ул. Красносельская, 51, г. Казань, 420066, Российская Федерация
danil.core7@gmail.com

Хамитов Ренат Минзашарифович, доцент кафедры «Информационные технологии и интеллектуальные системы» кандидат технических наук
ФГБОУ ВО «Казанский государственный энергетический университет»
ул. Красносельская, 51, г. Казань, 420066, Российская Федерация
hamitov@gmail.com

DATA ABOUT THE AUTHOR

Danil N. Alekseev, Student of the Department of Information Technologies and Intelligent Systems
Kazan State Power Engineering University
51, Krasnoselskaya Str., Kazan, 420066, Russian Federation
danil.core7@gmail.com

Renat M. Khamitov, Associate Professor «Information Technologies and Intelligent Systems», Candidate of Technical Sciences
Kazan State Power Engineering University
51, Krasnoselskaya Str., Kazan, 420066, Russian Federation
hamitov@gmail.com
SPIN-code: 7401-9166
ORCID: https://orcid.org/0000-0002-9949-4404
ResearcherID: ADQ-3954-2022
Scopus Author ID: 57222149321

Поступила 27.05.2024
После рецензирования 18.06.2024
Принята 05.07.2024

Received 27.05.2024
Revised 18.06.2024
Accepted 05.07.2024