НАУЧНЫЙ АСПЕКТ

na-journal.ru

2024

№4

ТОМ 44

УДК 001.8(082)

ББК 1
Н 34

Периодичность – 12 раз в год

Журнал зарегистрирован в Федеральной службе по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзор)

Свидетельство ПИ № ФС 77-84349

ISSN 2226-5694

Учредитель, главный редактор – Хасиятуллов Марат Габделахатович

Состав ред. коллегии представлен на сайте https://na-journal.ru

Адрес редакции:
443125, г. Казань, ул. Азата Аббасова, д. 21А, кв. 149

Издатель ООО «Аспект»

Адрес издательства:
443068, г. Самара, ул. Николая Панова, д. 16, оф. 34

Н 34 НАУЧНЫЙ АСПЕКТ № 4 2024. – Самара: Изд-во ООО «Аспект», 2024 . – Т44 . – 138 с.

Журнал «Научный аспект» является научным изданием и отражает результаты научной деятельности авторов по различным дисциплинам в области гуманитарных, естественных и технических наук.

УДК 001.8(082)
ББК 1

Почтовый адрес: 420100 г. Казань а/я 9
Официальный сайт: https://na-journal.ru
Электронная почта: public@na-journal.ru

Подписано к печати 20.05.2024
Дата выхода в свет 28.05.2024
Цена свободная

Бумага ксероксная. Печать оперативная. Заказ № .
Формат 60×84 /16. Объем 8,28 п.л. Тираж 100 экз.

Отпечатано в типографии «Куранты»
г. Казань, Сибирский тракт, 34к14, оф. 317, тел. +7 (843) 216-12-71

Содержание

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004

Сравнительный анализ протоколов передачи голоса через IP: SIP и WebRTC

Репетий Егор Олесевич

студент Санкт-Петербургского государственного университета телекоммуникаций имени профессора М. А. Бонч-Бруевича

Вистунов Степан Сергеевич

студент Санкт-Петербургского государственного университета телекоммуникаций имени профессора М. А. Бонч-Бруевича

***Аннотация**: В данной научной статье проводится анализ основных протоколов передачи голоса через IP, включая SIP (Session Initiation Protocol) и WebRTC (Web Real-Time Communication). Статья начинается с исторического обзора развития технологий IP-телефонии, акцентируя внимание на ключевых этапах эволюции и технических инновациях. Далее проводится сравнительный анализ протоколов, основываясь на таких критериях, как эффективность использования сетевых ресурсов, качество голосовой связи, безопасность передачи данных, и удобство интеграции с другими технологиями и платформами. Особое внимание уделено аспектам безопасности каждого из рассмотренных протоколов, включая механизмы аутентификации, шифрование данных и защиту от возможных сетевых атак.*

***Abstract**: This research paper analyzes the major voice over IP protocols, including SIP (Session Initiation Protocol) and WebRTC (Web Real-Time Communication). The paper begins with a historical overview of the development of IP telephony technologies, emphasizing key evolutionary milestones and technical innovations. Then a comparative analysis of protocols is carried out based on such criteria as efficiency of network resources utilization, quality of voice communication, security of data transmission, and ease of integration with other technologies and platforms. Special attention is given to the security aspects of each of the protocols reviewed, including authentication mechanisms, data encryption, and protection against possible network attacks.*

***Ключевые слова**: IP-телефония, протоколы передачи голоса, SIP (Session Initiation Protocol), WebRTC (Web Real-Time Communication), качество голосовой связи, безопас-*

ность передачи данных, анализ производительности, кросс-платформенная интеграция, цифровые коммуникации, инновации в телекоммуникациях.

***Keywords**: IP telephony, voice protocols, SIP (Session Initiation Protocol), WebRTC (Web Real-Time Communication), voice quality, data transmission security, performance analysis, cross-platform integration, digital communications, innovation in telecommunications.*

Введение

С момента своего зарождения в начале 1990-х годов, IP-телефония прошла долгий путь развития от примитивных попыток передачи голоса по сети интернет до становления ведущей технологии в области глобальных коммуникаций. Первоначально, ограниченные скорости передачи данных и высокие требования к задержкам и потерям пакетов ставили под сомнение практическую целесообразность IP-телефонии. Однако, с улучшением инфраструктуры интернета и развитием алгоритмов сжатия голоса, IP-телефония начала демонстрировать значительные преимущества перед традиционной телефонией.

Значительный вклад в развитие IP-телефонии внесли такие протоколы, как H.323 и SIP (Session Initiation Protocol). H.323 был одним из первых стандартов, обеспечивших передачу голоса, видео и данных через IP-сети, но со временем SIP стал более популярным благодаря своей гибкости и легкости интеграции с другими системами и технологиями.

Последующие годы принесли инновации, такие как протокол IAX (Inter-Asterisk eXchange), разработанный для оптимизации связи между серверами Asterisk, а также появление WebRTC, который революционизировал понятие коммуникаций в реальном времени, позволяя прямую передачу голоса, видео и произвольных данных между браузерами без необходимости установки дополнительных плагинов или приложений.

Сегодня IP-телефония является неотъемлемой частью мировой телекоммуникационной инфраструктуры, обеспечивая эффективную и экономичную связь для бизнеса и частных пользователей по всему миру. Настоящая статья стремится проанализировать и сравнить ключевые протоколы, лежащие в основе IP-телефонии, с особым вниманием к их производительности, безопасности и способности к интеграции, что

позволит лучше понять текущее состояние технологии и перспективы её дальнейшего развития.

Сравнительный анализ протоколов передачи голоса через IP

Сравнительный анализ протоколов передачи голоса через IP, таких как SIP (Session Initiation Protocol) и WebRTC (Web Real-Time Communication), позволяет глубже понять их особенности и применимость в различных сценариях IP-телефонии. Для анализа выбраны ключевые критерии: эффективность использования сетевых ресурсов, качество голосовой связи, безопасность передачи данных и удобство интеграции с другими технологиями и платформами.

Таблица 1. **Сравнительный анализ протоколов передачи голоса через IP: SIP и WebRTC**

Критерий	SIP	WebRTC
Эффективность использования сетевых ресурсов	Эффективно использует сетевые ресурсы, но может увеличить нагрузку при масштабировании из-за необходимости в медиа-серверах.	Высокая эффективность за счет использования P2P-технологий, минимизирует нагрузку на серверы.
Качество голосовой связи	Зависит от выбранного кодека и условий сети. Поддержка различных кодеков позволяет адаптироваться к изменениям.	Поддерживает адаптивные кодеки, автоматически адаптируется к условиям сети, обеспечивая высокое качество связи.
Безопасность передачи данных	Зависит от применения механизмов шифрования и аутентификации (TLS, SRTP), требует настройки.	Встроенное шифрование всех передаваемых данных по умолчанию, один из самых безопасных протоколов для передачи данных.
Удобство интеграции с другими технологиями и платформами	Гибкий и универсальный, требует более глубоких знаний для интеграции.	Прост в интеграции с веб-технологиями, не требует дополнительных плагинов или программ для веб-приложений.

Данная таблица наглядно демонстрирует различия между протоколами SIP и WebRTC по ключевым критериям, что помогает сделать обоснованный выбор в зависимости от специфических требований проекта.

Аспекты безопасности протоколов передачи голоса через IP

Аспекты безопасности протоколов передачи голоса через IP, в частности SIP и WebRTC, играют критически важную роль в обеспечении конфиденциальности и защиты данных в современных коммуникационных системах. Оба протокола предлагают комплексные механизмы для обеспечения безопасности на разных уровнях, от аутентификации пользователей до шифрования данных и защиты от сетевых атак.

Для аутентификации пользователей и устройств SIP использует механизмы, основанные на стандартах HTTP Digest Authentication, предусматривающие проверку подлинности посредством логина и пароля. Это обеспечивает первоначальный уровень безопасности, хотя и требует дополнительного внимания к защите учетных данных от перехвата.

WebRTC, с другой стороны, встраивает аутентификацию прямо в свою архитектуру, используя современные методы для создания безопасных соединений. Поскольку WebRTC обычно работает через браузеры, оно полагается на существующие механизмы аутентификации веб-платформ, такие как OAuth, обеспечивая уровень безопасности, соответствующий современным веб-стандартам.

Шифрование данных является еще одним важным аспектом безопасности. SIP поддерживает шифрование сигнальных и медиа-потоков с использованием протоколов, таких как TLS для сигнализации и SRTP для медиапотоков, что обеспечивает защиту данных от перехвата и прослушивания. Однако, правильное и последовательное применение этих механизмов зависит от конфигурации и политик безопасности конечных пользователей.

WebRTC интегрирует шифрование на всех уровнях коммуникации по умолчанию, используя DTLS для шифрования всех передаваемых данных и SRTP для медиа-потоков. Это гарантирует, что любые данные, передаваемые через WebRTC, защищены от внешних угроз, включая перехват и модификацию.

Защита от сетевых атак для обоих протоколов включает в себя механизмы для смягчения последствий таких угроз, как DDoS-атаки, атаки типа «человек посередине» и другие виды эксплойтов. В то время как SIP может требовать дополнительных мер безопасности на уровне сети и приложений, WebRTC полагается на встроенную защиту, предоставляемую браузерами и поддерживаемыми протоколами, что упрощает обеспечение безопасности на всех этапах коммуникации.

Заключение

В заключении, оба протокола играют важную роль в развитии IP-телефонии, предлагая решения для различных потребностей пользователей и бизнеса. Выбор между SIP и WebRTC должен базироваться на специфических требованиях проекта, учитывая такие факторы, как масштабируемость, безопасность, качество связи и интеграционные возможности. В конечном счете, сравнительный анализ этих протоколов подчеркивает их вклад в обеспечение эффективной, безопасной и качественной голосовой коммуникации через IP, открывая новые горизонты для инноваций в области телекоммуникаций.

Список литературы

1. Абрамова Е.А., Красов А.В., Поляничева А. В. Тенденции развития и безопасность IP-телефонии // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2023). Сборник научных статей. XII Международная научно-техническая и научно-методическая конференция. В 4 т.. Санкт-Петербург, 2023. С. 23–28.
2. Богомаз М.Э., Михайлова Л.А., Поляничева А. В. Инструменты обеспечения безопасности IP-телефонии // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2022). XI Международная научно-техническая и научно-методическая конференция. Санкт-Петербург, 2022. С. 170–172.
3. Макарова А.К., Поляничева А.В., Саматова К. А. Анализ уязвимостей оборудования передачи голосового трафика // Актуальные проблемы

инфотелекоммуникаций в науке и образовании (АПИНО 2022). XI Международная научно-техническая и научно-методическая конференция. Санкт-Петербург, 2022. С. 665–669.
4. Волкогонов В.Н., Поляничева А.В., Орлов Г.А., Кононов П. А. Программа мониторинга качества голосовой связи для IP-АТС Elastix // Свидетельство о регистрации программы для ЭВМ RU 2021681591, 23.12.2021. Заявка № 2021680233 от 08.12.2021.
5. Петрова Т.В., Ковцур М.М., Карельский П.В., Поляничева А. В. Подходы обнаружения беспроводной точки доступа злоумышленника в локальной вычислительной сети // Региональная информатика (РИ-2022). Юбилейная XVIII Санкт-Петербургская международная конференция. Материалы конференции. Санкт-Петербург, 2022. С. 572–573.

УДК 004.56

Моделирование поведения вирусов в программном обеспечении для предотвращения кибератак

Репетий Егор Олесевич

студент Санкт-Петербургского государственного университета телекоммуникаций имени профессора М. А. Бонч-Бруевича

Хохлова Екатерина Игоревна

студентка Российского государственного университета нефти и газа (национального исследовательского института) имени И. М. Губкина

***Аннотация**: В данной научной статье рассматривается применение методов моделирования для анализа и предотвращения кибератак, вызванных вирусами в программном обеспечении. Делается акцент на разработке комплексных математических и компьютерных моделей, которые могут предсказывать поведение вирусов на основе анализа их прошлых и текущих активностей. Эти модели предназначены для помощи специалистам по кибербезопасности в разработке эффективных стратегий защиты и реагирования на угрозы. Статья начинается с обзора основных типов вирусов, их*

методов распространения и воздействия на информационные системы. Далее, описывается методология построения моделей, включая алгоритмы машинного обучения и нейронные сети, для идентификации потенциального вредоносного поведения в программном обеспечении.

***Abstract**: This research paper discusses the application of modeling techniques to analyze and prevent cyberattacks caused by viruses in software. The development of comprehensive mathematical and computer models that can predict the behavior of viruses based on analysis of their past and current activity is emphasized. These models are intended to help cybersecurity professionals develop effective defense and threat response strategies. The paper begins with an overview of the main types of viruses, their methods of propagation and their impact on information systems. Next, a methodology for building models, including machine learning algorithms and neural networks, to identify potential malicious behavior in software is described.*

***Ключевые слова**: моделирование поведения вирусов, кибербезопасность, программное обеспечение, предотвращение кибератак, компьютерные модели, машинное обучение, нейронные сети, анализ вредоносного ПО, стратегии защиты в киберпространстве.*

***Keywords**: virus behaviour modelling, cybersecurity, software, cyber attack prevention, computer models, machine learning, neural networks, malware analysis, cyber defence strategies.*

Введение

В современном мире, где цифровизация проникает во все аспекты нашей жизни, вопросы кибербезопасности становятся всё более актуальными. Среди множества угроз в киберпространстве вирусы в программном обеспечении занимают особое место из-за их способности быстро распространяться и причинять значительный ущерб информационным системам. Поэтому разработка методов для эффективного предотвращения кибератак, вызванных вирусами, является критически важной задачей.

Одним из перспективных направлений в решении этой проблемы является моделирование поведения вирусов в программном обеспечении. Моделирование предоставляет возможность анализировать механизмы распространения вирусов, их поведение и воздействие на информационные системы, что позволяет разрабатывать более эффективные способы защиты и механизмы реагирования на угрозы.

Основные типы вирусов

Вирусы в программном обеспечении представляют собой вредоносные программы, созданные с целью внедрения в систему без ведома пользователя, выполнения нежелательных действий, таких как кража данных, разрушение информации или подрыв функциональности системы. Основные типы вирусов, их методы распространения и воздействия на информационные системы включают:

Компьютерные вирусы

Это программы, способные копировать себя и заражать другие программы или файлы. Они могут быть прикреплены к исполняемым файлам и активироваться при запуске зараженной программы, распространяясь через сменные носители, сетевые папки или электронную почту.

Черви

Черви самостоятельно распространяются через сети, не требуя действий со стороны пользователя для своего исполнения. Они могут эксплуатировать уязвимости в программном обеспечении или операционной системе, чтобы автоматически копироваться на другие компьютеры, часто приводя к перегрузке сетевых ресурсов.

Троянские программы (Трояны)

Эти вирусы маскируются под легитимное ПО, но при запуске выполняют вредоносные действия. Трояны не могут самостоятельно распространяться и требуют, чтобы пользователи сами установили зараженное ПО, часто под предлогом полезной функциональности.

Руткиты

Руткиты предназначены для скрытия или маскировки вредоносной активности в зараженной системе, позволяя атакующему долго оставаться незамеченным. Они могут скрывать определенные файлы, процессы, системные данные или даже другие виды вирусов.

Спайвэр (Шпионское ПО)

Спайвэр предназначен для сбора информации о пользователе или организации без их согласия. Эти программы могут отслеживать историю браузера, клавиатурный ввод, доступ к электронной почте и другую личную информацию.

Методы распространения

Вирусы могут распространяться различными способами, включая электронную почту, сетевые диски, сменные носители, загрузки из Интернета и социальную инженерию. Они также могут эксплуатировать уязвимости в ПО для автоматического распространения без ведома пользователя.

Воздействие на информационные системы

Вирусы могут приводить к различным видам вреда, включая уничтожение данных, кражу конфиденциальной информации, несанкционированный доступ к системным ресурсам, замедление работы компьютера или сети, а также использование зараженных систем для проведения DDoS-атак или распространения спама.

Методология построения моделей для идентификации потенциального вредоносного поведения в ПО

Методология построения моделей для идентификации потенциального вредоносного поведения в программном обеспечении является ключевым компонентом в сфере кибербезопасности. Этот процесс включает в себя несколько этапов, от сбора и предобработки данных до обучения моделей и их валидации. В основе методологии лежит использование алгоритмов машинного обучения и нейронных сетей, которые позволяют автоматизировать обнаружение угроз и улучшить защиту информационных систем.

Сбор и предобработка данных

Первый шаг заключается в сборе достаточного количества данных о нормальном и вредоносном поведении программ. Данные могут включать логи системных вызовов, сетевой трафик, образцы кода программ и другие параметры, которые могут указывать на вредоносную активность. После сбора данных следует их предобработка, которая включает очистку, нормализацию и возможно преобразование данных в формат, пригодный для машинного обучения.

Выбор признаков

Выбор наиболее информативных признаков является ключевым этапом, поскольку он напрямую влияет на эффективность обучения модели. Признаки должны отражать характеристики нормального и аномального поведения программы. Примеры признаков могут включать частоту определенных системных вызовов, аномальные изменения в системных файлах, необычный сетевой трафик и так далее.

Моделирование

Для идентификации вредоносного поведения широко используются различные алгоритмы машинного обучения и нейронные сети:

- Алгоритмы машинного обучения: к ним относятся решающие деревья, случайные леса, градиентный бустинг, наивный байесовский классификатор, логистическая регрессия и другие. Эти алгоритмы способны обучаться на данных и классифицировать поведение программ как нормальное или вредоносное.
- Нейронные сети: Глубокое обучение с использованием нейронных сетей, таких как сверточные нейронные сети (CNN) и рекуррентные нейронные сети (RNN), предоставляет мощные инструменты для анализа сложных и больших данных. Нейронные сети особенно эффективны в выявлении сложных паттернов и аномалий в данных.

Валидация и тестирование

После обучения модели необходимо провести их тестирование на отдельном наборе данных для оценки их эффективности и точности. Процесс валидации помогает определить, насколько хорошо модель справляется с идентификацией вредоносного поведения и насколько она устойчива к ошибкам.

Итеративное улучшение

На основе результатов тестирования модели могут быть доработаны для улучшения их точности и эффективности. Этот процесс является итеративным и продолжается до достижения удовлетворительных результатов.

Методология построения моделей для идентификации вредоносного поведения в программном обеспечении играет важную роль в современной кибербезопасности, предоставляя инструменты для более эффективного и автоматизированного обнаружения киберугроз.

Заключение

В заключение, моделирование поведения вирусов в программном обеспечении является обещающим направлением в предотвращении кибератак, которое, при правильном применении, может значительно уменьшить риски и повысить устойчивость информационных систем к вредоносным угрозам. Дальнейшие исследования и разработки в этой области будут способствовать созданию более совершенных и эффективных средств обнаружения и нейтрализации киберугроз, способствуя созданию безопасного цифрового будущего.

Список литературы

1. Кирилова К. С. и др. Проблема обезвреживания руткитов уровня ядрав системахспециальногоназначения//I-methods.—2020.—Т.12.—№ .3.—С. 2.

2. Штеренберг, С. И. Компьютерные вирусы / С. И. Штеренберг, А. В. Красов, А. Ю. Цветков. Том Часть 1. — Санкт-Петербург: Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича, 2015.— 63 с. — EDN CMMEML.
3. Андрианов В. И., Романов Г. Г., Штеренберг С. И. Экспертные системы в области информационной безопасности //Актуальные проблемы инфотелекоммуникаций в науке и образовании.— 2015. — С. 193–197.
4. Штеренберг С. И., Москальчук А. И., Красов А. В. Разработка сценариев безопасности для создания уязвимых виртуальных машин и изучения методов тестирования на проникновения–Информационные технологии и телекоммуникации, 2021 //Т.— 2021. — Т. 9. — С. 1–2.
5. Гельфанд А. М. и др. Оценка рисков и угроз безопасности в среде «Умный дом» //Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2020).— 2020. — С. 316–321.

УДК 004.56

Комплексный подход к минимизации ущерба от DDoS-атак

Репетий Егор Олесевич

студент Санкт-Петербургского государственного университета телекоммуникаций имени профессора М. А. Бонч-Бруевича

Хохлова Екатерина Игоревна

студентка Российского государственного университета нефти и газа (национального исследовательского института) имени И. М. Губкина

***Аннотация**: В данной научной статье рассматривается комплексный подход к минимизации ущерба от DDoS-атак, объединяющий технологические, административные и правовые аспекты защиты информационных систем. Исследуются различные методы и стратегии, которые могут быть применены на разных уровнях сетевой архитектуры и организационной структуры для предотвращения или снижения воздействия DDoS-атак. Статья начинается с анализа текущей ситуации в области кибербезопасности, с акцентом на увеличение количества и сложности DDoS-атак*

в последние годы. Рассматриваются основные типы DDoS-атак и их потенциальный ущерб для бизнеса и государственных организаций.

***Abstract**: This research paper discusses a comprehensive approach to minimizing damage from DDoS attacks, combining technological, administrative, and legal aspects of information systems protection. Various methods and strategies that can be applied at different levels of network architecture and organizational structure to prevent or reduce the impact of DDoS attacks are explored. The article begins by analyzing the current cybersecurity landscape, with a focus on the increasing number and sophistication of DDoS attacks in recent years. The main types of DDoS attacks and their potential damage to businesses and government organizations are discussed.*

***Ключевые слова**: DDoS-атаки, информационная безопасность, кибербезопасность, минимизация ущерба, защитная стратегия, фильтрация трафика, резервное копирование данных.*

***Keywords**: DDoS attacks, information security, cyber security, damage minimisation, defensive strategy, traffic filtering, data backup.*

Введение

В современном мире, где зависимость от интернет-технологий продолжает неуклонно расти, вопросы кибербезопасности занимают особое место в агенде организаций всех масштабов. Среди многочисленных угроз, DDoS-атаки (распределенные атаки типа «отказ в обслуживании») выделяются как один из самых серьезных вызовов, способных нарушить работу веб-сервисов, дестабилизировать информационные системы и привести к значительным финансовым потерям. Подобные атаки могут быть направлены на любую цель, от крупных корпораций до государственных структур, делая защиту от них критически важной задачей.

Традиционные методы борьбы с DDoS-атаками, такие как перенаправление трафика или блокировка IP-адресов, часто оказываются недостаточными для обеспечения адекватной защиты. В ответ на эти вызовы в научном сообществе и индустрии кибербезопасности активно разрабатываются комплексные подходы, способные не только эффективно справляться с текущими атаками, но и адаптироваться к постоянно меняющемуся ландшафту угроз.

DDoS-атаки в области кибербезопасности

Текущая ситуация в области кибербезопасности характеризуется растущей угрозой DDoS-атак, которые становятся всё более сложными и разнообразными. Эти атаки представляют собой скоординированные попытки злоумышленников нарушить работу целевых веб-сайтов, серверов или сетевой инфраструктуры, перегружая их запросами с множества различных источников. В последние годы наблюдается не только увеличение количества DDoS-атак, но и их эволюция в плане сложности и мощности.

Увеличение количества и сложности атак

Рост числа атак: По данным различных исследовательских организаций, количество DDoS-атак продолжает расти с каждым годом. Это обусловлено как увеличением числа устройств, подключенных к интернету, так и доступностью инструментов для проведения атак.

Сложность атак: Современные DDoS-атаки становятся всё более сложными, используя многоуровневые и многоцелевые стратегии для обхода традиционных средств защиты. Атаки могут одновременно нацеливаться на различные компоненты инфраструктуры, включая приложения, сетевые службы и физическое оборудование.

Изменения в мотивах и методах

- Мотивы: помимо прямого финансового выгода, такого как выкупы за прекращение атак, мотивы могут включать политические или идеологические цели, корпоративные споры и личные обиды. В некоторых случаях атаки проводятся для отвлечения внимания от других вредоносных действий.
- Доступность инструментов: Упрощение доступа к инструментам для проведения DDoS-атак, включая аренду ботнетов в даркнете, позволяет даже непрофессионалам инициировать мощные атаки.

Основные типы DDoS-атак и их потенциальный ущерб

DDoS-атаки (распределенные атаки типа «отказ в обслуживании») являются одной из наиболее распространенных и разрушительных угроз в сфере кибербезопасности. Они направлены на нарушение работы целевых веб-сайтов, серверов или сетевой инфраструктуры, вызывая перегрузку системы запросами, которые исходят из множества различных источников.

Таблица 1. **Основные типы DDoS-атак**

Тип атаки	Описание	Потенциальный ущерб
Атаки на уровне объема	Используют огромное количество трафика для перегрузки широкополосного соединения. Примеры: UDP-флуд, ICMP-флуд.	Простои в работе, финансовые потери, утрата клиентов из-за недоступности сервисов.
Атаки на уровне протокола	Направлены на слой протокола, используя его уязвимости для перегрузки системы. Примеры: SYN-флуд, Ping of Death, Smurf атаки.	Перегрузка серверов или сетевого оборудования, простои в работе, финансовые потери.
Атаки на уровне приложения	Направлены на приложения, часто используя мало трафика для достижения целей. Примеры: HTTP-флуд.	Замедление или сбой в работе приложений, репутационный ущерб, финансовые потери.

Данная таблица кратко иллюстрирует основные типы DDoS-атак, их методы и потенциальный ущерб для бизнеса и государственных организаций.

Минимизация потенциального ущерба от DDoS-атак

Минимизация потенциального ущерба от DDoS-атак — это комплексный процесс, включающий в себя несколько ключевых стратегий и технологических решений. Важно понимать, что полностью избежать атак невозможно, но можно значительно снизить их воздействие на инфраструктуру и обеспечить более быстрое восстановление после инцидентов. Вот несколько основных принципов:

Распределенная архитектура

Создание географически распределенной инфраструктуры помогает распределять нагрузку и уменьшает риск того, что одновременная атака затронет все системы. Это делает более сложным для атакующих полностью вывести из строя целевые сервисы.

Балансировка нагрузки

Применение балансировщиков нагрузки может помочь равномерно распределить трафик между серверами, предотвращая перегрузку отдельных узлов. Это увеличивает устойчивость системы к атакам, нацеленным на исчерпание ресурсов.

Анти-DDoS решения

Специализированные сервисы и оборудование, предназначенные для смягчения последствий DDoS-атак, могут эффективно фильтровать аномальный трафик, минимизируя его влияние на нормальную работу сервисов. Эти решения часто используют сложные алгоритмы для определения и блокировки атакующего трафика.

Облачные сервисы

Облачные платформы могут предложить дополнительные масштабируемые ресурсы для смягчения DDoS-атак. Их преимущество в том, что они могут быстро масштабироваться для обработки увеличенного трафика и предоставлять расширенные возможности для смягчения атак.

План реагирования на инциденты

Разработка и регулярное обновление плана реагирования на DDoS-атаки позволяет организации быстро и эффективно реагировать на инциденты. Важно включить в план процедуры идентификации атак, коммуникацию с командами и внешними партнерами, а также шаги по восстановлению работы после атак.

Обучение и осведомленность персонала

Повышение уровня осведомленности и обучение сотрудников помогает в распознавании признаков DDoS-атак и правильном реагировании на них. Важно, чтобы все члены команды понимали свою роль в процессе защиты от атак и восстановления после них.

Регулярное тестирование

Проведение регулярных тестов на проникновение и симуляции DDoS-атак помогает выявлять уязвимости и оценивать эффективность защитных мер. Это также позволяет командам практиковаться в действиях по реагированию на атаки и улучшать процедуры восстановления.

Применение этих стратегий и подходов в комплексе позволяет создать многоуровневую систему защиты, которая способна существенно снизить потенциальный ущерб от DDoS-атак и обеспечить более высокую устойчивость инфраструктуры к подобным угрозам.

Заключение

Таким образом, осознание важности и реализация комплексного подхода к минимизации ущерба от DDoS-атак не только защищает от потенциальных потерь, но и способствует устойчивому развитию и росту в условиях постоянно возрастающей киберугрозы. Это требует постоянного совершенствования, инвестиций в новейшие технологии и, самое главное, сотрудничества в рамках всей отрасли для обмена знаниями и лучшими практиками. В конечном итоге, стойкость к DDoS-атакам — это не однократное достижение, а непрерывный процесс обучения, адаптации и улучшения.

Список литературы

1. Зимин А. Е., Косов Н. А. Обеспечение информационной безопасности в процессе создания и использования программ для ЭВМ //Актуаль-

ные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2017).— 2017. — С. 343–348.
2. Кушнир Д. В., Скробов Д. В. Обеспечение безопасности в технологии блокчейн //Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2022).— 2022. — С. 642–648.
3. Петрова Т. В. и др. Подходы обнаружения беспроводной точки доступа злоумышленника в локальной вычислительной сети //Региональная информатика (РИ-2022).— 2022. — С. 572–573.
4. Ковцур М. М. и др. Исследование способов удаленного перехвата трафика в корпоративных сетях //Вестник Санкт-Петербургского государственного университета технологии и дизайна. Серия.— 2021. — Т. 1. — С. 68–75.
5. Бударный Г. С. и др. Разновидности нарушений безопасности и типовые атаки на операционную систему //Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2022).— 2022. — С. 406–411.

УДК 004.56

Эволюция Zero Trust: инновационные подходы к безопасности в эпоху цифровых угроз

Репетий Егор Олесевич

студент Санкт-Петербургского государственного университета телекоммуникаций имени профессора М. А. Бонч-Бруевича

***Аннотация**: В данной научной статье рассматривается концепция Zero Trust (Нулевое доверие) как фундаментальный подход к обеспечению информационной безопасности в условиях усиливающихся цифровых угроз. С увеличением числа кибератак и развитием облачных технологий и мобильного доступа, традиционные методы защиты оказываются недостаточно эффективными, что делает применение модели Zero Trust особенно важным. Данная модель основывается на идее, что безопасность должна обеспечиваться путем постоянной верификации всех запросов на доступ к ресурсам, независимо от их происхождения. Статья подробно анализирует основные аспекты и компоненты Zero Trust, включая механизмы идентификации и аутентификации*

пользователей. Обсуждаются как технологические, так и организационные аспекты внедрения данной модели в корпоративные системы.

***Abstract**: This research paper examines the concept of Zero Trust as a fundamental approach to information security in the face of increasing digital threats. With the increasing number of cyber-attacks and the development of cloud technologies and mobile access, traditional defense methods are proving to be insufficiently effective, making the application of the Zero Trust model particularly important. This model is based on the idea that security should be ensured by continuously verifying all resource access requests, regardless of their origin. The paper provides detailed analysis of the main aspects and components of Zero Trust, including user identification and authentication mechanisms. Both technological and organizational aspects of implementing this model in corporate systems are discussed.*

***Ключевые слова**: нулевое доверие, информационная безопасность, кибербезопасность, облачные технологии, аутентификация, идентификация пользователей, минимизация привилегий, верификация доступа, безопасность данных, противодействие киберугрозам.*

***Keywords**: Zero Trust, information security, cyber security, cloud technology, authentication, user identification, privilege minimisation, access verification, data security, countering cyber threats.*

1. Введение

В современном мире, где технологическое развитие стремительно ускоряется, цифровая безопасность становится важнейшей задачей для организаций всех масштабов. Рост числа кибератак, увеличение объема и разнообразия устройств, подключенных к корпоративным сетям, а также расширение использования облачных сервисов требуют переосмысления традиционных подходов к безопасности. В этих условиях особое значение приобретает концепция Zero Trust (Нулевое доверие), предложенная Джоном Киндервагом в 2010 году и представляющая собой фундаментальный подход к обеспечению информационной безопасности.

Концепция Zero Trust основана на предположении, что внутренние и внешние угрозы постоянно существуют, поэтому безопасность должна обеспечиваться не только на границах корпоративной сети, но и внутри неё. Суть подхода заключается в постоянной верификации всех запросов на доступ к ресурсам сети, независимо от их источника и местоположе-

ния. Это предполагает отказ от традиционной модели «доверия по умолчанию», где ресурсы внутри периметра считаются безопасными. Вместо этого каждый запрос обрабатывается как потенциально угрожающий, что требует его тщательной проверки.

Такой подход к безопасности включает строгую идентификацию и аутентификацию пользователей, минимизацию привилегий для минимизации рисков и использование современных технологий для контроля и анализа трафика в реальном времени. Zero Trust применяется не только к людям, но и к устройствам, приложениям и сетевым потокам, что позволяет создать многоуровневую защитную архитектуру.

Введение данной концепции стало важным шагом в борьбе с угрозами, поскольку позволяет организациям адаптироваться к постоянно меняющемуся ландшафту кибербезопасности и обеспечивать более надежную защиту своих цифровых активов. В данной статье будет рассмотрена эволюция подхода Zero Trust, анализ его ключевых компонентов и оценка его влияния на современные практики информационной безопасности.

2. Основные аспекты и компоненты Zero Trust

Концепция Zero Trust подразумевает строгий подход к безопасности, основанный на принципе «никому не доверяй, всё проверяй». Эта модель предусматривает несколько ключевых аспектов и компонентов, которые обеспечивают её эффективность:

1. Строгая идентификация и аутентификация пользователей

Один из главных компонентов Zero Trust — это механизмы идентификации и аутентификации. Это включает в себя:

- Многофакторная аутентификация (MFA): помимо традиционных паролей, система требует один или несколько дополнительных факторов подтверждения личности, таких как биометрия (отпечатки пальцев, распознавание лица), одноразовые пароли (OTP), или использование аппаратных токенов.

- Универсальное управление идентификацией (IAM): Платформы для управления идентификацией обеспечивают централизованное управление учётными записями пользователей, их ролями и доступом.

2. Минимизация привилегий

Доступ к ресурсам предоставляется на основе строгой необходимости. Это значит, что пользователи и устройства получают только те права доступа, которые абсолютно необходимы для выполнения их текущих задач.

- Принцип наименьших привилегий (PoLP): ограничивает доступ пользователей и программ к минимально необходимому уровню.
- Динамическое управление доступом: Права доступа могут меняться в зависимости от контекста, например, местоположения пользователя, устройства, используемого для доступа, и текущей сетевой безопасности.

3. Верификация и контроль доступа

- Сетевая сегментация: Разделение сетевых ресурсов на изолированные сегменты для уменьшения рисков распространения угроз внутри сети.
- Микросегментация: ещё более детализированное разделение сетей, которое позволяет контролировать восточно-западный трафик в центрах обработки данных и облачных средах.

4. Расширенная аналитика и мониторинг

- Продвинутые системы обнаружения и реагирования (EDR и NDR): отслеживают и анализируют поведение пользователей и устройств в сети для своевременного выявления и реагирования на подозрительные действия.
- Аналитика на основе искусственного интеллекта и машинного обучения: помогает в выявлении аномалий и предсказании угроз на основе анализа больших объемов данных.

Эти компоненты Zero Trust вместе создают многоуровневую защиту, которая постоянно проверяет и подтверждает каждый запрос на доступ

к ресурсам, обеспечивая тем самым более высокий уровень безопасности в современных информационных средах.

3. Внедрение модели Zero Trust в корпоративные системы

Внедрение модели Zero Trust в корпоративные системы требует комплексного подхода, который включает в себя изменения как в технологической инфраструктуре, так и в управленческих практиках организации.

На технологическом уровне, внедрение Zero Trust начинается с переосмысления архитектуры безопасности. Компании должны разработать систему, в которой каждый запрос на доступ к ресурсам проходит через многоуровневую систему проверок безопасности. Это означает интеграцию современных технологий идентификации и аутентификации, включая биометрию и криптографические методы. Также необходимо внедрение механизмов непрерывного контроля и анализа трафика, что позволяет оперативно реагировать на аномальные действия. Особое внимание уделяется защите данных: они должны быть защищены на всех уровнях, от хранения до передачи и обработки.

Организационные изменения также играют ключевую роль. Прежде всего, необходимо создать культуру безопасности среди сотрудников, которая подразумевает повышенное внимание к вопросам конфиденциальности и безопасности данных. Важно также пересмотреть политики доступа к ресурсам, обеспечивая их соответствие принципу минимально необходимых привилегий. Это потребует от отделов IT и безопасности тесного сотрудничества для разработки и внедрения новых процедур и стандартов, которые поддерживают динамичное управление доступом на основе контекста действий пользователя и текущих условий безопасности.

Таким образом, успешное внедрение Zero Trust требует глубокой интеграции технологий и корпоративной культуры, а также готовности к пересмотру традиционных подходов к безопасности на всех уровнях организации.

4. Заключение

В заключении, можно подчеркнуть, что Применение Zero Trust позволяет организациям не только адаптироваться к текущему уровню угроз, но и прогнозировать возможные риски, опережая потенциальные атаки благодаря непрерывной верификации всех элементов системы. Важным аспектом является интеграция этой модели на всех уровнях организации, от инфраструктуры до корпоративной культуры, что требует совместных усилий всех подразделений. Внедрение Zero Trust означает не только установление новых технологий, но и развитие нового подхода к обучению и поведению сотрудников, что в конечном итоге формирует более зрелую и осознанную среду в вопросах кибербезопасности. Таким образом, Zero Trust не только защищает, но и стимулирует бизнесы к инновациям и оперативной адаптации к постоянно меняющемуся цифровому ландшафту.

Список литературы

1. Гельфанд А. М. и др. Разработка модели распространения самомодифицирующегося кода в защищаемой информационной системе // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки.— 2018.— № . 8. — С. 91–97.
2. Фархуллина Л. Г. Применение принципа нулевого доверия // Информационные технологии обеспечения комплексной безопасности в цифровом обществе. Сборник материалов VI Всероссийской молодежной научно-практической конференции с международным участием. Уфа, 2023. С. 169–173.
3. Нгуен Ф.Х., Нгуен Т.А.Т., Зарипова Р. С. Zero Trust как инструмент защиты информационных активов компаний // Научно-технический вестник Поволжья. 2023. № 12. С. 656–658.
4. Суворов А. М., Цветков А. Ю. Исследование атак типа переполнение буфера в 64-х разрядных unix подобных операционных системах // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2018).— 2018. — С. 570–573.

5. Малинский С. В. Концепция безопасности zero trust: принципы и практика внедрения // Интеллектуальные транспортные системы. материалы Международной научно-практической конференции. Москва, 2022. С. 430–437.

УДК 004.56

Анализ эффективности Zero Trust в современных сетевых архитектурах

Репетий Егор Олесевич

студент Санкт-Петербургского государственного университета телекоммуникаций имени профессора М. А. Бонч-Бруевича

***Аннотация**: В данной научной статье осуществляется анализ эффективности применения модели Zero Trust в современных сетевых архитектурах. Обсуждается, как подход Zero Trust, который предполагает постоянную верификацию всех пользователей и устройств в сети независимо от их местоположения, способен улучшить безопасность сетевых систем. Рассматриваются различные аспекты этой модели, включая принципы минимальных привилегий, микросегментацию сети и стратегии шифрования, которые способствуют обеспечению высокого уровня защиты данных и систем.*

***Abstract**: This research paper analyzes the effectiveness of applying the Zero Trust model in modern network architectures. It discusses how the Zero Trust approach, which involves continuous verification of all users and devices on the network regardless of their location, can improve the security of networked systems. Various aspects of this model are examined, including principles of minimal privilege, network micro-segmentation, and encryption strategies that contribute to a high level of data and system security.*

***Ключевые слова**: нулевое доверие, сетевая архитектура, кибербезопасность, микросегментация, принципы минимальных привилегий, шифрование данных, верификация идентификации, управление доступом, соответствие нормативам, адаптация к угрозам.*

***Keywords**: Zero Trust, network architecture, cybersecurity, micro-segmentation, minimum privilege principles, data encryption, identity verification, access control, compliance, threat adaptation.*

1. Введение

В эпоху цифровизации, когда количество кибератак и угроз безопасности данных растёт с каждым днём, традиционные подходы к сетевой безопасности, основанные на защите периметра, оказываются неэффективными. В этом контексте модель Zero Trust представляет собой радикальный пересмотр подходов к обеспечению безопасности. Суть модели Zero Trust заключается в предположении, что угрозы могут исходить как извне, так и изнутри сети, что требует постоянной верификации всех пользователей и устройств, пытающихся получить доступ к ресурсам сети, независимо от их местоположения.

Zero Trust отходит от традиционной модели доверия к устройствам внутри корпоративной сети, предлагая строгую политику «никому не доверять, всё проверять». Это подразумевает необходимость проверки каждой попытки подключения к сети, каждого запроса на доступ к ресурсам, что значительно снижает риски, связанные с атаками изнутри и другими угрозами.

2. Усиление безопасности сетей через применение модели Zero Trust

Анализ эффективности модели Zero Trust в современных сетевых архитектурах выявляет значительные преимущества этого подхода для улучшения безопасности информационных систем. Zero Trust — это не просто технологическая мера, а стратегия, которая изменяет фундаментальные принципы доступа к сетевым ресурсам, предполагая, что внутренняя сеть не более безопасна, чем внешняя. Этот подход исходит из предположения, что угрозы могут возникать в любой точке сети, и требует верификации каждого запроса на доступ к ресурсам, независимо от местоположения источника запроса.

1. Принципы Zero Trust и их влияние на безопасность

1. Строгая идентификация и аутентификация: Zero Trust требует, чтобы каждый пользователь и устройство в сети были точно идентифициро-

ваны и аутентифицированы перед предоставлением доступа к ресурсам. Это уменьшает риски, связанные с внутренними угрозами и атаками на уязвимости аутентификации.

2. Минимальные привилегии: Доступ предоставляется строго в соответствии с необходимостью. Это означает, что пользователям и устройствам предоставляется только тот уровень доступа, который необходим для выполнения их конкретных задач. Такой подход существенно снижает потенциальный ущерб от возможных нарушений безопасности.
3. Микросегментация: Сеть делится на мелкие сегменты, каждый из которых контролируется и защищается отдельно. Это помогает ограничить масштаб возможного нарушения, так как атака на один сегмент не распространяется автоматически на другие.

2. Практическое применение и вызовы

Применение модели Zero Trust включает в себя пересмотр существующих сетевых архитектур, что может быть сложным и затратным. Организациям необходимо инвестировать в технологии идентификации и аутентификации, инструменты мониторинга и аналитики для непрерывной оценки сетевого трафика и поведения пользователей.

Кроме того, переход к Zero Trust требует культурных изменений в организации, так как сотрудники и управленцы должны понимать и поддерживать постоянную необходимость верификации и ограничений доступа. Это может встретить сопротивление, особенно если ранее компания полагалась на более открытые и менее ограничивающие подходы к сетевой безопасности.

Несмотря на вызовы, анализ показывает, что применение модели Zero Trust может значительно улучшить общую безопасность сетевых систем, уменьшая вероятность успешных атак и ограничивая потенциальный ущерб от нарушений безопасности. По мере того как киберугрозы становятся всё более сложными и изощренными, подходы, основанные на Zero Trust, становятся не просто желательными, но и необходимыми для защиты критически важных ресурсов.

3. Ключевые механизмы защиты в модели Zero Trust

Модель Zero Trust подходит к вопросам безопасности с предположением, что внутренние и внешние угрозы всегда присутствуют, и поэтому требует строгой проверки всех попыток доступа к ресурсам сети. Эта модель включает несколько ключевых аспектов, которые помогают обеспечить высокий уровень защиты данных и систем.

Принципы минимальных привилегий обеспечивают, что каждому пользователю или устройству предоставляется только тот уровень доступа к ресурсам, который абсолютно необходим для выполнения конкретных задач. Это значительно снижает риск возможного ущерба в случае компрометации аккаунта, так как злоумышленник получит доступ только к ограниченному набору ресурсов. Принципы минимальных привилегий также способствуют сокращению ошибок, вызванных человеческим фактором, и уменьшают возможности для внутренних угроз.

Микросегментация сети разделяет сеть на множество мелких, изолированных сегментов. Каждый сегмент может иметь свои собственные правила и политики безопасности, что позволяет более тщательно контролировать трафик и ограничивать доступ к важным данным. Микросегментация также помогает минимизировать последствия атак, поскольку ограничивает возможность распространения вредоносного ПО или атакующего по сети. Это создает дополнительные барьеры и уровни защиты, значительно увеличивая общую безопасность системы.

Стратегии шифрования играют критическую роль в защите данных, передаваемых через сеть или хранящихся в облачных и физических хранилищах. Шифрование обеспечивает, что даже в случае несанкционированного доступа, конфиденциальная информация останется недоступной без соответствующих ключей дешифрования. В контексте Zero Trust, шифрование применяется не только к данным «в покое» и «в движении», но и к данным «в использовании», что обеспечивает комплексную защиту на всех этапах обработки информации.

Использование этих аспектов в рамках модели Zero Trust позволяет организациям значительно повысить уровень защиты своих цифровых активов, обеспечивая надежное противодействие как внутренним, так

и внешним угрозам. Это создает устойчивую и гибкую инфраструктуру, способную адаптироваться к постоянно меняющимся условиям и требованиям современной цифровой среды.

4. Заключение

В заключении можно утверждать, что модель Zero Trust демонстрирует значительную эффективность в укреплении безопасности современных сетевых систем. Подход, основанный на принципе «никому не доверяй, всё проверяй», обеспечивает комплексную защиту данных и ресурсов в условиях угроз, становящихся всё более сложными и изощренными.

Применение стратегий, таких как минимальные привилегии, микросегментация и широкомасштабное шифрование, позволяет организациям значительно уменьшить риски несанкционированного доступа и утечек данных. Эти механизмы защиты способствуют созданию глубоко защищенной среды, где каждый запрос на доступ тщательно проверяется, а права доступа строго регулируются на основе актуальной необходимости.

Кроме того, внедрение Zero Trust способствует повышению общей культуры безопасности внутри организации, акцентируя внимание на постоянной вигилантности и ответственности за защиту критически важных активов. Это приводит к укреплению защитных мер на всех уровнях и способствует более глубокому пониманию потенциальных угроз и методов их нейтрализации.

Таким образом, Zero Trust является не только техническим решением, но и философией, которая может радикально изменить подходы к обеспечению безопасности в цифровую эпоху. Эта модель доказала свою ценность и эффективность, делая её важным элементом стратегии кибербезопасности для любой организации, стремящейся защитить свои данные в условиях постоянно возрастающих киберугроз.

Список литературы

1. Нгуен Ф.Х., Нгуен Т.А.Т., Зарипова Р. С. Zero Trust как инструмент защиты информационных активов компаний // Научно-технический вестник Поволжья. 2023. № 12. С. 656–658.

2. Суворов А. М., Цветков А. Ю. Исследование атак типа переполнение буфера в 64-х разрядных unix подобных операционных системах // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2018).— 2018. — С. 570–573.
3. Tian X., Song H. A Zero Trust method based on BLP and BIBA model // Proceedings — 2021 14th International Symposium on Computational Intelligence and Design, ISCID 2021. 14. 2021. С. 96–100.
4. Гельфанд А. М. и др. Разработка модели распространения самомодифицирующегося кода в защищаемой информационной системе // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки.— 2018.— № . 8. — С. 91–97.
5. Бударный Г. С. и др. Разновидности нарушений безопасности и типовые атаки на операционную систему //Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2022).— 2022. — С. 406–411.

УДК 004.56

Безопасность идентификации в модели Zero Trust: методы и технологии

Репетий Егор Олесевич

студент Санкт-Петербургского государственного университета телекоммуникаций имени профессора М. А. Бонч-Бруевича

***Аннотация**: В данной научной статье рассматриваются методы и технологии, используемые для обеспечения безопасности идентификации в рамках модели Zero Trust. Подход Zero Trust требует от организаций выполнения строгой идентификации и аутентификации всех пользователей и устройств, пытающихся получить доступ к сетевым ресурсам, независимо от их местоположения. Статья подробно описывает, как современные технологии, включая многофакторную аутентификацию, биометрию и поведенческую аналитику, могут укрепить процессы идентификации в согласии с принципами Zero Trust.*

***Abstract**: This research paper examines the methods and technologies used to provide identity security within the Zero Trust model. The Zero Trust approach requires organizations to*

perform strong identification and authentication of all users and devices attempting to access network resources, regardless of their location. This article details how modern technologies including multi-factor authentication, biometrics, and behavioral analytics can strengthen identity processes in concert with Zero Trust principles.

***Ключевые слова**: нулевое доверие, безопасность идентификации, многофакторная аутентификация, биометрическая аутентификация, поведенческая аналитика, контроль доступа, кибербезопасность, идентификационные технологии, защита данных, сетевая безопасность.*

***Keywords**: Zero Trust, identity security, multi-factor authentication, biometric authentication, behavioural analytics, access control, cybersecurity, identity technologies, data protection, network security.*

1. Введение

В современном мире, где киберугрозы постоянно эволюционируют и становятся всё более сложными, традиционные подходы к защите сетевых ресурсов оказываются недостаточными. Модель Zero Trust, которая принципиально меняет подход к сетевой безопасности, предполагает, что ни одно устройство или пользователь не должны быть доверены по умолчанию, даже если они находятся внутри защищённой корпоративной сети. Основным краеугольным камнем этой модели является безопасность идентификации, которая требует проверки каждого, кто пытается получить доступ к ресурсам.

Эффективная реализация модели Zero Trust требует применения передовых методов и технологий для идентификации и аутентификации, способных обеспечить высокий уровень безопасности. В данной статье рассматриваются различные аспекты безопасности идентификации, включая многофакторную аутентификацию, биометрические технологии и поведенческую аналитику, которые помогают минимизировать риски нарушения безопасности и обеспечивать эффективный контроль доступа.

2. Методы и технологии, используемые для обеспечения безопасности идентификации в рамках модели Zero Trust

Модель Zero Trust требует комплексного подхода к безопасности, особенно когда дело доходит до идентификации пользователей и устройств.

Основная идея заключается в том, что никому не следует доверять по умолчанию, даже если пользователь или устройство находится внутри сети. Это требует внедрения многоуровневых технологий и методов аутентификации и идентификации для обеспечения защиты данных и ресурсов. Ниже рассмотрены ключевые технологии и методы, применяемые в этой модели.

1. Многофакторная аутентификация (MFA)

MFA является краеугольным камнем в обеспечении безопасности идентификации в рамках модели Zero Trust. Этот метод требует от пользователя предоставления двух или более доказательств своей легитимности перед получением доступа к ресурсам. Эти факторы включают что-то, что знает пользователь (пароль), что-то, что имеет пользователь (токен или смартфон), и что-то, что является пользователем (биометрия).

2. Биометрическая аутентификация

Биометрические технологии, такие как сканеры отпечатков пальцев, распознавание лица или радужной оболочки глаза, обеспечивают уникальную форму аутентификации, которую трудно подделать. В контексте Zero Trust биометрия добавляет ещё один слой защиты, существенно уменьшая риск несанкционированного доступа.

3. Поведенческая аналитика

Поведенческая аналитика использует алгоритмы машинного обучения для анализа поведения пользователя в реальном времени и может выявлять отклонения, которые могут указывать на потенциальные угрозы или фрод. Например, необычный вход в систему из нового местоположения или необычное время может требовать дополнительной аутентификации.

4. Непрерывная аутентификация

В рамках модели Zero Trust аутентификация не является разовым событием. Непрерывная аутентификация предполагает, что системы безопасности постоянно проверяют статус пользователя во время его сессии. Это может включать периодическую проверку подлинности и автоматическое выход из системы при обнаружении подозрительной активности.

5. Зашифрованное соединение

Использование VPN и других технологий шифрования соединений, таких как TLS, помогает обеспечить, что данные, передаваемые между поль-

зователем и сетью, защищены от перехвата или подмены. В условиях Zero Trust любой трафик, внутренний или внешний, считается потенциально опасным и должен быть зашифрован.

Эти методы и технологии обеспечивают строгий контроль доступа и идентификации, что является критически важным для поддержания безопасности сети в рамках модели Zero Trust. Эффективное их использование позволяет организациям значительно уменьшить риски связанные с кибератаками и утечками данных.

3. Интеграция передовых технологий аутентификации в модель Zero Trust для усиления безопасности

В современном мире, где киберугрозы постоянно эволюционируют, принципы модели Zero Trust предоставляют надёжный подход к обеспечению безопасности в сетевых системах. В основе этой модели лежит идея, что безопасность должна поддерживаться на всех уровнях взаимодействия внутри сети, и каждый запрос на доступ к ресурсам должен тщательно проверяться.

Использование многофакторной аутентификации в рамках Zero Trust подразумевает, что для подтверждения личности требуется несколько доказательств, что значительно усложняет задачу потенциальным атакующим. Биометрические технологии дополняют этот процесс, предоставляя уникальные и трудно подделываемые методы идентификации, такие как отпечатки пальцев или распознавание лица. Такие методы в сочетании с поведенческой аналитикой, которая отслеживает действия пользователя на предмет необычной активности, создают мощную систему верификации.

Эта система работает непрерывно, адаптируясь к поведению пользователей и текущим угрозам, что позволяет обнаруживать и реагировать на потенциальные нарушения безопасности в реальном времени. Таким образом, интеграция этих технологий в архитектуру Zero Trust не только повышает уровень безопасности сетей, но и создаёт динамичную среду, где меры безопасности постоянно адаптируются к меняющемуся ландшафту угроз, обеспечивая тем самым защиту от самых сложных и изощрённых атак.

4. Заключение

В заключение, обеспечение безопасности идентификации в рамках модели Zero Trust представляет собой критически важный элемент для защиты современных сетевых архитектур от разнообразных и постоянно развивающихся киберугроз. Эта модель требует строгого подхода к проверке всех пользователей и устройств, стремясь минимизировать риски, связанные с несанкционированным доступом и утечками данных. Многофакторная аутентификация, биометрические технологии и поведенческая аналитика служат фундаментальными компонентами в строительстве надежной системы безопасности, которая не только защищает, но и адаптируется к новым угрозам.

Применение этих технологий в контексте Zero Trust позволяет создать глубокую защиту, которая укрепляется через постоянную верификацию и оценку действий внутри сети. Такой подход не только обеспечивает защиту от известных угроз, но и способствует быстрому выявлению и реагированию на аномальное поведение, что является ключом к противодействию современным атакам.

Список литературы

1. Небаева К. А. Разработка необнаруживаемых стегосистем для каналов с шумом //СПб.: СПбГУТ.— 2014. — Т. 176.
2. Штеренберг С. И. Анализ работы алгоритмов защиты информации на основе самомодифицирующегося кода с применением стеговложения //Наукоемкие технологии в космических исследованиях Земли.— 2016. — Т. 8.— № . 2. — С. 86–90.
3. Astakhova L. V. Zero Trust Model as a factor of influence on the information behavior of organization employees // Scientific and Technical Information Processing. 2022. Т. 49. № 1. С. 60–64.
4. Малинский С. В. Концепция безопасности Zero Trust: принципы и практика внедрения // Интеллектуальные транспортные системы. материалы Международной научно-практической конференции. Москва, 2022. С. 430–437.

5. Нгуен Ф.Х., Нгуен Т.А.Т., Зарипова Р. С. Zero Trust как инструмент защиты информационных активов компаний // Научно-технический вестник Поволжья. 2023. № 12. С. 656–658.

УДК 004.56

Микросегментация сети как ключевой элемент стратегии Zero Trust

Репетий Егор Олесевич

студент Санкт-Петербургского государственного университета телекоммуникаций имени профессора М. А. Бонч-Бруевича

***Аннотация**: В данной научной статье анализируется роль микросегментации сети в контексте стратегии Zero Trust. Микросегментация сети, позволяющая разделять сетевые ресурсы на мелкие, управляемые и защищённые сегменты, представляет собой эффективный способ минимизации внутренних и внешних угроз безопасности. Статья детализирует, как микросегментация улучшает контроль доступа к ресурсам, повышает видимость трафика в сети и обеспечивает более строгую защиту данных.*

***Abstract**: This research paper analyzes the role of network micro-segmentation in the context of Zero Trust strategy. Network micro-segmentation, which allows network resources to be divided into small, manageable, and secure segments, is an effective way to minimize internal and external security threats. This article details how micro-segmentation improves access control to resources, increases visibility into network traffic, and provides tighter data protection.*

***Ключевые слова**: микросегментация сети, нулевое доверие, контроль доступа, безопасность сети, атакующая поверхность, управление безопасностью, идентификация и аутентификация, авторизация, видимость сетевого трафика, защита данных.*

***Keywords**: network micro-segmentation, Zero Trust, access control, network security, attack surface, security management, identification and authentication, authorisation, network traffic visibility, data protection.*

1. Введение

В условиях современной киберугрозной среды, где нападения становятся всё более изощренными и разрушительными, традиционные под-

ходы к сетевой безопасности, основанные на защите периметра, уже не могут гарантировать надежную защиту. Модель Zero Trust, которая предполагает отсутствие внутреннего доверия и проверку каждого запроса на доступ, приобретает все большее значение. Одним из фундаментальных компонентов этой модели является микросегментация сети — методика, которая разделяет сеть на множество мелких, управляемых сегментов, каждый из которых защищается индивидуально.

Микросегментация сети обеспечивает строгий контроль доступа к ресурсам, позволяя минимизировать внутренние и внешние угрозы. Этот подход помогает ограничить латеральное перемещение потенциальных злоумышленников внутри сети, существенно снижая риски атак и утечек данных. Кроме того, микросегментация увеличивает видимость взаимодействий внутри сети, позволяя оперативно реагировать на подозрительную активность и предотвращать возможные нарушения безопасности.

2. Роль микросегментации сети в контексте стратегии Zero Trust

Микросегментация сети играет критически важную роль в реализации стратегии Zero Trust, подхода к безопасности, который не допускает автоматического доверия к активам или пользователям, вне зависимости от их местоположения внутри или вне корпоративной сети. Эта стратегия основана на предположении, что угрозы могут возникнуть в любом месте, и поэтому необходима строгая проверка всех попыток доступа к ресурсам сети. Микросегментация усиливает эту модель, предоставляя несколько ключевых преимуществ:

Ограниченный доступ: Микросегментация деликатно разделяет сеть на мелкие сегменты, каждый из которых имеет свои собственные контрольные и защитные меры. Это означает, что доступ к ресурсам в одном сегменте не предоставляет автоматически доступ к другим сегментам. Это ограничивает возможности злоумышленника перемещаться по сети и достигать ценных данных, если он получит доступ к одной части системы.

Улучшенная видимость и контроль: Разделение сети на микросегменты увеличивает видимость трафика и активности в каждом сегменте. Благо-

даря этому, администраторы сети могут более точно наблюдать за аномальными действиями, что критически важно для обнаружения и предотвращения внутренних угроз и нарушений.

Уменьшение атакующей поверхности: Микросегментация значительно уменьшает атакующую поверхность, делая более трудным для злоумышленников нахождение и эксплуатацию уязвимостей в более широком масштабе. Каждый сегмент сети может быть защищен согласно своей уникальной потребности в безопасности, что уменьшает риски для всей сети.

Сокращение последствий нарушений: В случае нарушения безопасности, микросегментация помогает ограничить ущерб, изолируя атаку в одном сегменте и предотвращая её распространение по другим частям сети. Это позволяет быстрее и эффективнее реагировать на инциденты, сокращая время простоя и минимизируя потери.

Гибкость и масштабируемость: Микросегментация предлагает гибкость в управлении безопасностью и масштабируемость защитных мер, позволяя организациям адаптироваться к изменениям в инфраструктуре или операционной среде без переработки всей сетевой архитектуры.

Таким образом, микросегментация сети является неотъемлемым элементом стратегии Zero Trust, обеспечивая глубокую защиту, улучшенный контроль и уменьшенные риски, что делает её идеальным решением для защиты современных цифровых предприятий от сложных киберугроз.

3. Усиление сетевой безопасности через микросегментацию в рамках модели Zero Trust

Микросегментация сети — это стратегия безопасности, которая дробит сеть на более мелкие, управляемые и защищаемые части, обеспечивая повышенный контроль доступа, лучшую видимость трафика и усиленную защиту данных. Давайте подробнее рассмотрим, как эти аспекты улучшают общую безопасность сетевой инфраструктуры.

1. Улучшенный контроль доступа

Микросегментация позволяет настроить конкретные политики безопасности для каждого отдельного сегмента сети. В традиционной сетевой архитектуре, как только пользователь получает доступ к сети, он потен-

циально может получить доступ к широкому спектру ресурсов. Микросегментация же ограничивает доступ пользователя только к ресурсам внутри того сегмента, к которому он имеет разрешение. Это минимизирует возможности злоумышленников, которые, даже получив доступ к одной части сети, не смогут свободно перемещаться по другим сегментам без дополнительной аутентификации и авторизации.

2. Повышенная видимость трафика

Разделение сети на меньшие части также позволяет более детально наблюдать за трафиком в каждом сегменте. Управляющие могут точно видеть, какие данные передаются, кто и когда их запросил, и могут быстро обнаруживать аномальную активность. Это значительно упрощает мониторинг сети и позволяет оперативно реагировать на подозрительные действия, ещё до того, как они приведут к серьезным нарушениям безопасности.

3. Строгая защита данных

Микросегментация защищает критически важные данные, ограничивая количество точек доступа к ним и уменьшая тем самым атакующую поверхность. Каждый сегмент может быть индивидуально защищён согласно специфике хранимых в нём данных. Например, сегменты, содержащие конфиденциальную информацию, могут иметь более строгие политики безопасности и дополнительные уровни аутентификации. Это не только ограничивает доступ к данным, но и минимизирует возможные ущербы в случае успешной кибератаки на один из сегментов.

4. Заключение

В заключение, микросегментация сети является неотъемлемым и критически важным элементом стратегии Zero Trust, предлагая мощный инструмент для укрепления общей безопасности сетевых архитектур. В мире, где угрозы безопасности становятся всё более сложными и изощренными, традиционные методы защиты периметра уже не способны эффективно справляться с задачей защиты критически важных данных. Микросегментация, предлагая детализированное разделение сети на управляемые и защищаемые сегменты, позволяет организациям значительно умень-

шить атакующую поверхность и предотвратить латеральное перемещение потенциальных злоумышленников.

Список литературы

1. Astakhova L. V. Zero Trust model as a factor of influence on the information behavior of organization employees // Scientific and Technical Information Processing. 2022. Т. 49. № 1. С. 60–64.
2. Андрианов В. И., Романов Г. Г., Штеренберг С. И. Экспертные системы в области информационной безопасности //Актуальные проблемы инфотелекоммуникаций в науке и образовании.— 2015. — С. 193–197.
3. Штеренберг С. И., Москальчук А. И., Красов А. В. Разработка сценариев безопасности для создания уязвимых виртуальных машин и изучения методов тестирования на проникновения–Информационные технологии и телекоммуникации, 2021 //Т.— 2021. — Т. 9. — С. 1–2.
4. Нгуен Ф.Х., Нгуен Т.А.Т., Зарипова Р.С Zero Trust как инструмент защиты информационных активов компаний // Научно-технический вестник Поволжья. 2023. № 12. С. 656–658.
5. Фархуллина Л. Г. Применение принципа нулевого доверия // Информационные технологии обеспечения комплексной безопасности в цифровом обществе. Сборник материалов VI Всероссийской молодежной научно-практической конференции с международным участием. Уфа, 2023. С. 169–173.

УДК 004.77

Роль RESTful API в интеграции современных веб-сервисов

Репетий Егор Олесевич

студент Санкт-Петербургского государственного университета телекоммуникаций имени профессора М. А. Бонч-Бруевича

Хохлова Екатерина Игоревна

студентка Российского государственного университета нефти и газа (национального исследовательского института) имени И. М. Губкина

***Аннотация**: В данной научной статье анализируется роль RESTful API в контексте интеграции современных веб-сервисов. Описывается, как принципы REST (Representational State Transfer) влияют на разработку и взаимодействие веб-приложений в условиях распределённых систем. Статья детально рассматривает аспекты, такие как масштабируемость, гибкость и безопасность, которые являются ключевыми при интеграции различных веб-сервисов. Основное внимание уделяется методам обеспечения эффективного взаимодействия между различными веб-платформами с помощью RESTful API, а также оценке преимуществ, которые эти методы предоставляют в контексте современной веб-разработки.*

***Abstract**: This research paper analyzes the role of RESTful APIs in the context of modern web services integration. It describes how REST (Representational State Transfer) principles affect the development and interaction of web applications in distributed systems environments. The paper elaborates on aspects such as scalability, flexibility and security that are key when integrating different web services. The focus is on techniques for enabling efficient interoperability between different web platforms using RESTful APIs, and an assessment of the benefits these techniques provide in the context of modern web development.*

***Ключевые слова**: RESTful API, веб-сервисы, интеграция систем, масштабируемость, гибкость, безопасность API, версионирование API, JSON, XML, разработка веб-приложений.*

***Keywords**: RESTful API, web services, system integration, scalability, flexibility, API security, API versioning, JSON, XML, web application development.*

1. Введение

В современной веб-разработке, где бизнес и технологии быстро развиваются, эффективная интеграция различных веб-сервисов играет ключевую роль. Одним из основных инструментов, облегчающих эту интеграцию, является RESTful API (Representational State Transfer API). Этот стиль архитектуры программного обеспечения предоставляет разработчикам гибкий, масштабируемый и интуитивно понятный способ связи между различными веб-сервисами.

RESTful API позволяет веб-сервисам обмениваться данными и функциональностью через стандартизированные запросы и ответы, что делает возможным их взаимодействие без необходимости знания внутренней реализации друг друга. Это обеспечивает важные преимущества, такие как независимость компонентов, возможность масштабирования и упрощение архитектуры системы. Используя стандарты HTTP для создания, чтения, обновления и удаления ресурсов, RESTful API устанавливает прозрачный и эффективный механизм общения между сервисами.

2. Влияние REST (Representational State Transfer) на разработку и взаимодействие веб-приложений

Принципы REST (Representational State Transfer) оказывают значительное влияние на разработку и взаимодействие веб-приложений, особенно в условиях распределённых систем. REST представляет собой архитектурный стиль, который использует стандарты и протоколы интернета, в частности HTTP, для создания взаимодействующих веб-сервисов. Вот как принципы REST влияют на разработку и интеграцию веб-приложений:

1. Безсостояние (Statelessness): В REST каждый запрос от клиента к серверу должен содержать всю информацию, необходимую серверу для понимания и выполнения запроса. Сервер не сохраняет никакого состояния клиента между запросами. Это упрощает архитектуру сервера и облегчает масштабирование, так как любой сервер может обрабатывать любой запрос в любое время.

2. Единообразие интерфейса (Uniform Interface): REST требует, чтобы интерфейс между клиентом и сервером был единообразным, что упрощает и обобщает взаимодействие через веб-приложения. Это достигается за счет использования стандартных HTTP-методов (GET, POST, PUT, DELETE и т.д.), что позволяет разным системам легко взаимодействовать друг с другом.
3. Разделение клиента и сервера: Принцип разделения клиентской и серверной части позволяет разрабатывать и масштабировать эти компоненты независимо друг от друга. Клиент не должен знать внутренние механизмы сервера, а сервер не заботится о пользовательском интерфейсе, что повышает гибкость системы.
4. Кеширование: REST поддерживает кеширование ответов. Ответы на запросы могут быть помечены как кешируемые или некешируемые, что позволяет клиентам переиспользовать полученные ранее данные и снижает нагрузку на сервер.
5. Система слоев (Layered System): Клиенты взаимодействуют с конечным сервером через интернет, не имея информации о промежуточных слоях. Это позволяет организовывать различные уровни безопасности, балансировки нагрузки или шифрования независимо от клиента, что увеличивает безопасность и надёжность системы.
6. Код по требованию (optional): REST позволяет расширять функциональность клиента за счет скачивания и выполнения кода в форме скриптов или апплетов, что может динамически изменять поведение клиента.

Эти принципы делают REST идеальным для создания распределённых систем, где разные веб-приложения и сервисы могут эффективно взаимодействовать, масштабироваться и развиваться независимо друг от друга, что обеспечивает их высокую доступность и производительность.

3. Ключевые аспекты интеграции различных веб-сервисов

Интеграция различных веб-сервисов требует внимания к таким аспектам, как масштабируемость, гибкость и безопасность, которые тесно переплетены и взаимозависимы в современной архитектуре веб-приложений.

Масштабируемость в контексте веб-сервисов означает способность системы адаптироваться к изменяющимся объемам запросов без снижения производительности. В идеале, система должна быть способна обрабатывать растущее количество запросов путем добавления ресурсов, что часто достигается через развертывание дополнительных серверов и балансировку нагрузки. Это обеспечивает непрерывную доступность и быстродействие сервиса, что критически важно для пользовательского опыта и надежности бизнес-операций.

Гибкость системы важна для поддержания ее актуальности и эффективности перед лицом постоянно меняющихся технологических и бизнес-требований. Веб-сервисы должны быть спроектированы таким образом, чтобы их можно было легко модифицировать, обновлять или расширять без значительного воздействия на существующую инфраструктуру. Это достигается через модульную архитектуру и использование общих интерфейсов, что позволяет компонентам системы независимо друг от друга эволюционировать и интегрироваться.

Безопасность — это, пожалуй, самый важный аспект, поскольку уязвимости в одном веб-сервисе могут подвергнуть риску всю интегрированную систему. Безопасное взаимодействие между веб-сервисами достигается через применение шифрования, безопасных аутентификационных протоколов и строгих политик доступа. Также важно регулярно проводить аудиты безопасности и тестирование на проникновение, чтобы обнаруживать и устранять потенциальные угрозы.

Таким образом, удачная интеграция веб-сервисов зависит от того, насколько хорошо система может масштабироваться в ответ на растущие запросы, насколько она гибка в обновлениях и модификациях, и как эффективно она защищена от внешних и внутренних угроз.

4. Методы обеспечения эффективного взаимодействия между различными веб-платформами с помощью RESTful API

RESTful API оказывает глубокое влияние на разработку и взаимодействие веб-платформ благодаря своему универсальному и модульному

подходу. В архитектуре распределенных систем, где масштабируемость и гибкость являются критически важными, RESTful API предлагает эффективный способ управления взаимодействиями между различными веб-сервисами. Этот подход упрощает интеграцию, облегчает расширение приложений с помощью добавления новых серверов и поддерживает безопасное взаимодействие через применение современных стандартов безопасности.

С использованием стандартных HTTP методов для доступа к ресурсам и обмена данными, RESTful API способствует созданию надежных и легко доступных сервисов. Такая структура поддерживает безсостоянийные операции, что устраняет зависимости между последовательными запросами и обеспечивает более высокую надежность системы. При этом модульная природа RESTful API позволяет легко адаптировать или расширять функционал системы, обеспечивая её развитие без значительных переработок существующих компонентов.

Благодаря универсальным URI для доступа к ресурсам, RESTful API обеспечивает унификацию и стандартизацию взаимодействия между разнообразными клиентскими приложениями, что значительно упрощает интеграцию новых сервисов и внешних платформ. Также простота подключения к стандартам безопасности, включая OAuth и HTTPS, позволяет разработчикам эффективно защищать данные и транзакции в рамках своих приложений.

Эти характеристики делают RESTful API важным инструментом для разработчиков, стремящихся к созданию адаптивных, масштабируемых и безопасных веб-сервисов, поддерживающих современные требования к взаимодействию и обмену данными.

5. Заключение

В заключении можно подчеркнуть, что RESTful API играет ключевую роль в современной веб-разработке, обеспечивая эффективную интеграцию и взаимодействие между различными веб-сервисами. Благодаря своей универсальности, масштабируемости и гибкости, RESTful API стал неотъемлемым инструментом для создания динамичных, расширяемых

и безопасных веб-платформ. Используя стандартизированные протоколы и методы, он позволяет разработчикам строить многослойные архитектуры, которые могут легко адаптироваться к изменяющимся бизнес-требованиям и технологическим условиям. Таким образом, RESTful API не просто улучшает процесс разработки и поддержки веб-приложений, но и способствует созданию более стабильной и инновационной цифровой среды.

Список литературы

1. Андрианов В. И., Романов Г. Г., Штеренберг С. И. Экспертные системы в области информационной безопасности //Актуальные проблемы инфотелекоммуникаций в науке и образовании.— 2015. — С. 193–197.
2. Чайкин М. О. Разработка веб-приложения для мониторинга параметров работы манипуляционных механизмов // International Journal of Open Information Technologies. 2023. Т. 11. № 2. С. 60–65.
3. Воронин В.В., Громов П. А. Проектирование и разработка RESTful API веб-сервера // Информационные технологии XXI века. Сборник научных трудов. Ответственный за выпуск Е. А. Шеленок. Хабаровск, 2015. С. 474–480.
4. Зимин А. Е., Косов Н. А. Обеспечение информационной безопасности в процессе создания и использования программ для ЭВМ //Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2017).— 2017. — С. 343–348.
5. Коновалов Г. Г. Создание сервиса сокращения ссылок на языке java с использованием фреймворка Spring 5 // Тенденции развития науки и образования. 2023. № 103–8. С. 159–161.

УДК 004.49

Эволюция компьютерных вирусов: история и будущие тенденции

Репетий Егор Олесевич

студент Санкт-Петербургского государственного университета телекоммуникаций имени профессора М. А. Бонч-Бруевича

Яковлева Арина Андреевна

студентка Санкт-Петербургского государственного экономического университета

***Аннотация**: В данной научной статье рассматривается развитие компьютерных вирусов с момента их появления до современных дней, а также анализируются предполагаемые тенденции их развития в будущем. Статья начинается с исторического обзора первых вирусов, описывая ключевые моменты, которые повлияли на их эволюцию и привели к возникновению современных видов вредоносного ПО. Далее описывается, как изменения в технологиях и методах кибератак влияют на разработку и распространение вирусов, а также рассматриваются основные факторы, способствующие их адаптации и выживанию в постоянно меняющемся цифровом ландшафте.*

***Abstract**: This research article reviews the development of computer viruses from their inception to the modern day, and analyzes the anticipated trends in their development in the future. The paper begins with a historical overview of the first viruses, describing the key points that influenced their evolution and led to the emergence of modern types of malware. It then describes how changes in technology and cyberattack techniques affect the development and spread of viruses, and examines the key factors that contribute to their adaptation and survival in the ever-changing digital landscape.*

***Ключевые слова**: компьютерные вирусы, история вирусов, вредоносное программное обеспечение, кибератаки, эволюция вирусов, антивирусные технологии, методы защиты, будущие тенденции в кибербезопасности, адаптация вирусов, изменения в цифровых технологиях.*

***Keywords**: computer viruses, virus history, malware, cyberattacks, virus evolution, anti-virus technologies, defence techniques, future trends in cybersecurity, virus adaptation, changes in digital technologies.*

1. Введение

С момента своего первого появления в начале 1970-х годов, компьютерные вирусы претерпели значительное развитие, превратившись из простых экспериментальных программ в сложные инструменты кибервойны, способные наносить серьезный ущерб как отдельным пользователям, так и целым корпорациям и государственным структурам. Это введение в научную статью направлено на анализ ключевых этапов эволюции вирусов, которые сформировали современный ландшафт киберугроз.

История компьютерных вирусов начинается с ранних экспериментов, таких как создание Creeper Virus, который распространялся в сети ARPANET, демонстрируя потенциал саморепликации программного кода. С тех пор мир стал свидетелем появления множества вирусов, каждый из которых усовершенствовал методы внедрения, скрытности и нанесения ущерба. От простых шуток и хулиганских программ к серьёзным вредоносным атакам, таким как ILOVEYOU и WannaCry, вирусы стали инструментом для достижения экономической выгоды, политического давления и кибершпионажа. Важной вехой в развитии вирусов стало их массовое распространение с появлением Интернета, когда возможности для инфекции и скорости распространения вредоносного ПО значительно возросли. Это потребовало новых подходов к кибербезопасности и спровоцировало разработку продвинутых антивирусных программ и механизмов защиты.

Современные компьютерные вирусы используют сложные методы для избежания обнаружения, включая полиморфизм и метаморфизм, что делает борьбу с ними ещё более трудоёмкой. В ответ на это сообщество по кибербезопасности развивает всё более сложные системы защиты, стремясь опережать киберпреступников на шаг вперёд.

2. Ключевые особенности первых вирусов

Первые компьютерные вирусы, появившиеся в 1970-х и 1980-х годах, были относительно простыми по своей структуре и целям. Несмотря на их ограниченные технические возможности по сравнению с современными вирусами, эти ранние вирусы заложили основу для развития вредоносного

ПО. Вот несколько ключевых особенностей, которые характеризуют первые компьютерные вирусы:

1. Самовоспроизведение: Одной из главных особенностей первых вирусов была их способность к саморепликации. Эти программы могли автоматически копировать себя на другие файлы, диски или компьютеры без ведома пользователя. Это позволяло вирусам быстро распространяться среди компьютерных систем.
2. Простота: Ранние вирусы были относительно просты в своем дизайне. Они обычно состояли из небольших фрагментов кода, которые могли изменять загрузочные секторы на дискетах или вставлять себя в исполняемые файлы.
3. Экспериментальный характер: многие из первых вирусов были созданы скорее из любопытства или в качестве эксперимента, а не с целью нанесения ущерба. Программисты тестировали границы новых технологий и исследовали, как можно манипулировать программным обеспечением и операционными системами.
4. Ограниченное распространение: Поскольку интернет еще не был широко распространен, первые вирусы передавались в основном через физические носители, такие как дискеты. Это ограничивало их способность к распространению по сравнению с современными вирусами, которые могут распространяться глобально в интернете.
5. Нанесение вреда: хотя большинство ранних вирусов не было разработано с намерением причинять вред, некоторые из них могли вызывать непреднамеренные проблемы, такие как потеря данных, перезагрузка системы или замедление работы компьютера.
6. Образовательный аспект: Ранние вирусы также играли ключевую роль в развитии области кибербезопасности. Они заставили ИТ-специалистов и разработчиков программного обеспечения серьезнее относиться к вопросам безопасности и разработать первые антивирусные программы и стратегии защиты.

Эти особенности ранних компьютерных вирусов заложили основу для понимания того, как вредоносные программы могут взаимодействовать с операционными системами и как важно развивать защитные механизмы для борьбы с киберугрозами.

3. Современные компьютерные вирусы и их особенности

Современные компьютерные вирусы значительно отличаются от своих предшественников по сложности и потенциальному ущербу, который они могут нанести. С развитием цифровых технологий и всемирной сети Интернет, вирусы стали более адаптивными и утончёнными, используя сложные методы для обхода защитных систем и достижения своих разрушительных целей.

Современные вирусы часто разрабатываются с конкретными мотивами, включая финансовую выгоду, шпионаж, саботаж или политическое влияние. Они могут быть направлены на кражу личных данных, паролей, банковской информации и других конфиденциальных данных. Кроме того, некоторые вирусы способны превращать заражённые машины в часть ботнетов, используемых для проведения масштабных DDoS-атак или распространения спама.

Вирусы сегодня могут внедряться и скрываться в системе с использованием различных техник маскировки, включая полиморфизм и метаморфизм, что позволяет им изменять свой код при каждом новом заражении, усложняя тем самым их обнаружение антивирусными программами. Эти вирусы также способны эксплуатировать уязвимости в программном обеспечении, даже если они были недавно обнаружены и ещё не закрыты (так называемые «нулевые дни»).

Интернет значительно упростил распространение вирусов, позволяя им заражать компьютеры по всему миру за считанные минуты. Это также способствует быстрому распространению новых вариантов вирусов, так как злоумышленники могут легко обмениваться информацией и методами атак.

Более того, современные вирусы становятся всё более специализированными и могут целенаправленно атаковать определённые типы инфраструктур или даже конкретные предприятия, что делает борьбу с ними особенно сложной. В ответ на это, специалисты по кибербезопасности непрерывно работают над улучшением защитных технологий и методов предотвращения атак, чтобы оставаться на шаг впереди киберпреступников и обеспечивать безопасность в цифровом мире.

4. Взаимосвязь между технологическими инновациями и адаптацией вирусов

Изменения в технологиях и методах кибератак имеют глубокое влияние на разработку и распространение вирусов, поскольку они тесно связаны с постоянно эволюционирующим цифровым ландшафтом. По мере того как технологии становятся более продвинутыми, возрастает и сложность кибератак, что требует от вирусов адаптации к новым защитным мерам и эксплуатации новых уязвимостей.

Одним из ключевых аспектов влияния технологических изменений на вирусы является увеличение количества подключенных устройств, включая мобильные устройства, IoT (интернет вещей) и облачные сервисы. Эти устройства часто имеют различные уровни безопасности, что создаёт новые возможности для вирусов. Например, IoT устройства, которые не всегда регулярно обновляются или поддерживаются, могут легко стать целями для вирусов, предоставляя киберпреступникам «заднюю дверь» в более защищённые сети.

С появлением облачных технологий вирусы также начали адаптироваться к тому, чтобы эффективно работать в этих средах. Облачные платформы могут предлагать масштабируемость и ресурсы, которые могут быть использованы вирусами для распределенных атак, таких как DDoS. Кроме того, вирусы могут нацеливаться на кражу данных из облачных хранилищ, эксплуатируя неправильно настроенные или уязвимые конфигурации.

Усовершенствования в методах кибератак также привели к тому, что современные вирусы стали использовать более сложные и маскировочные техники для обхода обнаружения. Использование полиморфизма и метаморфизма, когда вирусы меняют свой код при каждом выполнении, помогает им избегать традиционных антивирусных средств. Эти методы заставляют разработчиков антивирусного программного обеспечения пересматривать свои подходы, интегрируя более продвинутые системы обнаружения, основанные на поведении, а не только на сигнатурах.

5. Заключение

В заключение можно подчеркнуть, что адаптация и выживание компьютерных вирусов в постоянно изменяющемся цифровом ландшафте обусловлены рядом ключевых факторов. Первостепенное значение имеет технологическое развитие, которое открывает новые возможности для вирусов через увеличение количества подключаемых устройств и расширение облачных платформ. Это предоставляет вирусам более широкий спектр целей и более сложные среды для маскировки.

Кроме того, усовершенствования в методах кибератак позволяют вирусам эффективнее обходить существующие меры безопасности, используя полиморфные и метаморфные техники для избежания обнаружения антивирусными программами. Такие методы требуют от специалистов по кибербезопасности непрерывного обучения и адаптации к новым угрозам.

Важную роль играет также глобализация и всеобщая связность, которые облегчают распространение вирусов на международном уровне. Это требует координированных усилий на глобальном уровне для разработки и внедрения эффективных антивирусных решений и стратегий кибербезопасности.

Таким образом, сложность и постоянно растущая угроза компьютерных вирусов подчеркивают необходимость продолжительного и интенсивного взаимодействия между разработчиками технологий, специалистами по безопасности и законодательными органами для обеспечения защиты цифровой инфраструктуры в изменчивом мире.

Список литературы

1. Бударный Г. С. и др. Разновидности нарушений безопасности и типовые атаки на операционную систему //Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2022).— 2022. — С. 406–411.
2. Штеренберг С. И., Москальчук А. И., Красов А. В. Разработка сценариев безопасности для создания уязвимых виртуальных машин и из-

учения методов тестирования на проникновения–Информационные технологии и телекоммуникации, 2021 //Т.— 2021. — Т. 9. — С. 1–2.
3. Штеренберг, С. И. Компьютерные вирусы / С. И. Штеренберг, А. В. Красов, А. Ю. Цветков. Том Часть 1. — Санкт-Петербург: Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича, 2015.— 63 с. — EDN CMMEML.
4. Штеренберг С.И., Бударный Г.С., Чумаков И. В. Анализ безопасности доменных систем // Региональная информатика (РИ-2022). Юбилейная XVIII Санкт-Петербургская международная конференция. Материалы конференции. Санкт-Петербург, 2022. С. 587–588.
5. Штеренберг С.И., Морозов В.Е., Андрианов В. И. Технологии программной защиты в интернете // Часть 2. Санкт-Петербург, 2015.

УДК 004.77

Веб-ассемблер в действии: революция в производительности веб-приложений

Репетий Егор Олесевич

студент Санкт-Петербургского государственного университета телекоммуникаций имени профессора М. А. Бонч-Бруевича

***Аннотация**: В данной научной статье анализируется влияние технологии WebAssembly на производительность веб-приложений. WebAssembly представляет собой новый формат бинарного кода, который позволяет веб-приложениям выполняться с производительностью, сопоставимой с нативными приложениями. Это стало возможным благодаря предоставлению механизма для выполнения кода на стороне клиента в веб-браузере, который является более эффективным и безопасным по сравнению с традиционным JavaScript. Статья подробно рассматривает архитектуру WebAssembly, основные концепции и принципы работы этой технологии. Особое внимание уделяется возможностям WebAssembly в области ускорения вычислительных процессов и улучшения взаимодействия между браузером и пользовательским кодом.*

***Abstract**: This research paper analyzes the impact of WebAssembly technology on the performance of web applications. WebAssembly is a new binary code format that allows web applications to execute with performance comparable to native applications. This is made possible*

by providing a mechanism for executing client-side code in a web browser that is more efficient and secure than traditional JavaScript. This article takes a detailed look at the WebAssembly architecture, basic concepts and principles of this technology. Special attention is paid to WebAssembly's capabilities in accelerating computational processes and improving the interaction between the browser and the user code.

***Ключевые слова**: WebAssembly, производительность веб-приложений, бинарный код, веб-технологии, клиентская оптимизация, безопасность веб-приложений, игровые технологии, виртуальная реальность, дополненная реальность, изолированное выполнение кода.*

***Keywords**: WebAssembly, web application performance, binary code, web technologies, client-side optimisation, web application security, gaming technologies, virtual reality, augmented reality, isolated code execution.*

1. Введение

В современном мире цифровых технологий, где скорость и эффективность веб-приложений играют решающую роль, технология WebAssembly открывает новые горизонты для разработчиков и пользователей. WebAssembly, сокращённо Wasm, представляет собой бинарный формат инструкций для стековой виртуальной машины, разработанный для выполнения на страницах веб-браузеров. Эта технология решает проблему производительности, которая долгое время сдерживала веб как платформу для выполнения приложений, требующих интенсивных вычислений.

WebAssembly позволяет коду, написанному на языках программирования высокого уровня, таких как C++ и Rust, быть скомпилированным в бинарный формат, который выполняется в браузере с производительностью, сравнимой с нативными приложениями. Это означает, что сложные приложения, включая видеоигры, графические редакторы и даже виртуальную и дополненную реальность, теперь могут эффективно функционировать в стандартном веб-браузере без потери производительности.

Введение WebAssembly в экосистему веб-разработки представляет собой значительный шаг вперёд в понимании того, как можно улучшить взаимодействие пользователя с веб-приложениями. Отмеченный множеством преимуществ, включая уменьшение времени загрузки, увеличение скоро-

сти выполнения и улучшение возможностей безопасности, WebAssembly уже начал революционизировать подход к разработке веб-приложений. В данной статье рассматриваются ключевые аспекты этой технологии и её влияние на производительность веб-приложений, предоставляя читателю глубокий анализ потенциала и возможностей WebAssembly в современной веб-разработке.

2. Основные концепции и принципы работы WebAssembly

WebAssembly, или Wasm, представляет собой технологию, разработанную для выполнения кода на веб-страницах с производительностью, сравнимой с нативными приложениями. Эта технология предназначена для того, чтобы быть быстрой, безопасной, портативной и доступной в любом современном веб-браузере, независимо от используемой операционной системы.

Архитектура WebAssembly оптимизирована для быстрой загрузки, компиляции и выполнения, благодаря чему код на Wasm запускается почти со скоростью, сопоставимой с нативным кодом. Особенностью WebAssembly является его бинарный формат, который браузеры могут быстро анализировать и выполнить. Это отличается от традиционного JavaScript, который требует дополнительного времени для парсинга текста и его компиляции в машинный код.

Безопасность в WebAssembly достигается за счёт выполнения кода в изолированной песочнице в браузере, что предотвращает доступ к памяти устройства или к другим программам. Эта изоляция защищает пользователя от вредоносного кода и возможных уязвимостей.

WebAssembly также поддерживает код, написанный на различных языках программирования, таких как C++, Rust и других, что позволяет разработчикам использовать языки, наиболее подходящие для их задач, и компилировать их в Wasm для выполнения в браузере.

Интеграция с существующими веб-технологиями также является ключевым аспектом WebAssembly. Он может работать совместно с JavaScript, что позволяет разработчикам оптимизировать части своих приложений с использованием Wasm, сохраняя при этом гибкость и доступность JavaScript.

Таким образом, WebAssembly не просто предоставляет новый инструмент для веб-разработчиков, но и значительно расширяет возможности веб-приложений, делая их более мощными, быстрыми и безопасными.

3. Возможности WebAssembly в области ускорения вычислительных процессов и улучшения взаимодействия между браузером и пользовательским кодом

WebAssembly значительно улучшает производительность веб-приложений, ускоряя вычислительные процессы и улучшая взаимодействие между браузером и пользовательским кодом. Эта технология позволяет выполнение кода почти со скоростью нативных приложений, что открывает новые возможности для веб-разработки, особенно в областях, требующих интенсивных вычислений, таких как игры, видеообработка и научные вычисления.

За счет использования бинарного формата, который быстрее и эффективнее обрабатывается браузерами по сравнению с традиционным JavaScript, WebAssembly минимизирует время загрузки и время выполнения кода. Это улучшение производительности происходит благодаря тому, что WebAssembly оптимизируется на этапе компиляции, что позволяет браузерам быстро превращать бинарный код в машинный код, готовый к выполнению.

Помимо ускорения вычислений, WebAssembly также улучшает взаимодействие между браузером и пользовательским кодом. Он позволяет более тесно интегрировать код, написанный на различных языках программирования, с веб-платформами, благодаря чему разработчики могут лучше использовать возможности браузера и одновременно сохранять привычные и мощные подходы программирования.

Благодаря изоляции кода в песочнице, WebAssembly также гарантирует, что выполнение кода не только быстрое, но и безопасное. Это создает надежную среду для выполнения кода, который взаимодействует с браузером, обеспечивая защиту данных пользователя и системы.

В совокупности эти улучшения делают WebAssembly важным инструментом для разработчиков, стремящихся создать мощные и высокоэффек-

тивные веб-приложения, которые могут конкурировать по производительности с десктопными приложениями, обеспечивая при этом глобальную доступность и портативность, характерные для веб-технологий.

4. Заключение

В заключении можно подчеркнуть, что WebAssembly революционизировал подход к разработке веб-приложений, предоставляя значительные улучшения в производительности, которые ранее были недоступны в чисто JavaScript-средах. С его помощью разработчики могут теперь создавать приложения, которые выполняются с нативной скоростью, что открывает двери для более сложных и ресурсоёмких проектов, таких как игры, графические редакторы, и программы для видеообработки, прямо в браузере.

Технология WebAssembly позволяет использовать различные языки программирования, улучшая тем самым доступ к веб-разработке для широкого круга программистов. Это обеспечивает не только скорость и эффективность, но и безопасность, благодаря изоляции выполнения кода в браузере.

В свете этих преимуществ, WebAssembly значительно расширяет границы того, что возможно в вебе, предоставляя новые возможности для инноваций и развития. Таким образом, WebAssembly не просто улучшает существующие подходы к созданию веб-приложений, но и формирует основу для будущих инноваций в области веб-технологий.

Список литературы

1. Дикарев А.С., Гуренко В. В. Анализ технологии веб-ассемблирования для разработки высокоскоростных веб-приложений // Технологии инженерных и информационных систем. 2020. № 1. С. 55–63.
2. Красов А.В., Пестов И.Е., Алехин Р.В., Казанцев А. А. Программа управления IoT устройствами на базе облачных платформ // Свидетельство о регистрации программы для ЭВМ RU 2022613439, 14.03.2022. Заявка № 2022612581 от 24.02.2022.

3. Овчаров З. А. Методы оптимизации браузерных вычислений на основе распараллеливания потоков // Вестник Технологического университета. 2023. Т. 26. № 11. С. 205–209.
4. Красов А.В., Пешков А.И., Шариков П. И. Анализатор байт-кода java — программы для скрытого вложения цифрового водяного знака посредством автоматического редактирования байт-кода class-файла // Свидетельство о регистрации программы для ЭВМ RU 2020617872, 15.07.2020. Заявка № 2020616770 от 29.06.2020.
5. Красов А.В., Левин М.В., Цветков А. Ю. Метод управления трафиком в гибридной программно-определяемой сети // Информационные технологии и телекоммуникации. 2016. Т. 4. № 2. С. 53–63.

УДК 004.42

Принципы объектно-ориентированного программирования

Репетий Егор Олесевич

студент Санкт-Петербургского государственного университета телекоммуникаций имени профессора М. А. Бонч-Бруевича

Хохлова Екатерина Игоревна

студентка Российского государственного университета нефти и газа (национального исследовательского института) имени И. М. Губкина

***Аннотация**: В данной научной статье рассматриваются ключевые принципы объектно-ориентированного программирования (ООП) и их практическое применение в современной разработке программного обеспечения. Освещаются такие фундаментальные концепции ООП, как инкапсуляция, наследование и полиморфизм, и объясняется, как эти принципы способствуют созданию модульного, масштабируемого и легко поддерживаемого кода. Статья детально анализирует, как использование этих принципов может улучшить качество программного продукта и упростить процесс разработки за счёт предоставления чёткой структуры и сокращения дублирования кода.*

***Abstract**: This research paper discusses the key principles of object-oriented programming (OOP) and their practical application in modern software development. Fundamental OOP*

concepts such as encapsulation, inheritance, and polymorphism are highlighted and the way these principles contribute to the creation of modular, scalable, and easily maintainable code is explained. The article analyzes in detail how the use of these principles can improve the quality of a software product and simplify the development process by providing a clear structure and reducing code duplication.

***Ключевые слова**: объектно-ориентированное программирование (ООП), инкапсуляция, наследование, полиморфизм, паттерны проектирования, модульность кода, масштабируемость программного обеспечения, поддержка кода, разработка программного обеспечения, принципы программирования.*

***Keywords**: object-oriented programming (OOP), encapsulation, inheritance, polymorphism, design patterns, code modularity, software scalability, code maintenance, software development, programming principles.*

1. Введение

Объектно-ориентированное программирование (ООП) представляет собой один из фундаментальных парадигм в программировании, который существенно влияет на подходы к проектированию и разработке программного обеспечения. ООП базируется на концепции инкапсуляции данных и поведения в объекты, что способствует созданию более модульного и масштабируемого кода. Эти объекты становятся строительными блоками программ, что позволяет разработчикам более эффективно управлять сложностью больших и многофункциональных систем.

Использование ООП в разработке программного обеспечения началось несколько десятилетий назад, но до сих пор остаётся ведущей методологией благодаря своей способности к абстракции, переиспользованию компонентов и лёгкости в поддержке. Принципы наследования, полиморфизма и инкапсуляции не только упрощают разработку, но и помогают программистам создавать более надёжные и гибкие решения, адаптируемые к изменяющимся требованиям или технологическим условиям.

В современной разработке программного обеспечения ООП способствует более гладкой интеграции и совместной работе различных систем и технологий, что критически важно в эру цифровизации и глобализации. Различные паттерны проектирования, разработанные на основе ООП, как например,

Singleton, Observer, Factory, Strategy и многие другие, предоставляют проверенные решения для часто встречающихся проблем в программировании.

2. Фундаментальные концепции ООП

Объектно-ориентированное программирование (ООП) основывается на трёх фундаментальных концепциях: инкапсуляция, наследование и полиморфизм. Эти принципы взаимодействуют, создавая систему, где программы состоят из модулей, которые можно легко модифицировать, расширять и поддерживать.

1. Инкапсуляция

Инкапсуляция — это концепция ООП, которая заключается в сокрытии внутренних данных и поведений объекта от внешнего мира. В практическом программировании это означает, что объекты управляют своим состоянием через методы, и только они могут изменять свои внутренние данные. Другие части программы взаимодействуют с объектом через его публичный интерфейс (публичные методы), не зная о внутреннем устройстве объекта.

Преимущества:

- Безопасность: Скрывая внутренние составляющие, предотвращаются несанкционированные или ошибочные изменения состояния объектов.
- Удобство использования и поддержка: Внешние компоненты системы не зависят от внутренней реализации объектов, что упрощает внесение изменений в код.

2. Наследование

Наследование позволяет новому классу перенимать (наследовать) свойства и методы уже существующего класса. Новый класс, называемый подклассом, может добавлять свои уникальные характеристики или изменять поведение унаследованных методов. Класс, от которого производится наследование, часто называют суперклассом или родительским классом.

Преимущества:

- Повторное использование кода: позволяет использовать уже проверенный код, не дублируя его.
- Иерархическая структурирование: облегчает понимание связей между различными компонентами программы и их организацию.

3. Полиморфизм

Полиморфизм — это способность объектов использовать методы производного класса, даже если они были объявлены в родительском классе. Это означает, что одна и та же операция может быть выполнена различным образом на объектах разных классов. Полиморфизм обычно достигается через механизмы такие, как переопределение методов (method overriding) и перегрузка методов (method overloading).

Преимущества:

- Гибкость: Программы становятся более гибкими и адаптируемыми, поскольку одинаковые операции могут работать по-разному с объектами разных классов.
- Масштабируемость: легко добавлять новые классы, которые используют уже существующие интерфейсы, без изменения существующего кода.

Эти три принципа вместе создают мощный инструментарий для разработчиков, позволяющий строить сложные системы из взаимосвязанных компонентов, которые легко тестировать, поддерживать и расширять, сохранив при этом строгую организацию и чистоту кода.

3. Влияние ООП на улучшение качества программного продукта и оптимизацию процесса разработки

Применение принципов объектно-ориентированного программирования (ООП) значительно повышает качество программного продукта и упрощает процесс его разработки. ООП обеспечивает разработчикам чёткую структуру для построения программ, что позволяет лучше организовать код и уменьшить его дублирование.

Четкая структура, которую предлагает ООП, делает код более понятным и логичным. Организация программы в виде взаимосвязанных объектов, каждый из которых имеет определённые свойства и поведение, способствует лучшему пониманию функций каждой части системы и их взаимодействия. Это упрощает процесс добавления новых функций или модификации существующих, так как изменения в одной части программы меньше влияют на другие части.

Инкапсуляция данных в объектах минимизирует зависимости между различными частями кода, что упрощает тестирование и отладку, поскольку можно сосредоточиться на одном объекте или методе, не затрагивая остальную часть программы. Это также помогает в обеспечении безопасности, так как чувствительные данные и операции можно скрыть внутри объекта, делая их недоступными извне.

Наследование позволяет разработчикам строить новые классы на основе уже существующих, расширяя их функциональность или модифицируя их поведение. Это сокращает необходимость повторного написания кода, поскольку можно использовать уже разработанные и протестированные решения. При этом сохраняется возможность вносить специфические изменения для каждого нового класса без влияния на родительские классы.

Полиморфизм, в свою очередь, дает возможность использовать один и тот же интерфейс для различных данных. Это позволяет программным компонентам быть взаимозаменяемыми, что упрощает расширение функциональности и адаптацию программы к новым требованиям без переписывания существующего кода.

В итоге, объектно-ориентированное программирование делает код более модульным, легко расширяемым и удобным для поддержки, что в конечном итоге повышает качество программного продукта и упрощает процесс его разработки и внедрения изменений.

4. Заключение

В заключение можно утверждать, что объектно-ориентированное программирование (ООП) представляет собой ключевой подход в современной разработке программного обеспечения, который значительно

улучшает как качество конечного продукта, так и процесс его создания. Принципы ООП, такие как инкапсуляция, наследование и полиморфизм, обеспечивают чёткую структурированность, модульность и расширяемость кода, что способствует легкой поддержке и адаптации программных систем к изменяющимся требованиям.

Через сокращение дублирования кода, обеспечение безопасности и упрощение тестирования, ООП позволяет разработчикам более эффективно управлять большими проектами, уменьшая при этом вероятность ошибок и повышая надёжность приложений. Наследование и полиморфизм предоставляют мощные инструменты для повторного использования кода и создания гибких систем, способных адаптироваться под различные задачи без переписывания существующих решений.

Список литературы

1. Казанский А. А. Объектно-ориентированный анализ и программирование на Visual Basic 2013 // Учебник / Сер. 68 Профессиональное образование. (1-е изд.) Москва, 2019.
2. Радынская В. Е., Поляничева А. В., Ахрамеева К. А. Разработка метода защиты ядра программных приложений с применением самомодифицирующегося кода //Региональная информатика и информационная безопасность.— 2019. — С. 136–141.
3. Красов А. В., Шариков П. И. Метод использования самомодифицирующегося кода для защиты приложения с кодовым зашумлением //Телекоммуникационные и вычислительные системы.— 2016. — С. 118–121.
4. Дюличева Ю. Ю. Перспективы развития современных обучающих сред для изучения объектно-ориентированного программирования // Проблемы современного педагогического образования. 2014. № 46–4. С. 98–104.
5. Искра Н.А., Близнюк Н.М., Жешко А. А. Подход к обучению объектно-ориентированному проектированию и программированию через реализацию игрового приложения // Объектные системы. 2016. № 13. С. 25–31.

УДК 004

Цифровые технологии в архитектуре

Исина Асем Зайсановна

кандидат технических наук, доцент Евразийского национального университета имени Л. Н. Гумилёва (Республика Казахстан, Астана)

Кендебай Архалых

магистрант Евразийского национального университета имени Л. Н. Гумилёва (Республика Казахстан, Астана)

***Аннотация**: На сегодняшний день цифровые технологий упрощают разные виды деятельности людей, например, в архитектуре. Данная работа определяет, насколько упрощена работа архитекторов, и как это будет влиять в будущем. Актуальность исследования определяется тем, что новые технологии стремительно внедряются в архитектурную систему. Цель исследования состоит в анализе цифровых технологий, влияющих на архитектурную деятельность.*

***Abstract**: Nowadays, digital technologies simplify different activities of people, for example, in architecture. This paper determines how much has the work of architects been simplified and what effect will it have in the future. The relevance of the study is determined by the fact that new technologies are rapidly being introduced into the architectural system. The purpose of the study is to analyze the digital technologies that affect architectural work*

***Ключевые слова**: цифровая грамотность — безопасность в информационном обществе, фундамент, самое важное знание 21 века, цифровые технологии, архитектура, инженерия, гражданское строительство.*

***Keywords**: digital literacy — security in the information society, the foundation, the most important knowledge of the 21st century, digital technologies, architecture, engineering, civil engineering.*

Важнейшие знания 21 века, одна из наших основных тем Цифровая грамотность — это желание и умение уверенно и эффективно использовать цифровые технологии во всех сферах жизни человека. Использование этой технологии открывает путь к повышению качества жизни людей.

Действительно, не годами, а месяцами, даже неделями и днями человеческое сообщество входит в волшебный мир цифровизации. Технолo-

гии цифровизации — это новые инструменты чудесного мира, которого человечество еще никогда не испытывало. То есть эти технологии сейчас разрабатываются. Они уже оставляют позади те самые информационные технологии, которыми мы восхищаемся.

Сегодня концептуально система образования ведется по трем основным направлениям: цифровизация образовательного процесса, цифровой образовательный контент, цифровизация управления образованием. Цифровизация школьного образования в Казахстане является одним из основных направлений в процессе его реформирования. Видение школ будущего часто связано с постепенным переходом всех предметов на облачную систему образования. Речь идет об онлайн-уроках и виртуальных лабораториях, открытом образовательном контенте, гибком и индивидуальном подходе к каждому участнику. Студенты могут вместе работать над домашними заданиями онлайн. Школьные библиотеки превратились в информационные и компьютерные центры. Процесс обучения будет привязан к идентификатору каждого учащегося, что позволит проводить оценивание и выставление оценок.

Цифровизация школ означает создание удобных и эффективных инструментов для всех участников этого процесса: учащихся, их родителей, учителей, администрации системы образования. Кроме того, важно, что цифровизация процесса обучения представляет собой своеобразный синтез реального и цифрового мира в оптимальном балансе взаимодействия человека и виртуальной среды.

При этом цель создания цифровой платформы, улучшающей качество жизни людей, активно участвует в реализации предложенной Президентом программы «Цифровой Казахстан-2020».

Что касается гибкости и компетентности XXI века, то они должны формироваться во всей образовательной деятельности, начиная с начальной школы. Остановимся на некоторых аспектах цифровизации образования, в частности, на внедрении системы общего образования. В последнее время наблюдается процесс создания и использования онлайн-ресурсов открытого общего образования, общего развития, от индивидуальных заданий до полноценных курсов и модулей. Активно ведется формирование закрепленных компетенций. Единая платформа онлайн-курсов позволяет

каждому быстро адаптироваться к информационным потокам, оценивать информацию, принимать решения в особых ситуациях, словом, овладевать навыками XXI века.

Ре-цифровизация социальной парадигмы жизни людей открывает возможность расширить поле мышления людей и получить новые знания. Одним из основных направлений современного образования является нетворкинг, использование социальных сетей в качестве образовательных ресурсов и проведение дистанционных мастер-классов, тренингов. Типичными чертами цифрового образования с использованием сетевых технологий являются гибкость, мобильность, продуктивность, диалогичность и интерактивность, ориентация на прием медиапотоков.

Основная цель цифровизации — повышение конкурентоспособности, улучшение качества жизни населения, ускорение и упрощение образовательного процесса, снижение нагрузки на детей, учителей и родителей. Самое главное — повысить качество образования. Наши дети должны быть конкурентоспособными на международном уровне в различных областях, включая искусственный интеллект и большие данные. Как подчеркнул глава государства, цифровизация страны — это не цель, а средство достижения абсолютного превосходства Казахстана. Весь процесс требует последовательности, порядка и комплексного подхода.

Основной задачей цифровизации в сфере образования является повышение качества образования, то есть подготовка молодежи страны, конкурентоспособной в различных сферах, включая «искусственный интеллект» и «большие данные» на международном уровне.

Архитектурная, инженерная и строительная отрасль переживает значительный переход от традиционных трудоемких методов к автоматизации за счет использования цифровых технологий и сыграла значительную роль в этой революции. В целом, цифровые технологии обычно относятся к информационным и коммуникационным технологиям, которые облегчают разработку, хранение и обработку информации и способствуют различным формам связи между людьми и электронными системами, а также между электронными системами в цифровых двоичных вычислительных системах. В архитектурной, инженерной и строительной отраслях, где большое внимание уделяется этим технологиям, становится все более

важным оценивать их влияние на поведение пользователей, поскольку цифровые технологии меняют взаимосвязь между строительным процессом и поведением человека. В результате роста концепций цифровизации и автоматизации в промышленности больше внимания стало уделяться цифровым технологиям. Их можно использовать для логистических операций, потоков знаний в реальном времени, интеграции цепочек поставок и расширения человеческого взаимодействия за счет использования цифровых технологий, особенно трудоемких видов деятельности.

Информационное моделирование, дополненная реальность, виртуальная реальность, фотограмметрия, устройства радиочастотной идентификации, геоинформационные системы, системы глобального позиционирования, носимые устройства, устройства безопасности, код быстрого ответа, искусственный интеллект, робототехника, блокчейн, наземные мобильные устройства и устройства лазерного сканирования считаются многообещающими цифровыми технологиями в сфере архитектуры, инжиниринга и строительства. В целом общеизвестно, что польза от использования цифровых технологий в архитектурной, инженерной и строительной отрасли огромна. Например, технологии AR-VR показали многообещающие результаты в ряде областей, связанных с отраслью архитектуры, инжиниринга и строительства, предлагая решения проблем сотрудничества и коммуникации. Такие методы, как BIM для проектной документации и управление архитектурой позволяют командам сотрудничать, сотрудничать и обмениваться данными проекта. Обобщены различные примеры интеграции VR.

Приложения, связанные с архитектурой, инжинирингом и строительством, показывают многообещающие результаты, такие как городское планирование, акустический анализ, анализ проектирования и поддержка принятия решений, безопасность зданий, улучшение пространственного восприятия и результаты обучения в области инженерии. регионы. Кроме того, RFID может помочь повысить прозрачность и производительность цепочки поставок. Аналогичным образом, для сбора данных, мониторинга, визуального осмотра и оценки.

Для этих целей могут быть использованы технологии GPS-ГИС. Хотя использование робототехники потенциально может повысить эффектив-

ность и безопасность, оно не должно сокращать общие возможности трудоустройства в строительном секторе в долгосрочной перспективе. С помощью камеры смартфона приложение считывает QR-код и отображает бесконфликтный шаблон на дисплее. Пользователи могут комбинировать, интерпретировать или использовать плоскости сечения, чтобы легко визуализировать дизайн чертежей. Кроме того, ИИ предлагает большой потенциал для значительного повышения эффективности за счет быстрого и точного анализа больших объемов данных. Более того, системы и технологии искусственного интеллекта могут решать сложные нелинейные функциональные проблемы, а после их оснащения прогнозы и обобщения можно делать с высокой скоростью. В связи с растущим спросом на цифровые технологии в архитектурной, инженерной и строительной отрасли необходимы дополнительные исследования в этой области. Однако предыдущие обзорные исследования в этой области внесли значительный вклад, но имеют различные ограничения. Сначала они использовали ручные методы. Во-вторых, не существует всеобъемлющих обзоров современных исследований, охватывающих все аспекты цифровых технологий под одной крышей. Например, Ибем и Лария провели обзор литературы, в котором основное внимание уделялось только закупкам строительных проектов. Го и др. просмотрел литературу, но ограничился границей безопасности строительства. Пауэлс и др. просмотрел литературу, но ограничился преимуществами технологий семантической паутины. Дарко и др. Провел первое комплексное наукометрическое исследование, оценивающее современное состояние исследований искусственного интеллекта в сфере архитектуры, инжиниринга и строительства, но ограниченное сферой искусственного интеллекта. В контексте этих доказательств эти аналитические исследования не дают полной картины современного состояния исследований цифровых технологий в архитектурной, инженерной и строительной отрасли. Фактически, до сих пор не хватает исследований, которые бы давали подробную картину и понимание литературы по цифровым технологиям в конкретной области AEC. Таким образом, целями данного исследования являются: RQ1. Какова ежегодная тенденция публикаций о цифровых технологиях в отрасли архитектуры, инжиниринга и строительства? РК2. Какие журналы публикуют исследования

о вкладе цифровых технологий в индустрию AEC? РК3. В каком контексте (стране) проводилось исследование? РК4. Какие цифровые технологии появятся больше всего среди всех других цифровых технологий? RQ5. Каковы новые достижения, проблемы, негативное отношение и будущие направления развития цифровых технологий в архитектурной, инженерной и строительной отрасли? Чтобы восполнить этот пробел в исследованиях, для поиска ответов на эти вопросы в интересах научных кругов и промышленности использовались как количественные (т. е. библиометрический анализ), так и качественные (т. е. систематический обзор) подходы. Ожидается, что ответы на эти вопросы будут способствовать познанию литературы и цифровых технологий. В результате цифровых технологий, более широкого развития рынка и возросшего интереса со стороны практиков, заинтересованных сторон в сфере строительства и исследователей к глобальной среде, чтобы извлечь выгоду из новых разработок и будущих направлений.

По сравнению с XX веком (1975–2000 гг.) и XXI веком (2001–2020 гг.) в XXI веке (2001–2020 гг.) наблюдается более высокая тенденция публикаций по цифровым технологиям в архитектуре, машиностроении и строительстве. Робин факт: цифровые технологии стали самым мощным технологическим инструментом 21 века. Билал и др. и свидетельства растущей тенденции использования цифровых технологий в архитектуре, машиностроении и строительстве в 21 веке. В XX веке, в конце 1990-х годов, исследования в области цифровых технологий набрали обороты, поскольку вычислительные возможности и желание исследователей использовать цифровые технологии для решения масштабных сложных задач стали возрастать. Благодаря этому количество публикаций могло достичь небольшого пика в 1996 году. Однако в 2014–2020 годах тенденция публикаций о цифровых технологиях в архитектуре, машиностроении и строительстве усилилась. Такое увеличение количества публикаций может быть связано с необходимостью и уровнем внедрения в архитектурной, инженерной и строительной отраслях. Это показывает, что внедрение цифровых технологий в 21 веке увеличивается и увеличивает его преимущества. Например, сложно представить себе стройку по сравнению с XX веком. Однако в 21 веке фотограмметрия сделала визуализацию проще и быстрее. Ана-

логично, процесс мониторинга строительной площадки надежно реализуется с помощью цифровых технологий.

Список литературы

1. Wang, M.; Wang, C.C.; Sepasgozar, S.; Zlatanova, S. A Systematic Review of Digital Technology Adoption in Off-Site Construction: Current Status and Future Direction towards Industry 4.0. Buildings 2020, 10, 204. [CrossRef].
2. Patel, S.; Patt, Y. Introduction to Computing Systems: From Bits & Gates to C & Beyond; McGraw-Hill Professional: New York, NY, USA, 2019.
3. Xue, X.; Shen, Q.; Ren, Z. Critical Review of Collaborative Working in Construction Projects: Business Environment and Human Behaviors. J. Manag. Eng. 2010, 26, 196–208. [CrossRef].
4. Alaloul, W.S.; Liew, M.; Zawawi, N.A.W.A.; Kennedy, I. B. Industrial Revolution 4.0 in the construction industry: Challenges and opportunities for stakeholders. Ain Shams Eng. J. 2020, 11, 225–230. [CrossRef].
5. Maskuriy, R.; Selamat, A.; Maresova, P.; Krejcar, O.; David, O. O. Industry 4.0 for the construction industry: Review of management perspective. Economies 2019, 7, 68. [CrossRef].
6. Woodhead, R.; Stephenson, P.; Morrey, D. Digital construction: From point solutions to IoT ecosystem. Autom. Constr. 2018, 93, 35–46. [CrossRef].
7. Ozturk, G. B. Interoperability in building information modeling for AECO/FM industry. Autom. Constr. 2020, 113, 103122. [CrossRef].
8. Kim, K.; Kim, H.; Kim, H. Image-based construction hazard avoidance system using augmented reality in wearable device. Autom. Constr. 2017, 83, 390–403. [CrossRef].

УДК 004

Digitization and technological development in the sphere of international tourism

Куткович Алина Игоревна

студент Западного филиала Российской академии народного хозяйства и государственной службы при Президенте Российской Федерации

***Abstract**: The article deals with the impact of digitalisation and technology development on the sphere of international tourism. The author analyses how the latest technological solutions, including mobile applications, artificial intelligence, virtual and augmented reality, as well as blockchain, are transforming the ways of planning, booking and conducting travel. Special attention is given to personalising travel services, ensuring secure and transparent transactions, and promoting sustainable and green tourism. The article emphasises the importance of global collaboration and data sharing for the sustainable development of the tourism industry, highlighting the role of technology in creating more inclusive and accessible international tourism.*

***Аннотация**: В статье рассматривается влияние цифровизации и развития технологий на сферу международного туризма. Автор анализирует, как новейшие технологические решения, включая мобильные приложения, искусственный интеллект, виртуальную и дополненную реальность, а также блокчейн, трансформируют способы планирования, бронирования и проведения путешествий. Особое внимание уделяется персонализации туристических услуг, обеспечению безопасности и прозрачности сделок, а также продвижению устойчивого и экологичного туризма. Статья подчеркивает важность глобального сотрудничества и обмена данными для устойчивого развития туристической отрасли, выделяя роль технологий в создании более инклюзивного и доступного международного туризма.*

***Keywords**: digitalisation in tourism, technology development, international tourism, artificial intelligence, mobile applications, virtual reality, augmented reality, blockchain, sustainable tourism.*

***Ключевые слова**: цифровизация в туризме, развитие технологий, международный туризм, искусственный интеллект, мобильные приложения, виртуальная реальность, дополненная реальность, блокчейн, устойчивый туризм.*

Digitalisation and technological developments are playing a key role in transforming the international tourism industry. These changes are affecting all aspects of the industry, from how people search and book travel, to how they interact with

customers and manage tourism facilities. Let's take a look at exactly how digitalisation and technology are affecting international tourism and what innovations are expected in the future.

One of the key areas where technology is having a significant impact is the personalisation of travel services. Modern data analytics technologies allow travel companies to collect, analyse and use large amounts of data on customer preferences and behaviour. This makes it possible to offer personalised solutions and offers that best match the needs and interests of each tourist.

In the age of digitalisation, tourism is being transformed by vast and diverse data sources. Notable among them are communication systems, including data from mobile operators and social networks, as well as web resources, both personal and corporate websites. Transaction data from booking systems and retailers, as well as information from physical sensors that capture travellers' interests and movements, play an important role. Crowdsourcing — publicly available content from platforms such as YouTube and TripAdvisor — is also making an important contribution.

This data is becoming a key resource in the creation of tourism products and services. They allow future travellers to learn about the features of a travel destination and its offerings in advance. While travelling tourists have the opportunity to share their impressions, thus enriching the information space with their experience. In turn, this confirms the concept of "experience economy", which is actively discussed in the international scientific environment.

The digital economy, which is formed at the intersection of information technologies and tourism business, puts information and access to it at the centre of attention. Modern information and communication technologies and services are becoming not just tools, but the main drivers of development, providing the necessary environment for the prosperity of the digital economy in tourism [4, p.125].

Mobile applications have long been an integral part of travelling. They provide convenient access to booking flights, hotels, car rentals and other travel services. In addition, mobile applications offer useful features such as maps, travel guides, local sightseeing tips and even real-time translators, which makes travelling more comfortable and safer [2, p.137].

Virtual (VR) and augmented reality (AR) technologies open new opportunities for international tourism. With their help, potential tourists can "visit" hotels,

attractions and even entire cities in advance without leaving their homes. This not only helps in the decision to travel, but also enriches the on-site experience itself by providing interactive maps, directions and information about attractions.

Artificial intelligence (AI) and machine learning are helping to automate and optimise many processes in the travel industry. From chatbots for customer service to algorithms that determine optimal itineraries and offers, these technologies significantly improve the quality and efficiency of services [1, p.103].

Blockchain technology finds its application in tourism to ensure the security and transparency of bookings and payments. It allows to simplify processes and reduce commissions, as well as provides protection against fraud. Blockchain can be used to create a robust review and rating system that is resistant to tampering, which makes the choice of travel services more transparent and trustworthy for consumers.

Between 2018 and 2024, Russia's strategic initiatives place a special emphasis on the National project "Tourism and Hospitality Industry". This project is being created taking into account the latest trends and challenges of our time, aimed at developing the tourism industry in the digital era. Among the key objectives of the project is to enrich the digital space of Russian tourism. This includes improving the quality and attractiveness of information materials, developing online services for tourists and digitalising consumer interaction. In addition, special attention is paid to the digital transformation of management processes in tourism, which involves the introduction of modern technologies in all aspects of activity, from planning to the implementation of tourist products [5, p.207].

The use of digital technologies requires innovative steps from the state, in particular, uniting public and private technology companies for the development of tourism. As world experience shows, the partnership of private and public sectors, using an unprecedented amount of digital data, aims to analyse the current tourism market, as well as to develop measures to respond to the demands of current consumers.

Digitalisation also contributes to the development of sustainable and green tourism. Today, this is directly linked to the use of various online platforms that provide tourists with many options to choose nature-oriented travel destinations and book hotels with minimal environmental impact. With the digitalisation of tourism, people are becoming more aware and can participate in conservation

programmes. Also innovations in tourism significantly reduce the negative impact of this industry on the environment, control the use of resources and waste treatment processes.

The development of international co-operation in tourism is important and advanced technological solutions contribute to the creation of more effective strategies for the development of this industry. Improvements lead to better service levels for tourists and the development of the economy and infrastructure of tourist areas [3, p.188].

International tourism faces various challenges such as climate change and increasing population mobility. Nevertheless, thanks to technology, safe and comfortable travelling conditions are being created. Advanced technological solutions in the field of international tourism, not only contribute to its development, but also enrich the global cultural and economic exchange, opening new opportunities to improve services and create unforgettable experiences for tourists from different countries.

References

1. Pavlyuk, E. S. Management in advertising: language. culture. digital technologies: a monograph / E. S. Pavlyuk, L. V. Pavlyuk, M. I. Grigoryan [etc.]; ed. by E. S. Pavlyuk. — Moscow: Rusains, 2023.— 103 p.
2. Moskvitin, G. I. Management and personnel management: innovations, digital technologies: a collection of articles / G. I. Moskvitin, V. A. Kozyrev, T. N. Yarova. — Moscow: Rusains, 2020.— 137 p.
3. Vasilieva, M. V. Marketing research of the service sphere: traditional and digital technologies: a textbook / M. V. Vasilieva, V. A. Budasova, E. A. Krug [et al]; ed. by M. V. Vasilieva. — Moscow: Rusains, 2024.— 188 p.
4. Moskvitina, G. I. Management: digital technologies, methods, control: a collection of articles / edited by G. I. Moskvitina, T. V. Yarov, Collective of authors. — Moscow: Rusains, 2020.— 125 p.
5. Maslennikov, V. V. Digital management: a textbook / V. V. Maslennikov, Y. V. Lyandau, I. A. Kalinina [et al]. — Moscow: Knorus, 2022.— 207 p.

УДК 004

ERP- и CRM-система ODOO

Ерофеева Анастасия Дмитриевна

студент факультета Прикладной информатики Института цифровых технологий управления и информационной безопасности

***Аннотация**: В статье рассматривается бельгийская ERP- и CRM — система «ODOO», которая завоевывает мир своей многофункциональностью, гибкостью и возможностью доработки. Даётся общее представление о системе, её структуре, ограничениях и возможных сферах применения.*

***Abstract**: The article deals with the Belgian ERP and CRM system "ODOO", which conquers the world by its multifunctionality, flexibility and possibility of modification. The general idea about the system, its structure, limitations, and possible spheres of application is given.*

***Ключевые слова**: erp, crm, odoo, система, открытый код, модуль.*

***Keywords**: erp, crm, odoo, system, open-source, module*

История системы

История системы началась в 2005 году под названием «TinyERP». Основателями системы являются Фабиан Пинкарс и Антони Лесусс [1]. Они написали простую систему управления предприятием на языке Python, которая была предназначена для внедрения на небольших бельгийских фирмах, работающих в основном в сфере услуг и дистрибуции [2]. В 2008 году система была переименована в «OpenERP». Фабиан Пинкарс усердно работал над системой, и она стала набирать популярность. Совместно с Антони Лесусс они разработали десятки модулей. В 2010 году в компании уже работали более 100 сотрудников, которые занимались продажей услуг на «Open ERP». Она произвела переворот на рынке ERP и вышла далеко за пределы традиционных ERP-игроков. Интеграция бизнес — процессов больше не ограничивается продажами, бухгалтерским учетом, инвентаризацией и закупками. В июне 2014 года компания выпустила версию 8 с потрясающей CMS и электронной коммерцией, торговой точкой, интегрированным механизмом бизнес-аналитики и многим дру-

гим. В мае 2014 года компания и продукт были переименованы в «Odoo». На текущий момент в фирме Odoo S.A. работает более 250 сотрудников, она имеет 6 офисов (в Бельгии, Люксембурге, Индии, Гонконге, два офиса в США — в Сан — Франциско и Нью — Йорке). В настоящее время Odoo ERP включает в себя 46 модулей и более 17 000 приложений. Вам понадобятся не все из них, но некоторые из них актуальны для компаний в разных отраслях. Также насчитывается около 12 миллионов пользователей «Odoo» по всему миру, а более 800 компаний-партнёров в 120 странах занимаются распространением и внедрением системы. «Odoo» успешно используется в таких известных в России компаниях как Auchan, Danone, Hyundai, Айкрафт, Onity и Stark Control. В октябре 2023 года вышла семнадцатая версия системы.

Структура системы

Структура системы максимально упрощена и включает три компонента (рис. 1):

1. СУБД PosgreSQL (служит для хранения всех данных системы и части настроек).
2. Сервер приложений (реализует бизнес-логику конкретного предприятия).
3. Web-сервер (предоставляет доступ через web-браузер ко второму компоненту).

В зависимости от объёма решаемых задач все части системы могут быть установлены на одном сервере или размещены на разных.

Версии системы

Odoo делится на две версии: community и enterprise.

Community — бесплатная версия системы с открытым исходным кодом. Данная версия содержит базовый набор модулей, например, CRM, Продажи, Закупки, Склад и Производство. Все модули настроены на стандартные бизнес-процессы компаний. Если же ваши бизнес-процессы имеют некоторые нюансы и особенности, то систему можно с легкостью

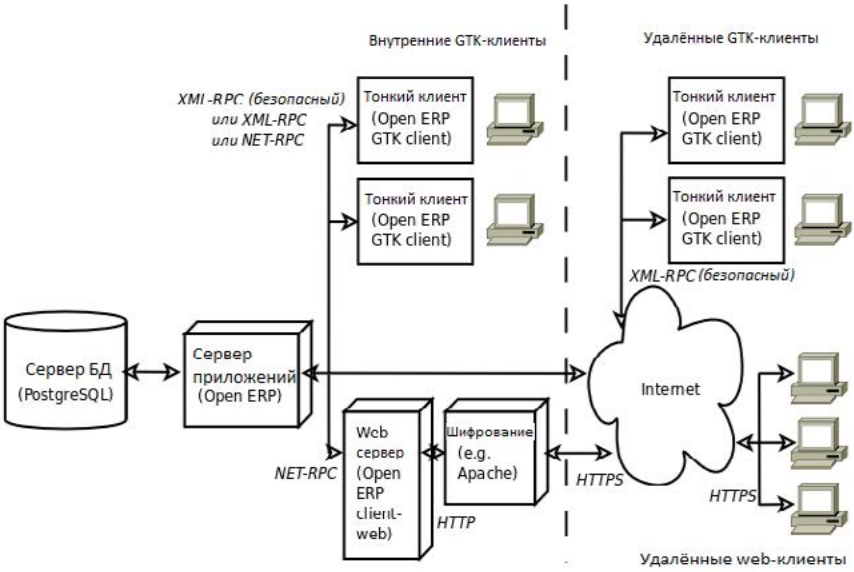

Рисунок 1. **Структура системы**

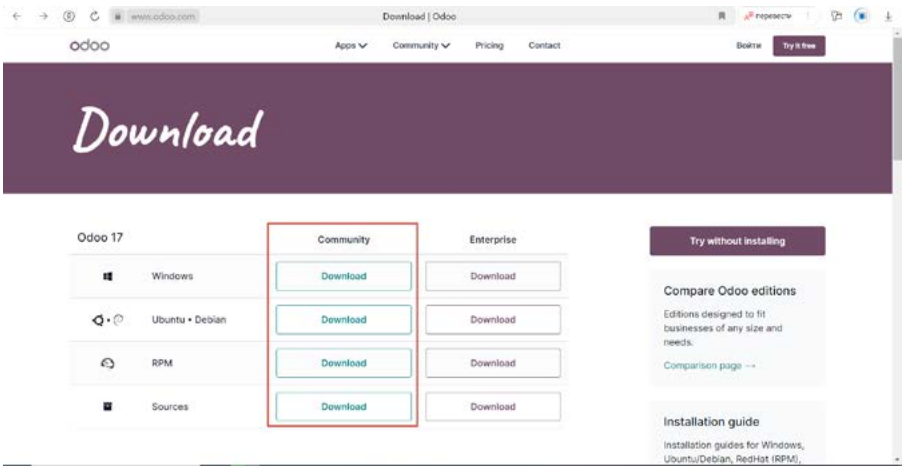

Рисунок 2. **Скачать бесплатную версию**

доработать. Для знакомства с odoo можно скачать бесплатную версию на официальном сайте odoo.com (рис. 2)

Enterprise — платная версия системы, которая имеет больше готовых модулей и функций, мобильное приложение полностью адаптивное под любое устройство, ежегодное обновление версии, а также бесперебойную техническую поддержку. Есть несколько вариантов размещения системы:

1. Online [3] — ПО, размещенное в облаке. Можно использовать только стандартные приложения и функционал по умолчанию. Не будет предусмотрено добавление сторонних приложений. Ежемесячная абонентская плата зависит от количества установленных приложений и количества пользователей.
2. В облаке (cloud или odoo.sh) [4]. При работе с odoo.sh мы получаем три инстанса: dev (среда разработки), stage (промежуточная среда), prod (производственная среда). Через интуитивно понятную панель мониторинга odoo.sh доступны командная строка, журналы, резервные копии баз данных.
3. Локально (on-premise) [4]. Систему необходимо также установить и настроить. Обслуживать сервер придется самостоятельно. Стоимость лицензии на одного пользователя, как и для odoo.sh.

Актуальные цены на лицензии для той или иной страны необходимо смотреть на официальном сайте odoo.

Лицензии

Приложения для системы можно приобретать на сайте odoo.apps. Там есть как платные, так и бесплатные приложения, созданные сторонними специалистами. Эти приложения могут распространяться под разными лицензиями: GPL-3, AGPL-3, LGPL-3, OPL-1

GPL-3

GNU General Public License — это свободная лицензия (копилефтная, т.е. если вы использовали в своей программе библиотеку под копилефтной лицензией, то вам придется распространять вашу программу под ней

же) для программного обеспечения, разработанная Фондом Свободного Программного Обеспечения (Free Software Foundation).

AGPL-3

Affero GPL (AGPL) [6] —усиленная версия GPL, которая отличается от прародителя только одним пунктом (раздел 13). Идея лицензии AGPL заключается в устранении "лазейки поставщика услуг приложений (ASP)", которая, существовала в GPL. Лазейка в ASP означала, что поставщики программного обеспечения как услуги (SaaS) и другого программного обеспечения, которое работало преимущественно по сети, были освобождены (или потенциально могли оспорить освобождение) от условий лицензии GPL. Это потому, что они технически не "распространяли" его в традиционном смысле.

LGPL-3

В отличие от основной лицензии GPL, использование работы под LGPL [5,6] в качестве части более крупной работы не обязывает лицензировать всю большую работу под LGPL или раскрывать ее исходный код. Однако код самой библиотеки должен быть доступен по первому запросу в соответствии с условиями лицензии LGPL.

OPL-1 (Odoo Proprietary License v1.0) [6]

Проприетарная лицензия Odoo версии 1.0.

Для использования данного программного обеспечения и связанных файлов (ПО) необходимо приобрести действующую лицензию у авторов ПО, обычно через Odoo Apps, или получить письменное соглашение от них. Можно создавать модули Odoo, которые используют ПО в качестве библиотеки, не копируя при этом исходный код или материалы. Разрешается распространять такие модули под совместимой с лицензией Odoo (например, LGPL, MIT или другой проприетарной лицензией) лицензией на выбор, но запрещается публиковать, распространять, сублицензировать или продавать копии ПО или его модификаций.

Список литературы

1. The Tryton Project [Электронный ресурс]. Сайт конференции IWEEE — Режим доступа: http://www.iweee.org/2015-las_palmas/presentations/files/Nicoe-1- IWEEE2015.pdf (дата обращения: 17.03.2024).
2. Odoo [Электронный ресурс]. Сайт Wikipedia — Режим доступа: https://ru.wikipedia.org/wiki/Odoo (дата обращения: 17.03.2024).
3. Odoo online [Электронный ресурс]. Сайт — Режим доступа: https://www.cybrosys.com/blog/odoo-online (дата обращения: 17.03.2024).
4. Odoo on-premise [Электронный ресурс]. Сайт — Режим доступа: https://bassaminfotech.com/odoo-sh-vs-on-premise-hosting/ (дата обращения: 17.03.2024).
5. LGPL лицензия [Электронный ресурс] Сайт — Режим доступа: https://www.odoo.com/ru_RU/blog/odoo-news-5/adapting-our-open-source-license-245 (дата обращения: 17.03.2024).
6. Лицензии [Электронный ресурс] Сайт — Режим доступа: https://www.cybrosys.com/blog/an-overview-of-different-odoo-16-licenses (дата обращения: 17.03.2024).

УДК 004

Обзор моделей оптимального размещения рабочих мест

Сивушков Иван Дмитриевич

студент магистратуры Института информационных технологий и анализа данных Иркутского национального исследовательского технического университета

***Аннотация**: В статье рассматривается проблема автоматизации планировки офисного пространства. Проведён обзор академических исследований в данной области. Поставлена задача поиска оптимального размещения рабочих мест в комнате, с целью максимизации количества размещённых сотрудников. Рассмотрены варианты математической модели представления рабочего места. Перечислены популярные методы решения задачи упаковки.*

***Abstract**: The paper deals with the problem of automating office space layout. The review of academic researches in this field is carried out. The problem of finding the optimal placement of workstations in a room in order to maximize the number of placed employees is posed. Variants of mathematical model of workplace representation are considered. Popular methods for solving the packing problem are listed.*

***Ключевые слова**: рабочее место, математическое моделирование, комбинаторная оптимизация, задача упаковки, упаковка прямоугольников, упаковка окружностей; методы решения задачи упаковки.*

***Keywords**: workplace; mathematical modeling, combinatorial optimization, packing problem, rectangle packing, circle packing, packing problem-solving methods.*

Актуальность

При расширении рабочего штата, возникают трудности в обеспечении достаточного офисного пространства. Для решения этой проблемы этого компании открывают новые офисы или повышают эффективность использования имеющихся помещений. Аналогично, при спаде деятельности компании необходимо оптимизировать расходы, сокращая излишнее офисное пространство. Для обоих случаев необходимо построить новый план офисного помещения. Однако, чем больше площадь офисного пространства, тем более трудоёмким может быть процесс разработки соответствующего плана.

Методы решения

Существует множество способов упростить и ускорить производство офисного плана, одни из основных направлений:

- Привлечение специалистов;
- Использование программного обеспечения.

На сегодняшний день существуют различные программные решения, направленные на упрощение процесса планировки офисных пространств. Эти инструменты представляют собой онлайн или десктопные конструкторы, предоставляющие пользователю возможность поэтажного проектирования помещений. Они базируются на использовании шаблонов и гото-

вых фигур, размещаемых на чертеже. Примеры: Remplanner[1], Planner 5D[2], HomeStyler[3]

Также следует обратить внимание на академические исследования, проведённые в данной области:

В статье Augmented space planning: Using procedural generation to automate desk layouts [1] авторы рассматривают задачу компоновки рабочих мест в коммерческих офисах и алгоритмы её решения.

Далее, в статье Using genetic algorithm to automate the generation of open-plan office layout [2] рассматривается вариант автоматизации открытой планировки офиса с использованием генетического алгоритма.

В следующей статье Office Space Allocation Optimization [3] авторы рассматривают задачу распределения офисных помещений по отделам и сотрудникам, с целью минимизации использования офисного пространства, а также применяют и сравнивают алгоритмы жадного поиска и поиска с запретами для решения этой задачи.

Также полезной является статья Tell2Design: A Dataset for Language-Guided Floor Plan Generation [4] в которой рассматривается задача генерации планов этажей офиса на основе описания на естественном языке. В статье представлен набор данных Tell2Design, состоящий из планов этажей, связанных с текстовыми инструкциями, а также seq2seq модель для решения поставленной задачи и её сравнение с другими моделями текстовой генерации изображений.

Чем можно улучшить

Базовый подход к планированию офисного пространства, заключающийся в размещения примитивов на чертеже, может быть усовершенствован с применением автоматизации процессов пространственной конфигурации внутри отдельных комнат. Для достижения данной цели

[1] Remplanner — онлайн инструмент для составления технических чертежей [Электронный ресурс]. URL: https://remplanner.ru/ (дата обращения: 24.04.2024)

[2] Planner 5D — онлайн инструмент для планирования помещений [Электронный ресурс]. URL: https://planner5d.com/ru (дата обращения: 24.04.2024)

[3] Homestyler — Floor Planner Online [Электронный ресурс]. URL: https://www.homestyler.com/ (дата обращения: 24.04.2024)

предлагается воспользоваться математическим моделированием и методами комбинаторной оптимизации, направленными на поиск оптимального размещения объектов в ограниченном двумерном пространстве.

Этот подход позволит эффективно решать задачи по оптимизации организации пространства, учитывая ограничения и целевые функции, что в итоге способствует более рациональному использованию данных ресурсов.

Постановка задачи

Дана комната в офисе. В качестве объектов, размещаемых в комнате, выбраны рабочие места сотрудников офиса. Требуется разработать оптимальную конфигурацию расположения рабочих мест в комнате.

Система ограничений состоит из:

1. Характеристик комнаты: Прямоугольная форма;
2. Характеристик рабочего места: Полукруглый рабочий стол; Наличие компьютера с подключённым монитором.
3. Установленных требований к нормативу площади для одного рабочего места с компьютером.
4. Характеристик расположения рабочего места в комнате: Угол поворота относительно южной (нижней) стены; Наличие пути от каждого рабочего места до выхода из помещения.

Целевая функция задачи оптимизации заключается в нахождении максимального количества сотрудников, размещённых за рабочими местами в данном офисном помещении, с учётом системы ограничений.

Математическая модель задачи

Следующим шагом для поставленной задачи выбирается математическая модель.

В качестве первого варианта, был рассмотрен подход, использованный в работе “Математика социальной дистанции — это урок геометрии” [5]. В связи с чем была выбрана модель задачи упаковки кругов в прямоугольник.

Цель задачи — вместить максимально возможное количество одинаковых непересекающихся кругов в прямоугольную область, не нарушив её границ. В нашем случае круги обозначают площадь рабочего места, а прямоугольная область обозначает комнату.

Порядок действий:

1. Определение минимального радиуса окружности на основе норматива площади рабочего места.
2. Размещение рабочего стола в окружности рабочего места.
3. Размещение окружностей рабочих мест в прямоугольной комнате.

На рисунке 1 изображены примеры расположения рабочих мест, когда монитор рабочего стола размещён тыльной стороной к стене, где зелёные круги обозначают область рабочего места

Когда рабочее место рассматривается как окружность, рабочие столы не соприкасаются и между ними есть боковой проход, что позволяет пройти от рабочего места к выходу из комнаты.

Разместив рабочие столы в окружности, получаем, что их не получится поставить вплотную к стене или в углу. И каждое рабочее место будет отрезать дополнительное неиспользуемое пространство.

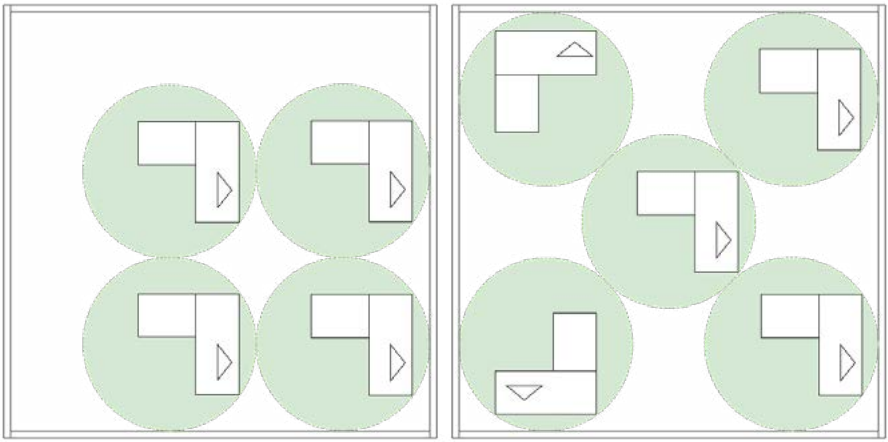

Рисунок 1. **Примеры заполнения рабочими местами прямоугольной комнаты**

Несмотря на это, данный вариант позволяет организовать просторное расположение сотрудников, что может подходить для коворкингов и компаний с проектной организационной структурой, когда важно иметь возможность пересматривать расположение рабочих мест по мере работы над проектами и при изменении количества задействованных сотрудников.

В связи с выявленным ограничением модели, был рассмотрен следующий вариант. За основу второй модели взята задача упаковки прямоугольных объектов в прямоугольную область [6].

Цель задачи — вместить максимально возможное количество непересекающихся прямоугольных объектов в прямоугольную область, не нарушив её границ. В нашем случае прямоугольные объекты обозначают площадь рабочего места, а прямоугольная область обозначает комнату.

Порядок действий:

1. Определение ширины и высоты прямоугольного объекта на основе норматива площади рабочего места. В случае с прямоугольной областью рабочего места, ширина и высота могут варьироваться.
2. Размещение рабочего стола в области прямоугольного объекта;
3. Размещение прямоугольного рабочего места в прямоугольной комнате;

На рисунке 2 изображены примеры расположения рабочих мест, когда монитор рабочего стола размещён тыльной стороной к стене, где зелёные прямоугольники обозначают область рабочего места.

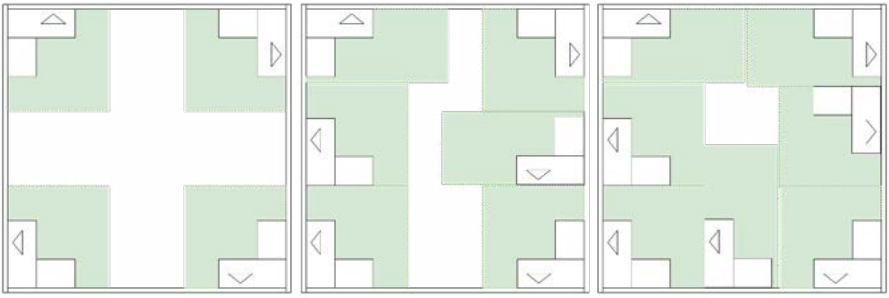

Рисунок 2. **Примеры заполнения комнаты прямоугольными рабочими местами**

В данном варианте, когда рабочее место рассматривается как прямоугольник, рабочие столы, как и в предыдущем варианте, не соприкасаются и позволяют, пройти от рабочего места к выходу их комнаты.

Разместив рабочие места как прямоугольные области, получаем, что все условия из системы ограничений соблюдаются, но результаты размещения сильно зависят от выбранных размеров прямоугольных рабочих мест. В связи с этим при поиске оптимального размещение потребуется перебор различных размеров рабочих мест, что вычислительно усложнит решение задачи. Ограничить количество вариантов перебора можно, предварительно подобрав несколько вариантов размеров рабочего места.

Этот вариант позволяет организовать плотное размещение сотрудников, что может подходить для организаций, в которых сотрудники разделены по отделам и расположение рабочих мест меняется не часто.

Методы решения

В литературе, задачи упаковки также иногда называются задачами раскроя. Термины “упаковка” и “раскрой” — используются для описания задач распределения объектов в ограниченное пространство, чтобы минимизировать потери и максимизировать использование пространства.

Для решения задачи упаковки прямоугольных объектов можно воспользоваться простыми эвристическими методами:

- Методы следующего подходящего (Next fit), первого подходящего (First fit), наилучшего подходящего (Best fit)[7],
- Метод “Снизу-Слева” (Bottom-Left)[8].

А также модификациями простыми эвристических методов[9]:

- Методы гильотинного раскроя (Guillotine Best Area Fit, Guillotine Best Short Side Fit, Guillotine Best Long Side Fit, Guillotine Worst Fit Rules),
- Методы наибольшего прямоугольника (Maximal Rectangle),
- Skyline методами (Skyline Bottom-Left, Skyline Best Fit).

Для решения задачи упаковки, как в случае с кругами, так и с прямоугольниками, существует несколько популярных методов:

Методы локального поиска [10]. Основываются на итеративном улучшении расположения объекта. Эффективность зависит от начальных параметров алгоритма, так как поиск может остановиться в локальном оптимуме, так и не достигнув глобального. К методам локального поиска относятся: Имитация отжига и Поиск с запретами.

Генетические алгоритмы [11]. Для генерации новых решений и последующего приближения к оптимальному, в исследованиях применяют генетические операторы скрещивания, мутации и отбора. Эффективность сильно зависит от выбора формата представления решения, функции приспособленности, операторов скрещивания и мутации.

Заключение

Таким образом, в результате проведённого исследования была рассмотрена задача оптимального размещения рабочих мест в комнате, с целью максимизации количества сотрудников. Проведён обзор академических исследований в данной области. В ходе исследования были рассмотрены две математические модели представления рабочих мест и сделаны выводы об их применении в разных ситуациях. Перечислены популярные методы, которые подходят для решения задач упаковки. В дальнейшем планируется реализация математической модели и одного из алгоритмов в виде программного продукта.

Список литературы

1. Anderson, C. Augmented space planning: Using procedural generation to automate desk layouts / Anderson C, Bailey C, Heumann A, Davis D. // International Journal of Architectural Computing.— 2018.— № 16(2). — С. 164–177.
2. Chen, C. Using genetic algorithm to automate the generation of open-plan office layout / Chen C, Chacón Vega RJ, Kong TL // International Journal of Architectural Computing.— 2020.— № 19(3). — С. 449–465.
3. Pereira, Office space allocation optimization. / Pereira, Rui & Cummiskey, Kevin & Kincaid, Rex. // IEEE Systems and Information Engineering Design Symposium, SIEDS10.— 2010. — С. 112–117.

4. Leng. Tell2Design: A Dataset for Language-Guided Floor Plan Generation. / Leng, Sicong & Zhou, Yang & Dupty, Mohammed & Joyce, Sam & Lu, Wei // Association for Computational Linguistics.— 2023.
5. Patrick, H. The Math of Social Distancing Is a Lesson in Geometry [Электронный ресурс] // Quanta magazine.— 2020. —URL: https://www.quantamagazine.org/the-math-of-social-distancing-is-a-lesson-in-geometry-20200713/ (дата обращения: 23.04.2024)
6. Кочетов, Ю. А. Задачи упаковки для двумерных прямоугольных предметов [Электронный ресурс] // Институт математики им. С. Л. Соболева СО РАН, Новосибирск.— 2022. — URL: https://persons.iis.nsk.su/files/persons/pages/rustep16dec22kochetov.pdf (дата обращения: 25.04.2024)
7. Coffman, E. G. Approximation algorithms for bin-packing–an updated survey / E. G. Coffman, M. R. Garey, D. S. Johnson // Algorithm design for computer system design, Springer, Vienna.— 1984. — С. 49–106.
8. Jukka, J. A Thousand Ways to Pack the Bin. A Practical Approach to Two-Dimensional Rectangle Bin Packing [Электронный ресурс] — 2010. — URL: https://github.com/rougier/freetype-gl/blob/master/doc/RectangleBinPack.pdf (дата обращения: 25.04.2024)
9. Lodi, A. Recent advances on two-dimensional bin packing problems / A. Lodi, S. Martello, and D. Vigo // Discrete Appl. Math.— 2002.— № . 123 (1–3). — С. 379–396.
10. Щербина, О. А. Метаэвристические алгоритмы для задач комбинаторной оптимизации (обзор) // Таврический вестник информатики и математики. 2014.— № 1 (24). — С. 56–72.
11. Подлазова, А. В. Генетические алгоритмы на примерах решения задач раскроя // Проблемы управления. 2008.— № 2. — С. 57–63.

УДК 004.031

Разработка современных приложений на основе микросервисной архитектуры

Горохов Андрей Сергеевич

студент кафедры Прикладной математики и информатики Казанского национального исследовательского технического университета

***Аннотация**: В статье рассматриваются вопросы разработки приложений на основе микросервисной архитектуры, где каждая часть программной системы представлена в виде отдельного самодостаточного компонента со своими функциями. Проводится анализ сравнения микросервисной и монолитной архитектур. Результатом исследование является решение об использовании микросервисной архитектуры в своих проектах.*

***Abstract**: The article deals with the issues of application development on the basis of microservice architecture, where each part of the software is presented as a separate self-sufficient component with its own functions. The comparison of microservice and monolithic architectures is analyzed. The result of the study is the decision to use microservice architecture in the projects.*

***Ключевые слова**: разработка программного обеспечения, микросервисная архитектура, монолитная архитектура, контейнеризация приложений, микрослужба, ограниченный контекст.*

***Keywords**: software development, microservice architecture, monolithic architecture, application containerization, microservice, limited context.*

Введение

Современная индустрия разработки программного обеспечения в последние годы переживает значительные изменения в связи с ростом требований к гибкости, масштабируемости и надежности систем. Исторически сложившиеся монолитные приложения стали неэффективными в условиях быстрого изменения бизнес-требований. Вследствие этого, разработчики все чаще обращаются к микросервисам как эффективному подходу к созданию сложных приложений [5].

Под микросервисом будем понимать независимую программу, которая выполняет четко очерченный набор функций и взаимодействует с другими приложениями через определенный абстрактный интерфейс [2].

Основная часть

Микросервисная архитектура — это подход к созданию приложения, подразумевающий отказ от единой, монолитной структуры и переход к небольшим независимым сервисам, которые взаимодействуют между собой по определенным API. Это означает, что вместо того, чтобы исполнять все ограниченные контексты на одном сервере, мы используем несколько небольших микрослужб, каждое из которых соответствует своему ограниченному контексту и взаимодействует с остальными службами по протоколам HTTP/HTTPS, WebSockets или AMQP.

Каждая микрослужба реализует бизнес-логику и специфические возможности того ограниченного контекста приложения, за который данная служба отвечает. Ограниченный контекст же представляет собой определенную границу внутри программной системы, которая инкапсулирует в себе набор бизнес-правил, моделей данных, функций и сущностей для описания и моделирования поддомена в более крупном приложении. Из этого можно сделать вывод, что концепция микрослужбы является производной от шаблона ограниченного контекста в предметно-ориентированном проектировании. Таким образом, микрослужба похожа на ограниченный контекст, но она также является распределенной службой. Это значит, что она создается как отдельный процесс для каждого ограниченного контекста и должна использовать упомянутые ранее распределенные протоколы [1].

Переход к микросервисной архитектуре определяет ряд задач, которые необходимо решать в процессе разработки. Рассмотрим основные из них:

Задача 1: Определение границ каждой микрослужбы

Для решения данной задачи необходимо выделить группы взаимосвязанных логических объектов в проектируемой системе. Нужно отталкиваться от того, что микрослужбы должны быть независимыми друг от друга. Один и тот же логический объект в разных микрослужбах может иметь разные значения.

Задача 2: Создание запросов, извлекающих данные из разных микрослужб

Часто возникает ситуация, что на одной странице у пользователя должны отображаться данные из разных микрослужб. Для решения данной задачи применяется паттерн — Агрегатор. Суть его состоит в том, что мы создаем еще один программный компонент, который будет являться шлюзом, инкапсулирующим в себе запросы к разным микрослужбам, агрегации этих запросов и выдача клиенту.

Задача 3: Обеспечение согласованности между разными микрослужбами

В рамках программной системы необходимо проводить бизнес-процессы, которые затрагивают более одной микрослужбы. Данную систему необходимо держать в согласованном состоянии. Микрослужба обладает только своим набором логических объектов и управляет сохранением данного набора в абстрактном хранилище. Управлять хранилищем другой микрослужбы запрещено в рамках микросервисной архитектуры. Для решения данной проблемы используются интеграционные события, которые должны обрабатываться асинхронно. Если в микрослужбе А произошло какое-то событие, то, если это необходимо, мы должны оповестить об этом микрослужбу В, которая обработает данное событие, тем самым поддерживая систему в согласованном состоянии.

Графически две архитектуры представлены на Рисунке 1.

Архитектура микросервисов ускоряет разработку и упрощает масштабирование, так как каждый сервис разрабатывается отдельно от других. Обычно, для каждого сервиса организуется своя команда разработчиков и создается отдельный git-репозиторий, что положительно сказывается на понимании кода и скорости разработки нового функционала. Также, архитектура микросервсиов позволяет рационально использовать ресурсы приложения. Например, если одна из микрослужб потребляет много памяти и долго обслуживает запросы, то мы можем развернуть еще один экземпляр данной службы или увеличить ее аппаратные средства, тем самым решив проблему точечно, не задевая другое приложение. Стоит отме-

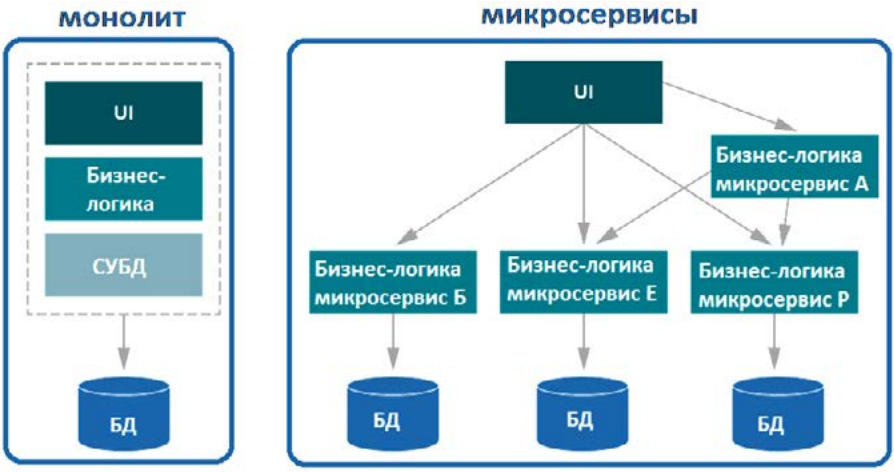

Рисунок 1. **Различие микросервисной и монолитной архитектуры**

тить еще одно важное преимущество — это отказоустойчивость. Если один из наших сервисов перестал работать, то остальное приложение продолжит функционировать и отвечать на запросы пользователей. Например, если у нас перестал работать микросервис “Заказов”, то пользователь не сможет заказать товар, но тем не менее он сможет просматривать товары, полученные от микросервиса “Товары” до момента, когда микросервис “Заказов” станет опять работоспособным. Это улучшает пользовательский опыт и поможет сохранить клиентов.

В монолитной архитектуре приложение представляет собой единое целое. Это означает, что если один процесс испытывает самую большую нагрузку от клиентов, то необходимо масштабировать все приложение. Также, единая кодовая база создает трудности при внедрении нового функционала. Например, когда над приложением работает большая команда разработчиков, которые будут внедрять свой код в одну кодовую базу. Это неминуемо приведет к конфликтам в git-репозитории, связанные с изменением одного и того же функционала. Эти конфликты необходимо будет решать, а это время. Также, любой сбой приложения в монолитной архитектуре неминуемо приведет к недоступности всего приложения. Тем не менее, монолитная архитектура проста в реализации, она не требует

дополнительных технологий, ведь взаимодействие осуществляется простыми вызовами функций.

Сравнительный анализ микросервисной и монолитной архитектуры представлен в Таблице 1.

Таблица 1

Аспект	**Монолитная архитектура**	**Микросервисная архитектура**
Разработка и развертывание	Одна кодовая база; простой процесс развертывания	Децентрализованная разработка; каждый сервис разрабатывается и развертывается по отдельности
Масштабируемость и производительность	Масштабируемость всего приложения; нерациональное использование ресурсов	Децентрализованная масштабируемость; оптимальное распределение ресурсов исходя из потребностей каждого сервиса
Обслуживание и расширяемость	Единая кодовая база; трудно внедрять новый функционал и поддерживать систему	Независимые сервисы; проще внедрять новый функционал, не задевая другие части приложения
Технологическое разнообразии и автономность	Ограниченный стек технологий	Возможность применять разнообразные технологии
Отказоустойчивость	Единая точка отказа	Изолированные точки отказа
Взаимодействие	Прямые вызовы функций; простые коммуникации и обмен данными	Необходимо применять дополнительные технологии для координации и согласованности данных

Исходя из сравнительного анализа, можно выделить следующие плюсы микросервисной архитектуры:

Гибкость и масштабируемость

Обновление приложения можно проводить по компонентам. Это значительно увеличивает гибкость и скорость разработки.

Мультиязычность и кроссплатформенность:

С микросервисами приложение независимо от конкретного языка программирования или технологии. Зачастую для сервисов использую разные языки программирования, лучше подходящих к решению определенной задачи.

Уменьшение ошибок

Поскольку каждый микросервис отвечает только за одну конкретную функцию, код становится более чистым и понятным, что уменьшает вероятность возникновения ошибок. Более того, изменения в одном микросервисе не влияют на остальные сервисы, что значительно упрощает поддержку и обновление приложения. Кроме того, микросервисы могут быть развернуты и масштабированы независимо друг от друга, что также способствует уменьшению ошибок. Например, если один из микросервисов перегружен, его можно легко масштабировать без влияния на работу других сервисов.

Упрощение и ускорение работы

Так как функциональность приложения разделена на отдельные микросервисы, каждый модуль становится проще понять для разработчиков. Изменения в одном модуле не требуют перепроектирования всего приложения.

Повышение стабильности

В случае возникновения проблемы, она не повлияет на всё приложение, а затронет только определенную его часть, оставляя остальную часть работоспособной. Это помогает обеспечить более высокую производительность для пользователей и исключает простой системы.

В микросервисной архитектуре разработчику часто приходится работать с несколькими микросервисами одновременно. Разворачивание всех этих

микросервисов на одной машине может быть неудобным, особенно если их много. Для более быстрого развертывания системы рекомендуется использовать виртуализацию, такую как виртуальные машины или контейнеры. Docker является платформой, использующей легковесные контейнеры, которая требует меньше системных ресурсов по сравнению с виртуальными машинами. В процессе развертывания часто используется собственный docker-репозиторий, где хранятся образы с готовыми микросервисами. Для настройки локальной инфраструктуры разработчику просто нужно запустить несколько docker-образов с уже готовыми микросервисами [3].

Необходимо учитывать влияние инструментов управления жизненным циклом приложений на работу с микросервисами. DevOps и микросервисы переносят автоматизацию с приложения на инфраструктуру, а средства управления конфигурациями значительно ускоряют настройку рабочей инфраструктуры по сравнению с ручной настройкой. При переходе от монолитного программного обеспечения к микросервисам можно упростить сопровождение конфигураций, воссоздавая копии системы на компьютерах разработчиков [4].

Заключение

Микросервисы пользуются популярностью благодаря высокой масштабируемости и возможности независимого изменения отдельных модулей приложения без простоев. Однако для полного извлечения выгоды из этой архитектуры требуются значительные инвестиции, большой опыт работы с различными технологиями и четкое понимание преимуществ перехода на микросервисы. Несмотря на то, что микросервисы являются передовой технологией, решение о выборе их в качестве основной архитектуры должно быть взвешенным.

Список литературы

1. Fowler M. Microservices — a definition of this new architectural term / Martin Fowler. 2014. [Электронный ресурс]. URL: http://martinfowler.com/articles/microservices.html. (Дата обращения: 19.04.2024).

2. Артамонов И. В. Исторические аспекты появления микросервисной архитектуры // Объектные системы. Ростов-на-Дону: Олейник Павел Петрович, 2016. С. 21–24. (Дата обращения: 19.04.2024).
3. Календарев А. Современная веб-архитектура. От монолита к микросервисам // Системный администратор. 2017. 1–2 (170–171). С. 80–83. (Дата обращения: 21.04.2024).
4. Ларрусеа Х., Сантамария И., Коломно-Паласьос Р., Эберт К. Микросервисы // Открытые системы. СУБД. 2018. 3. С. 10–12. (Дата обращения: 21.04.2024).
5. Ньюмен С. Создание микросервисов. СПб.: Питер, 2016. 304 с. (Дата обращения: 20.04.2024).

УДК 004.422.81

Программа планирования размещения виртуальных машин по физическим серверам на языке программирования Python

Сагиров Айнур Инсафович

студент кафедры Автоматизированных систем обработки информации и управления Казанского национального исследовательского технического университета имени А. Н. Туполева

Эминов Фарид Ибрагимович

кандидат технических наук, доцент кафедры Автоматизированных систем обработки информации и управления Казанского национального исследовательского технического университета имени А. Н. Туполева

***Аннотация**: В статье представлено описание функционирования программы, основанной на генетическом алгоритме, для эффективного размещения виртуальных машин (ВМ) на серверах в центрах обработки данных (ЦОД), а также результаты исследования задачи размещения ВМ по физическим серверам. Целью применения программы является повышение эффективности использования ресурсов серверов и улучшение производительности ВМ за счёт оптимального распределения нагрузки между физическими серверами. Описан вычислительный эксперимент оценки эффективности программы. Проведенные эксперименты и анализ результатов позволяют сделать вы-*

вод о преимуществах предложенного алгоритма перед распространенным существующим алгоритмом размещения виртуальных машин — BFD.

***Abstract**: The paper presents a description of the functioning of a program based on genetic algorithm for efficient placement of virtual machines (VMs) on servers in data processing centers (DPCs), as well as the results of the study of the task of VM placement on physical servers. The purpose of the program application is to increase the efficiency of server resource utilization and improve VM performance through optimal load balancing between physical servers. The computational experiment of the program efficiency evaluation is described. The conducted experiments and analysis of the results allow concluding about the advantages of the proposed algorithm over BFD — a common existing algorithm for VM placement.*

***Ключевые слова**: генетический алгоритм, виртуальные машины, физические серверы, центр обработки данных, виртуализация, размещение виртуальных машин.*

***Keywords**: genetic algorithm, virtual machines, physical servers, data center, virtualization, virtual machine placement.*

Введение

В 2023 году российские компании-интеграторы отметили рекордный рост спроса на создание ЦОД [1]. Факторы, способствующие росту, включают уход российских клиентов от западных облаков, нехватку высококлассного оборудования для создания собственной инфраструктуры [2]. Большинство проблем, связанных с отсутствием продукции западных производителей, уже успешно решены, однако остаются задачи по улучшению работы с дата-центрами.

Снижение затрат в ЦОД можно достичь за счет использования технологии виртуализации. До недавнего времени многие ЦОД использовали лишь 10% своей общей емкости. Однако виртуализация предоставляет возможность эффективнее загрузить работой доступное оборудование. Путем применения виртуализации ЦОД может увеличить использование оборудования до значений в районе 80% [3].

Программа эффективного размещения ВМ

Задача размещения ВМ состоит в процессе их эффективного закрепления за физическими серверами. Наиболее распространёнными показате-

лями эффективности являются критерии использования ресурсов в соответствии с соглашением об уровне услуг (SLA) [4].

В системах виртуализации на этапе планирования миграции могут использоваться различные алгоритмы размещения. Для выбора серверов для размещения ВМ получили распространения эвристические алгоритмы — FFD (First Fit Decreasing), BFD (Best Fit Decreasing) [5].

Разработана основанная на генетическом алгоритме программа эффективного размещения ВМ на языке программирования Python. Суть алгоритма в создании и эволюции популяции хромосом с использованием турнирного отбора, одноточечного скрещивания и мутации перетасовкой. В ходе эволюции находится лучшая хромосома, которая указывает на эффективный способ размещения ВМ по серверам. Лучшей хромосомой является та, у которой значение приспособленности минимальна.

Хромосомы в данном случае представляются в виде матриц. Строки матрицы (гены) — это списки, состоящие из одной единицы и нескольких нулей. Каждый ген указывает на размещение одной ВМ на определенный сервер.

Основной функцией программы является evolve, которая возвращает лучшую хромосому (эффективное распределение ВМ):

Код функции evolve:

```
def evolve(pop, f_list, vm_list, pm_list, tournament_size, elite_size,
population_size, max_generations, crossover_rate, mutation_rate):
# Инициализация лучшей хромосомы и ее значения приспособленности
best_fitness = math.inf
best_chromosome = []
# Цикл по поколениям
for generation in range(max_generations):
# Вычисление значений приспособленности для текущей популяции
fitness_values = calculate_fitness(pop, f_list, vm_list, pm_list)
# Выборка родителей, обновление лучшей хромосомы
# и ее значения приспособленности
selected, best_fitness, best_chromosome = tournament_selection(pop,
fitness_values, tournament_size, elite_size, best_fitness,
```

```
best_chromosome)
# Генерация новой популяции
new_population = []
while len(new_population) < population_size:
# Выборка двух родителей
parent1, parent2 = random.sample(selected, 2)
# Кроссовер для создания двух потомков
child1, child2 = one_point_crossover(parent1, parent2, cross-
over_rate)
# Мутация потомков
mutated_child1 = shuffle_mutation(child1, mutation_rate)
mutated_child2 = shuffle_mutation(child2, mutation_rate)
# Добавление мутировавших потомков в новую популяцию
new_population.extend([mutated_child1, mutated_child2])
# Замена старой популяции новой
pop = new_population
# Возвращаем лучшую хромосому
return best_chromosome
```

На вход функция evolve принимает значения констант, популяцию случайно сгенерированных хромосом (функция create_population), списки ВМ и физических машин (ФМ) с указанием ресурсов (CPU и RAM), рассчитанную матрицу критериев (функции calculate_f_res, calculate_f_sla, create_u_matrix, create_f_matrix) для каждой пары ВМ и ФМ популяции.

Внутри функции evolve происходит вызов функций tournament_selection (турнирный отбор с элитизмом), one_point_crossover (одноточечное скрещивание), shuffle_mutation (мутация с перемешиванием), calculate_fitness (расчет приспособленности).

Вычислительный эксперимент

Проведена серия вычислительных экспериментов для оценки разработанного генетического алгоритма.

В ходе экспериментов задавались различные количества ВМ и ФМ. Количество ресурсов CPU и RAM для ВМ и ФМ генерировались случайным образом. Количество всего доступных CPU ФМ бралось случайно из значений [32, 48, 64, 80, 96], RAM — [128, 256, 512]. Количество занятых CPU ФМ выбиралось случайно из нормального распределения со средним значением 16 и стандартным отклонением 4. Аналогично для RAM — среднее значение 32, стандартное отклонение 8. Так же для ресурсов ВМ: CPU — среднее значение 10, стандартное отклонение 4, RAM — среднее значение 20, стандартное отклонение 6.

В экспериментах проводилось сравнение генетического алгоритма и алгоритма BFD с сортировкой по CPU. В начале каждого эксперимента случайно задавались ресурсы ФМ, затем в цикле из 100 итераций, генерировались случайные ВМ и размещались по ФМ с помощью генетического алгоритма и BFD. В конце каждого эксперимента рассчитывалось сред-

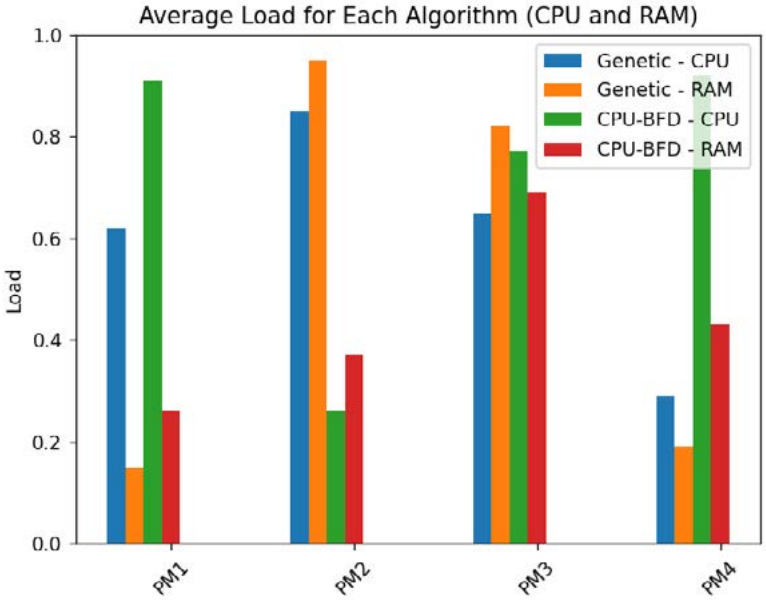

Рисунок 1. **Диаграмма средних загрузок ФМ для каждого алгоритма**

няя загрузка ФМ по ресурсам в процентах. Итог по загрузкам выводилось в виде диаграммы.

На рисунке 1 приведена диаграмма загрузок одного из экспериментов.

Результаты данного эксперимента типичны и для остальных из серии.

Исходя из результатов экспериментов, можно сделать вывод, что генетический алгоритм распределяет ВМ по ФМ более равномерно и практически не нарушает SLA производительности процессора. Также генетический алгоритм более эффективно загружает сервера, так как в среднем разница между загрузками CPU и RAM ФМ для генетического алгоритма меньше, чем у CPU-BFD.

Заключение

В результате выполнения данной работы была создана основанная на генетическом алгоритме программа размещения виртуальных машин по физическим серверам на языке программирования Python. Результаты проведённых экспериментов показывают эффективность генетического алгоритма при размещении ВМ по физическим серверам.

Список литературы

1. Тенденции на российском рынке дата-центров URL: https://www.tadviser.ru/a/265386 (дата обращения: 8.04.24).
2. Уход западных вендоров принес рекордную выручку российским провайдерам IaaS. Обзор: Облачные сервисы 2023 — Cnews URL: https://www.cnews.ru/reviews/oblachnye_servisy_2023/articles/uhod_zapadnyh_vendorov_prines_rekordnuyu (дата обращения: 8.04.24).
3. Канатьев К. Н. Анализ основных аспектов виртуализации / К. Н. Канатьев, А. А. Бусенков, В. Н. Большаков, О. Д. Куприков, А. С. Синюхин // Инновации и инвестиции. 2022. № 4. URL: http://ej.kubagro.ru/2019/09/pdf/07.pdf (дата обращения: 8.04.2024).
4. Тутов А. В. Многокритериальная оптимизация размещения виртуальных машин по физическим серверам в облачных центрах обработки данных / А. В. Тутов, Н. В. Тутова, А. С. Ворожцов, И. А. Андреев // T-Comm. 2021. № 1.

URL: https://cyberleninka.ru/article/n/mnogokriterialnaya-optimizatsiya-razmescheniya-virtualnyh-mashin-po-fizicheskim-serveram-v-oblachnyh-tsentrah-obrabotki-dannyh (дата обращения: 8.04.2024).
5. Moges F. Energy-aware VM placement algorithms for the OpenStack Neat consolidation framework / F. Moges, S. Abebe // J Cloud Comp 8, 2, 2019.

УДК 004.65

Инфологическое проектирование базы данных для программной системы формирования и анализа цифрового профиля студента, формируемого посредством прохождения анкет

Калеев Данил Андреевич

студент Казанского национального исследовательского технического университета имени А. Н. Туполева

***Аннотация**: Статья посвящена инфологическому проектированию базы данных для программной системы формирования и анализа цифрового профиля студента, который создается на основе прохождения анкет. Рассматриваются основные таблицы базы данных, их атрибуты и связи между ними. Результатом является сформированная инфологическая схема базы данных.*

***Abstract**: The article is devoted to the infological design of a database for a software system for forming and analyzing a student's digital profile, which is created based on passed questionnaires. The main tables of the database, their attributes, and relations between them are considered. The result is the formed infologic scheme of the database.*

***Ключевые слова**: разработка программного обеспечения, проектирование базы данных, цифровой профиль, анкетирование студентов.*

***Keywords**: software development, database design digital profile, student survey.*

Введение

Современный мир характеризуется высокой степенью информатизации и автоматизации всех процессов. В условиях быстрого развития

технологий и появления новых сервисов и продуктов, важным становится не только предоставление услуг, но и учет индивидуальных потребностей каждого. Это позволяет создать уникальный опыт взаимодействия и повысить уровень удовлетворенности пользователей. Для подобных целей и следует использовать цифровой профиль.

Под цифровым профилем будем понимать набор данных, который описывает интересы, предпочтения, характеристики и иная информация о пользователе.

Следует сразу оговорить, что эта статья не является подробной инструкцией и не утверждает, что данный пример проектирования является единственно верным.

Основная часть

База данных (БД) — совокупность данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными, независимая от прикладных программ [1].

«Проектирование базы данных (БД) — одна из наиболее сложных и ответственных задач, связанных с созданием автоматизированных информационных систем.» [2, с. 117].

Определим основные и вспомогательные таблицы базы данных, а также связи между ними.

Основными таблицами базы данных будут следующие:

1. Таблица «Студенты» (Students) содержит всю необходимую для функционирования системы информацию о студентах. Для возможности гибко настраивать анкеты нам необходимо знать о студенте — его ФИО, направление подготовки, семестр обучения.
2. Таблица «Анкеты» (Questionnaire) — информация об анкетах, что есть в системе. Нам необходимы такие данные, как название, возможно ли повторно пройти анкету, видима ли сейчас анкета для студентов в системе или нет.
3. Таблица «Вопросы» (Questions) содержит в себе текст вопроса, тип вопроса (вопросы могут быть с одним вариантом ответа (альтернативные) и с несколькими, а также открытыми (пользователь сам пишет ответ))

4. Таблица «Ответы» (Answers) содержит в себе текст ответа и ссылается на вопрос, на который отвечает.
5. Таблица «Характеристики» (Characteristics). Чтобы не усложнять схему базы данных данная таблица содержит не только характеристики, которыми могут обладать студенты (к примеру, «умение работать в команде»), но и предметы, которыми интересуется студенты (к примеру, «интересуется программированием на c#»). Имеет следующие атрибуты — Описание, тип (тип определяет является ли это чертой личности или же интересом к какой-либо дисциплине), минимальное и максимальное значения данной характеристики.
6. Таблица «Прогрессы студентов» (StudentProgresses). Данная таблица ссылается как на студента, так и на анкету, помимо этого необходимо хранить данные о позиции последнего отвеченного вопроса и завершено ли было прохождение.

В вспомогательные таблицы выделяются следующие:

1. Таблица «Вопрос анкеты» (TestQuestionPositions). Необходимо ссылаться на анкеты и вопросы, дополнительно хранив позицию каждого вопроса в анкете.
2. Таблица «Ответы студентов». Ссылается на прогресс студента и ответы. В случае, если при повторном прохождении анкеты старые ответы студентов не удаляются, стоит ввести дополнительные данные актуальности ответа.
3. Таблица «Влияние ответа на характеристики» (AnswerCharacteristicValue). Помимо ссылок на ответ и характеристику содержит в себе влияние каждого ответа на заданную характеристику. В последствии в программной системе после прохождения всей анкеты подразумевается учет влияния каждого ответа для формирования цифрового профиля.
4. Таблица «Характеристики студентов» (StudentCharacteristics). Таблица, с помощью которой выводится пользователю его цифровой профиль. Ссылается на студента и характеристики, хранит данные о степени выраженности той или иной характеристикой, которая учувствовала в анкетах, что проходил студент.

Инфологическая схема спроектированной базы данных представлена ниже.

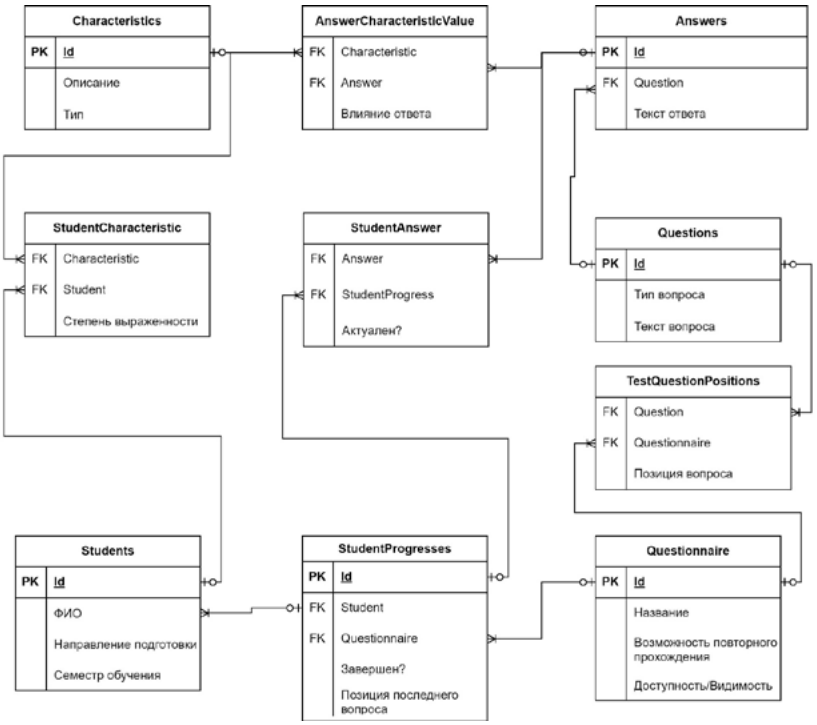

Рисунок 1. **Инфологическая схема**

Заключение

Таким образом, в данной статье были выявлены основные и вспомогательные таблицы базы данных для программной системы формирования и анализа цифрового профиля студента, определены какие данные должны храниться, а также сформирована инфологическая схема проектируемой базы данных.

Список литературы

1. ГОСТ 20886–85. Организация данных в системах обработки данных. Термины и определения [Текст]. — Взамен ГОСТ 20886–75; введ. 1986–07–01. — М.: Стандартинформ, 2005.

2. Гусева Т.И., Башин Ю. Б. Проектирование баз данных в примерах и задачах.-М.: «Радио и связь», 2005.

УДК 004.422:004.78

Лучшие практики контейнеризации для развертывания микросервисов в системах высокой нагрузки

Нуржанкызы Асем

магистрант Казахстанско-Британского технического университета (Республика Казахстан, Алматы)

***Аннотация**: В современном мире растущая сложность и требования к масштабируемости делают развертывание микросервисов в системах высокой нагрузки ключевой задачей для разработчиков и инженеров. Контейнеризация, особенно с использованием Docker, стала широко применяемым подходом к упаковке, доставке и запуску приложений, предоставляя преимущества в масштабируемости, портабельности и управлении зависимостями. В данной статье мы рассмотрим лучшие практики контейнеризации для эффективного развертывания микросервисов в системах высокой нагрузки. Представленные в статье рекомендации и примеры позволят разработчикам и инженерам создавать стабильные, масштабируемые и отказоустойчивые архитектуры микросервисов, способные эффективно работать в условиях высокой нагрузки.*

***Abstract**: In today's world, increasing complexity and scalability requirements make deploying microservices in high load systems a key challenge for developers and engineers. Containerization, especially using Docker, has become a widely adopted approach to package, deliver, and run applications, providing advantages in scalability, portability, and dependency management. In this paper, the authors explore containerization best practices for efficient deployment of microservices in high-load systems. The guidelines and examples presented in the paper will enable developers and engineers to create stable, scalable, and fault-tolerant microservices architectures that can operate efficiently in high load environments.*

***Ключевые слова**: микросервисы, контейнеризация, Docker, масштабируемость, портабельность, управление зависимостями, лучшие практики, высокая нагрузка.*

***Keywords**: microservices, containerization, Docker, scalability, portability, dependency management, best practices, high load.*

Введение

В условиях растущей сложности и требований к масштабируемости, развертывание микросервисов в системах высокой нагрузки становится ключевой задачей для современных разработчиков и инженеров. Контейнеризация, особенно с использованием Docker, стала широко применяемым подходом к упаковке, доставке и запуску приложений, предоставляя преимущества в масштабируемости, портабельности и управлении зависимостями.

В данной статье мы рассмотрим лучшие практики контейнеризации для эффективного развертывания микросервисов в системах высокой нагрузки. Мы изучим методы оптимизации и упрощения развертывания микросервисов с использованием контейнеризации, рассмотрим основные преимущества и нюансы применения Docker в контексте высоконагруженных систем. Представленные в статье рекомендации и примеры позволят разработчикам и инженерам создавать стабильные, масштабируемые и отказоустойчивые архитектуры микросервисов, способные эффективно работать в условиях высокой нагрузки.

Кроме того, мы проанализируем сценарии применения контейнеризации для масштабирования и управления микросервисами, рассмотрим методы обеспечения безопасности и мониторинга контейнеризованных приложений в контексте высоконагруженных систем. Надеемся, что эта статья станет полезным ресурсом для специалистов, работающих с микросервисными архитектурами и стремящихся к оптимальной работе при высоких нагрузках.

Использование отдельных контейнеров для каждого микросервиса

Одной из основных практик является использование отдельных контейнеров для каждого микросервиса [1]. Это позволяет изолировать каждый сервис, обеспечивая независимость в развертывании и масштабировании. Каждый контейнер содержит только необходимые зависимости и код, минимизируя размер и упрощая управление.

Эта практика снижает взаимное влияние микросервисов друг на друга, предотвращает «коллапс песочницы» (sandbox collapse) и позволяет эффективно масштабировать каждый сервис в зависимости от его нагрузки. Кроме того, использование отдельных контейнеров упрощает обновление и масштабирование приложения в целом, так как изменения в одном микросервисе не требуют пересоздания всех контейнеров.

Дополнительно, этот подход обеспечивает легкость в управлении зависимостями и версиями каждого микросервиса. Различные версии одного и того же сервиса могут существовать параллельно, не влияя на работу других сервисов. Такая гибкость позволяет управлять обновлениями и исправлениями в отдельных компонентах системы без перерывов в работе всего приложения.

Основные Преимущества

Изоляция и независимость

Каждый микросервис работает в своем собственном контейнере, изолированном от других сервисов, что обеспечивает независимость в развертывании и масштабировании [2].

Минимизация размера

Контейнер содержит только необходимые зависимости и код, что минимизирует его размер и упрощает управление ресурсами [3].

Эффективное масштабирование

Благодаря отдельным контейнерам, каждый микросервис может быть масштабирован независимо от других, обеспечивая оптимальное распределение нагрузки [4].

Упрощенное обновление и управление

Изменения в одном микросервисе не требуют пересоздания всех контейнеров, что упрощает обновление и масштабирование приложения в целом [5].

Гибкое управление зависимостями и версиями

Различные версии микросервисов могут существовать параллельно, позволяя эффективно управлять обновлениями и версиями каждого компонента системы [6].

Управление ресурсами и ограничения

Для обеспечения стабильности системы в условиях высокой нагрузки крайне важно эффективно управлять ресурсами каждого контейнера. Использование инструментов, таких как Docker Compose или Kubernetes, позволяет устанавливать ограничения на использование CPU, памяти и других ресурсов для каждого микросервиса [7]. Это предотвращает «доминирование» одного сервиса над другими и обеспечивает равномерную балансировку нагрузки.

Использование Docker Swarm или Kubernetes для оркестрации

Оркестрация контейнеров является неотъемлемой частью развертывания микросервисов в системах высокой нагрузки. Docker Swarm и Kubernetes предоставляют мощные инструменты для автоматизации развертывания, масштабирования и управления контейнерами [8]. Они обеспечивают высокую доступность, отказоустойчивость и возможность управления микросервисами на различных уровнях абстракции.

Резервное копирование и мониторинг

При работе с микросервисами в системах высокой нагрузки критически важно иметь механизмы резервного копирования и мониторинга [9]. Автоматизированные резервные копии контейнеров и их конфигураций позволяют быстро восстановить систему в случае сбоя. Системы мониторинга, такие как Prometheus или Grafana, предоставляют важные метрики по использованию ресурсов, производительности и состоянию контейнеров, что позволяет оперативно реагировать на изменения и проблемы [10].

Заключение

Внедрение лучших практик контейнеризации для развертывания микросервисов в системах высокой нагрузки существенно повышает эффективность, управляемость и отказоустойчивость системы. Использование отдельных контейнеров, управление ресурсами, оркестрация и мониторинг — ключевые составляющие успешного развертывания. Мы надеемся, что данная статья поможет разработчикам и инженерам создать более надежные и масштабируемые микросервисные архитектуры.

Список литературы

1. Burns, B., & Vohra, A. (2016). Kubernetes: Up and Running: Dive into the Future of Infrastructure.
2. O'Reilly Media. [1] Pahl, C. (2016). Docker: Up & Running: Shipping Reliable Containers in Production. O'Reilly Media.
3. Leitner, P., & Cito, J. (2018). Microservices in the Cloud: Native integration patterns for architecture. O'Reilly Media.
4. Loukides, M. (2017). What Is DevOps? O'Reilly Media.
5. Meilala, E., & Pääkkönen, P. (2020). "Microservices in High-Load Systems: Challenges and Best Practices." Conference Proceedings on Software Engineering.
6. Newman, S. (2015). Building Microservices: Designing Fine-Grained Systems. O'Reilly Media.
7. Ramel, S. (2019). "High-Performance Microservices: Best Practices." Journal of Microservice Architecture.
8. Burns, B., & Gopinath, A. (2017). "Containerization at Scale: Exploring Kubernetes' Massive Ecosystem." Conference Proceedings on Cloud Computing.
9. Fowler, M. (2014). "Microservices: A Definition of This New Architectural Term." Website: https://martinfowler.com/articles/microservices.html.
10. Linton, J. (2018). "Containerization with Docker: Simplifying Microservice Deployment".

УДК 004

Beyond the last mile: TSP and linear regression for enhanced delivery optimization

Юлдаш Сухраб

студент магистратуры Казахстанско-Британского технического университета (Республика Казахстан, Алматы)

***Abstract**: This paper discusses the application of optimization models in the field of delivery logistics, focusing on the traveling salesman problem (TSP) and linear regression. The travel-*

ing salesman problem solves the efficient routing of deliveries across multiple locations, while linear regression analyzes the relationships between variables such as delivered, undelivered, and canceled orders. Using platforms such as SAS Studio and Power BI, these models are estimated using a dataset taken from Kaggle covering various delivery settings. Optimization of the traveling salesman problem, guided by a genetic algorithm, achieves a near-optimal route with a total distance of 20. In contrast, linear regression analysis involving multiple variables reveals statistically significant results, but emphasizes the difficulties of predicting outcomes. The combination of both models offers a comprehensive understanding of optimization problems and predictive analytics in delivery logistics, informing decision making to improve operational efficiency. Moreover, this study contributes to the existing literature by presenting a comparative analysis of the two models, clarifying their strengths and limitations in addressing the complex challenges of modern delivery logistics.

***Аннотация**: В данной статье рассматривается применение моделей оптимизации в области логистики доставки, с акцентом на задаче о коммивояжере (TSP) и линейной регрессии. Задача о коммивояжере решает эффективное маршрутизирование доставок по множеству местоположений, в то время как линейная регрессия анализирует взаимосвязи между переменными, такими как доставленные, недоставленные и отмененные заказы. Используя платформы такие как SAS Studio и Power BI, мы оцениваем эти модели с использованием набора данных, взятого с Kaggle, охватывающего различные параметры настроек доставки. Оптимизация задачи о коммивояжере, направляемая генетическим алгоритмом, достигает практически оптимального маршрута с общим расстоянием 20. В отличие от этого, анализ линейной регрессии, включающий несколько переменных, раскрывает статистически значимые результаты, но подчеркивает сложности прогнозирования результатов. Сочетание обеих моделей предлагает комплексное понимание задач оптимизации и прогностической аналитики в логистике доставки, информируя принятие решений для улучшения операционной эффективности. Более того, данное исследование вносит вклад в существующую литературу, представляя сравнительный анализ двух моделей, проясняя их сильные стороны и ограничения в решении сложных задач современной логистики доставки.*

***Keywords**: TSP, delivery, linear regression, optimization models.*

***Ключевые слова**: TSP, доставка, линейная регрессия, модели оптимизации.*

Introduction

In the dynamic and rapidly changing field of delivery logistics, route management expertise and a sophisticated understanding of critical influencers are critical to maximizing operational effectiveness. This paper takes readers on a

thorough analysis of optimization models in the context of a delivery customization challenge, with a special emphasis on two extremely important methods: linear regression and the traveling salesman problem (TSP). The TSP is a well-known problem in operations research and computer science that addresses the difficult task of choosing the most effective delivery path between several sites for an all-encompassing last-mile approach. Simultaneously, a basic statistical technique called Linear Regression explores the world of predictive modeling by examining the correlations between variables like fulfilled orders, undelivered orders, and cancelled orders. By combining these models, we want to provide a comprehensive view of predictive analytics and route optimization, which will help close the gap between strategic decision-making and the complex details of actual delivery situations. By means of this dual investigation, in which algorithmic effectiveness and predictive precision are connected together, this research aims to provide insightful knowledge as well as a sophisticated comprehension to the broad domain of delivery logistics. It aims to support well-informed decision-making, offering a strong base for improving operational performance in the face of the constantly shifting demands of the contemporary delivery environment.

Literature review

The difficulties in combining last-mile delivery and first-mile collection into joint routes, highlighting the necessity for study in bi-directional flow design that takes time window restrictions and vehicle capacity into account. Using a data analysis method, the work presents a novel viewpoint by establishing the City Traveling Salesman Problem (CTSP) [1], in which numerous salesman have distinct city sets. It also suggests closed-form adjustment factors capturing non-trivial efficiency impacts [2]. In order to solve CTSP, the authors provide two enhanced genetic algorithms: Hill-Climbing GA and Greedy GA. They show that the Hill-Climbing GA performs better in terms of convergence rate and solution quality. Finally, focuses on the classic Traveling Salesman Problem (TSP) and suggests a better genetic algorithm based on dynamic mutation and random crossover, demonstrating faster convergence times and better optimum solutions than conventional genetic algorithms [3].

When taken as a whole, these publications offer an extensive perspective on routing effectiveness, a variety of issue structures, and algorithmic improvements, all of which expand optimization techniques in logistics and transportation planning. creating efficient machine learning algorithms and assessing different regression models to estimate agricultural production. Based on important metrics like the R2 score and Mean Absolute Error (MAE), the study finds that the two most effective models are the Random Forest Regression and the XGBoost Regression. These models are further refined by hyperparameter optimization, and for the particular dataset, Random Forest Regression is shown to be the most effective model [4].

The widespread practice of organizations, especially logistics firms, to include innovation in a range of domains, including organizational, technical, technological, and product domains. The importance of contemporary solutions like Industry 4.0, Big Data, and the Internet of Things is emphasized in the article as a means of satisfying consumer needs and improving supply chain and logistics management [5]. In Shanghai, China, a significant automaker is putting the concept into practice. Validation will come from further case studies. The research presents new DSCM value propositions and uses SAP technologies to demonstrate how they may be implemented in practice [6].

The effect of smart logistics technologies on the delivery of goods across several modes of transportation, highlighting the necessity of smart technologies and outlining a single plan for smart logistics suppliers. the use of several approaches, such as data analysis, simulation models, and optimization suggestions, to the optimization of warehouse operations. The study demonstrates how optimization interventions can result in more effective warehouse procedures, such as product classification and warehouse layout [7][8]. Supply chain risk management (SCRM) trends are analyzed using bibliometric measures, which highlight important themes and recommend integrating cutting-edge technologies like Industry 4.0, Big Data, IoT, and AI into SCRM. The paper also identifies shortcomings in the literature mapping technique and emphasizes the development of sustainable supply chain management [9].

The intricacy of optimizing freight routes, advising against simplifying and supporting the application of several methods for comparison. It highlights the availability of precise and quick optimization with freely available tools and emphasizes the need for trained staff for successful optimization [10].

pNSGA-II is a unique computational model inspired by Physarum and based on a genetic algorithm. When compared to previous algorithms, this technique solves bi-objective symmetric TSP better. The paper makes recommendations for future developments to lower computing complexity and broaden the model's use in multilayer network community detection [11].

TLGA, a two-level genetic algorithm, is presented for the Capacitated Traveling Salesman Problem (CTSP), showing that it can provide shorter tours than existing techniques. Potential applications in large TSPs for more efficient solutions are provided by TLGA's incorporation of evolutionary optimization for both levels [12].

Main provisions

The aim of this article is to analyses the delivery customs work according to the two optimization models: Traveling Salesman Problem (TSD) and Linear Regression (LR).

1. To explore existing services (Power BI, SAS Studio etc.) that helps to analyses and apply different types of algorithms to the chosen dataset.
2. To choose a dataset according to the topic, develop a program to apply our optimization methods.
3. To test program and analyses results.

Materials and Methodology

1. SAS Studio vs Power BI

Designed to provide users with an environment that is easy to use for statistical analysis, data management, and reporting, SAS Studio is an interactive web-based interface specifically made for SAS (Statistical Analysis System) software. Data scientists, analysts, and researchers may access it easily via a web browser, which does away with the need for conventional desktop installs. This dynamic tool caters to both non-programmers and programming aficionados by combining a code editor with a point-and-click interface. SAS Studio is unique in that it supports a wide range of programming languages, such as Python, SQL, and

SAS programming. It also excels in data exploration, statistical modeling, and visualization. It encourages cooperation by allowing users to exchange and work together on SAS projects in a cloud-based setting, going beyond solitary chores. Essentially, SAS Studio is a powerful and adaptable application that makes it easier to analyze data effectively and make wise decisions in the broad field of statistical analytics.

A powerful corporate analytics tool from Microsoft, Power BI helps users turn a wide range of information into actionable insights. With drag-and-drop interfaces, Power BI facilitates easy exploration and analysis of several data sources, including databases and cloud applications, while also enabling seamless integration. Using dynamic graphs and charts, users may create visually attractive dashboards and reports.

The tool's powerful data modeling features, enabled by DAX expressions, facilitate the creation of connections between data tables and the execution of intricate computations. Moreover, Power BI facilitates smooth collaboration and real-time analytics, allowing users to exchange insights both inside and outside of their companies. Power BI is scalable for small- and enterprise-scale data analytics projects because to its connection with Microsoft Azure services, which makes it a flexible option for promoting data-driven decision-making across a range of business sectors.

Although SAS Studio and Power BI are both strong tools, they each offer unique capabilities and applications, as well as benefits and drawbacks. It is noteworthy that the decision between SAS Studio and Power BI is contingent upon particular needs and preferences. When compared to SAS Studio, Power BI may have the following drawbacks:

Advanced Analytics skills: SAS Studio is well known for its statistical modeling and machine learning skills, which may even be more complicated and analytically thorough than Power BI. While Power BI offers certain statistical features, its primary concentration is on reporting and visuals. As a result, it could not provide as many sophisticated analytical approaches as SAS Studio.

Data Governance and Security: SAS Studio frequently offers strong data governance and security capabilities since it is a component of the larger SAS ecosystem. Even though Power BI is secure, it could need extra setups for businesses with strict data governance guidelines, particularly when handling regulated or sensitive data.

Programming Flexibility: SAS Studio offers a flexible environment for statistical analysis and data manipulation through programming. It is specifically developed for SAS programming. However, Power BI has more of an emphasis on a drag-and-drop interface for visualizations, which may restrict the range of sophisticated programming operations that can be completed within the tool.

Pricing Structure: Power BI has a subscription-based pricing structure, and although it has a free edition, some advanced capabilities can need extra license fees. Based on their unique requirements and use patterns, enterprises should carefully consider the financial implications of implementing SAS Studio, as pricing structures may vary depending on the deployment.

2. Dataset

The dataset I provide in the article is intended to fill in certain information gaps and enable meaningful analysis. Found by searching through a large dataset on the open-source Kaggle platform, this dataset is an important tool for optimizing delivery customs. It is available in csv file, covering a huge set of data. The dataset comprises order id, rider's id, first mile distances where it's road distance from rider's location to the pickup location, last mile distances where it's road distance from pickup location to the delivery location, weight of packages, delivered orders, undelivered orders and cancelled orders. Throughout this article, I detail the dataset's discovery process, describe its key characteristics, and outline any data preprocessing steps undertaken to ensure reliability.

3. Results and discussion

3.1. Traveling Salesman Problem (TSP)

A widely recognized optimization conundrum in operations research and computer science is Curious Traveling Salesman Issue (CTS). It requires deciphering the superior path for a salesman to journey in order to tour a cluster of cities precisely once before circling back to the starting place. The objective is to decrease the expedition's comprehensive length or expenditure. CTS is a timeless combinatorial optimization predicament with broad applications in circuit draft-

ing, logistics, and transit arrangement. CTS is an NP-hard dilemma for the reason that, despite its seemingly straightforward hypothesis, locating an optimum resolution becomes tougher as the number of cities rise. CTS is processed by a multitude of algorithms, including hunches and exact techniques, with its modifications being vital in addressing tangible routing and scheduling predicaments.

Several optimization models could be utilized in SAS Studio to solve the Travel Salesman Problem (TSP), which involves finding a salesman's most efficient path between a given set of locations. SAS Optimization provides a versatile setting for addressing TSP using various models. TSP is formulated as a binary optimization problem utilizing the Integer Linear Programming (ILP) paradigm, aiming to minimize the overall distance traveled under some restrictions. Moreover, users can add quadratic goal functions and restrictions using SAS Studio's support for the Constrained Quadratic Programming (CQP) model for TSP. These models make use of SAS Studio's robust optimization algorithms, permitting practitioners to quickly and effectively identify nearly optimal solutions for TSP situations!!! SAS Studio's capability to experiment with several TSP models offers a flexible framework for handling challenging routing problems and optimizing travel paths in diverse applications.

In this test, there are cities that located in on a grid. The cost of delivery from one city to another is given according to distance covered. The cost coefficients (the "distances") are stored in the distances. The entire distance on the ideal path is 20.

```
15 id = gasetup(3,
16                20,
17                1234);
18
19 call gasetobj(id,
20                 2,
21                 coeffs);
22
23
24 call gainit(id,
25              400);
```

Figure 1. **Problem Setup and Set Function.**

For this study, there are 20 locations within 1234 initial seed and 400 initial population size. Moreover, with 30 iterations to get the final and best result Figure 1.

The algorithm uses the default parameters as an elite value of 1 and a conventional tournament of size 2 because there is no GASETSEL. Additionally, the genetic operators are using the 'order' operator for crossover and the 'invert' operator for mutation since there is no GASETCRO or GASETMUT call Figure 2. The default mutation probability is 0.05.

```
niter = 30;
BestValue = j(niter,1);

call gagetval(value, id, 1);
BestValue[1] = value;

do i = 2 to niter;
   call garegen(id);
   call gagetval(value, id, 1);
   BestValue[i] = value;
end;
```

Figure 2. **Loop Execution.**

Iteration	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
BestValue	58	58	54	53	53	52	49	46	46	46	39	39	37	37	37	37	37	37	37	37	37	37	37	32	32	32	32	32	32	32

Final Best Value
32

Figure 3. **Iteration Results**

As a result, there are different values for each iteration as shown in Figure 3; for first and second iteration, the best value is 58, 54 in third, 53 in fourth and fifth, 52 in sixth, 49 in seventh, 46 in eighth-ninth-tenth, 39 in eleventh-twelfth, 37 in fifteenth-twenty third, and the is final best value of 32 after 30 iterations.

3.2. Linear Regression with Multiple Variables

In SAS Studio, There is a modeling method called linear regression and it's used for investigate and measure the associations of a dependent and one or many independent variables. When do linear regression analysis, SAS Studio offers a user-friendly environment that enables users to fit linear models, evaluate predictor significance, and generate predictions based on observed data. Analysts may quickly import and modify datasets; define the model structure, and analyze

important statistical metrics like coefficients, p-values, and R-squared values by utilizing SAS Studio's user-friendly interface! For users who would like work with a graphical interface, the tool provides point-and-click capabilities; for those who would rather code! it gives the freedom of SAS programming. SAS Studio's extensive feature set makes it possible model linear regression effectively and insightfully for a variety of analytical applications.

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	3732204931	1244068310	30.50	<.0001
Error	488	19904105302	40787101		
Corrected Total	491	23636310234			

Root MSE	6386.47798	R-Square	0.1579
Dependent Mean	7968.11382	Adj R-Sq	0.1527
Coeff Var	80.15044		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	11023	627.50209	17.57	<.0001
delivered_orders	1	-3.71142	9.60582	-0.39	0.6994
cancelled	1	-11.37166	25.76211	-0.44	0.6591
undelivered_orders	1	-24.49433	25.75908	-0.95	0.3421

Figure 4. **Linear Regression Analyses**

Results in linear regression which is multiple regression with 3 independent variables (delivered orders, undelivered orders and cancelled orders) and intercept p-value is significance which means this coefficient is statistically significantly different from zero (<.001) and that is very good outcome as shown in the figure 4. Additionally, r-square is 0.1579 and it is shown about 16% significance that is not very good but normal and adj r-square is 0.1527 with 15% which come out with normal situation.

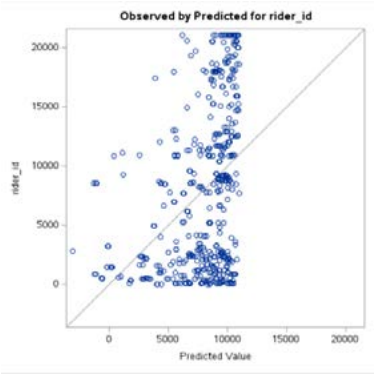

Figure 5. **Prediction Graph**

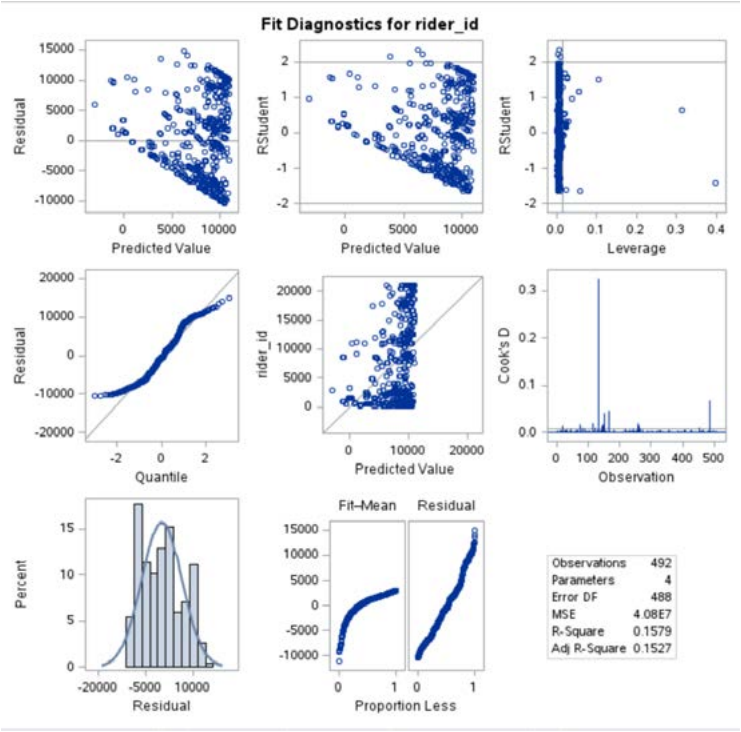

Figure 6. **Fit Diagnostics.**

As show in Figure 5, if we look for predicted observations of rider then the most of predicted values are not so close and far from the line because of having more than one independent variables in our study. Moreover, fit diagnostics results for rider is quite significant where predicted values far with having residual vice versa nearer in quantile values significantly. The percentage of residuals are up to 16% and it's quite good as shown in Figure 6. The time taken by linear regression process in dataset is 1.74 seconds in real time with 0.95 seconds in user CPU time.

Results and Discussion

The outcomes of the Traveling Salesman Problem (TSP) and the straight regression analysis with multiple variables reveal distinct insights for sure. The TSP optimization was done using a genetic algorithm with a seed of 1234, a population size of 400, and 200 iterations instead of 30, reaching a final best value of 32, achieving an optimal delivery route with a total distance of 20 is quite remarkable. In contrast, the linear regression analysis, utilizing three independent variables (delivered orders, undelivered orders, and cancelled orders), demonstrated statistically significant results with the intercept. Nonetheless, the modest R-squared value of 0.1579 indicated that only approximately 16% of the variability in the dependent variables was explained, highlighting the challenges of predicting complex outcomes with multiple variables. Despite these differences, both analyses contribute valuable insights into optimization and predictive modeling, showcasing the trade-offs and nuances inherent in addressing diverse analytical challenges.

The TSP results really demonstrate how well the genetic algorithm can optimize the delivery routes, producing a close to optimum solution given the given restrictions. Even though the findings of the linear regression analysis were statistically significant, they brought to light to the difficulties predicting the dependent variable from a variety of independent factors. The weak R-squared values do suggests that unaccounted variables can may be the reason for a significant percentage of the variability that can not be explained. With this, the fit diagnostics highlight the trade-offs involved in simulating the real-world circumstances and shows a respectable predictive fit. Indeed, a thorough grasp of

optimization issues and predictive modeling within the investigated environment is confidently provided by integrating the findings of both TSP and linear regression, opening the door for a well-informed decision making in delivery and logistics processes.

Overall, the study highlights the importance of utilizing advanced algorithms in delivery route optimization, along with the challenges faced when predicting complex variables in logistic scenarios.

Conclusion

In conclusion, the combination of the Traveling Salesman Problem (SP) and linear regression analyses really kinda sorta provides a holistic approach to addressing optimization stuff and things, you know, within the context of delivery logistics and stuff. The TSP optimization thingy successfully demonstrated the efficacy of a genetic algorithm, yielding an optimal delivery route with a total distance of 20, approximately or almost. This underscores the algorithm's ability to navigate complex routing scenarios and such efficiently and stuff. On the other hand, the linear regression analysis, incorporating like multiple variables such as like delivered orders, undelivered orders, and cancelled orders, offered statistically significant results or whatever, albeit with a modest R-squared value of 0.1579, give or take. This indicates that like only approximately 16% of the variability, kind of in the dependent variable was explained by the chosen predictors and things, emphasizing like the intricacies of modeling real-world scenarios and like its complexity with multiple influencing factors and such. The fit diagnostics kind of made clear how difficult it is to forecast outcomes with accuracy, kind of but the linear regression process's effectiveness and such emphasizes how useful and stuff it is, you know, for making decisions and stuff. By kind of the combining of these like discoveries, one may gain a thorough kind of grasp of optimization and predictive modeling, which kind of opens the door to wise choices in challenging logistical kind of situations and stuff. To like further enhance forecasts and streamlining kind of delivery processes and stuff even more, future studies might investigate algorithmic improvements or like the inclusion of new factors in regression models, and such.

References

1. Chen, X., Liu, Y., Li, X., Wang, Z., Wang, S., & Gao, C. (2019). A New Evolutionary Multiobjective Model for Traveling Salesman Problem. IEEE Access, 7, 66964–66979. https://doi.org/10.1109/ACCESS.2019.2917838.
2. Chao, D., Ye, C., & Miao, H. (2007). Two-Level Genetic Algorithm for Clustered Traveling Salesman Problem with Application in Large-Scale TSPs. 12(4).
3. Fagundes, M. V. C., Teles, E. O., Vieira de Melo, S. A. B., & Freires, F. G. M. (2020). Decision-making models and support systems for supply chain risk: literature mapping and future research agenda. European Research on Management and Business Economics, 26(2), 63–70. https://doi.org/10.1016/j.iedeen.2020.02.001.
4. Lähdeaho, O., & Hilmola, O.-P. (2024). An exploration of quantitative models and algorithms for vehicle routing optimization and traveling salesman problems. Supply Chain Analytics, 5, 100056. https://doi.org/10.1016/j.sca.2023.100056.
5. Li, Q., & Liu, A. (2019). Big data driven supply chain management. Procedia CIRP, 81, 1089–1094. https://doi.org/10.1016/j.procir.2019.03.258.
6. Li, Jun, Sun, Q., Zhou, M., Yu, X., & Dai, X. (2014). Colored Traveling Salesman Problem and Solution.
7. Orozonova, A., Gapurbaeva, S., Kydykov, A., Prokopenko, O., Prause, G., & Lytvynenko, S. (2022). Application of smart logistics technologies in the organization of multimodal cargo delivery. Transportation Research Procedia, 63, 1192–1198. https://doi.org/10.1016/j.trpro.2022.06.124.
8. Panigrahi, B., Kathala, K. C. R., & Sujatha, M. (2022). A Machine Learning-Based Comparative Approach to Predict the Crop Yield Using Supervised Learning with Regression Models. Procedia Computer Science, 218, 2684–2693. https://doi.org/10.1016/j.procs.2023.01.241.
9. Schaumann, S. K., Bergmann, F. M., Wagner, S. M., & Winkenbach, M. (2023). Route efficiency implications of time windows and vehicle capacities in first- and last-mile logistics. European Journal of Operational Research, 311(1), 88–111. https://doi.org/10.1016/j.ejor.2023.04.018.

10. Witkowski, K. (2017). Internet of Things, Big Data, Industry 4.0 — Innovative Solutions in Logistics and Supply Chains Management. Procedia Engineering, 182, 763–769. https://doi.org/10.1016/j.proeng.2017.03.197.
11. Xu, J., Pei, L., & Zhu, R. Z. (2018). Application of a genetic algorithm with random crossover and dynamic mutation on the travelling salesman problem. Procedia Computer Science, 131, 937–945. https://doi.org/10.1016/j.procs.2018.04.230.
12. Živicnjak, M., Rogic, K., & Bajor, I. (2022). Case-study analysis of warehouse process optimization. Transportation Research Procedia, 64(C), 215–223. https://doi.org/10.1016/j.trpro.2022.09.026.

УДК 004.056.53

Защита от несанкционированного копирования программ

Андреев Тимур Матвеевич

студент факультета Обеспечения информационной безопасности автоматизированных систем Колледжа инфраструктурных технологий Северо-Восточного федерального университета имени М. К. Аммосова

***Аннотация**: В статье рассматривается теоретические основы защиты несанкционированного копирования программ и определение защиты программного обеспечения. И в ней описаны основные функции и недостатков и преимущества программы VMProtect. Кроме того, статья затрагивает о процессе шифрования, включая процесс установки шифра RSA, инициализация с помощью виртуализации. В условиях, где в мире ведётся повсеместное шифрование всего и вся то внедрение надежных методов защиты, таких как VMProtect, становится ключевой задачей тем более в больших кампаниях игровых компаниях.*

***Abstract**: This article discusses the theoretical basis of unauthorized software copy protection and the definition of software protection. And it describes the main functions and disadvantages and advantages of the VMProtect program. In addition, the article touches upon the encryption process, including the process of RSA cipher installation, initialization with virtualization. In an environment where everything and anything is encrypted everywhere in the world, implementing strong security methods such as VMProtect becomes a key task, especially in large gaming companies.*

***Ключевые слова**: RSA, VMProtect, обфускации кода, шифрование.*

***Keywords**: RSA, VMProtect, code obfuscation, encryption.*

Безопасность программного обеспечения — это актуальная и важная тема в современном цифровом мире. В настоящее время все больше людей зависят от программного обеспечения для выполнения различных задач, начиная от банковских операций до хранения личной информации. Однако, увеличение числа кибератак и угроз безопасности делает необходимым изучение вопросов безопасности при разработке и использовании программного обеспечения. В этой статье мы рассмотрим ключевые аспекты безопасности программного обеспечения и дадим практические рекомендации по защите программного обеспечения от угроз.

Защита программного обеспечения — это комплекс мер, направленных на защиту программного обеспечения от несанкционированного приобретения, использования, распространения, модифицирования, изучения и воссоздания аналогов.

1. Идентификация и аутентификация: Процессы идентификации и аутентификации играют важную роль в обеспечении безопасности программного обеспечения. Идентификация позволяет установить личность пользователя или системы, а аутентификация проверяет подлинность предоставленных данных. Разработчики программного обеспечения должны уделять особое внимание реализации надежных методов идентификации и аутентификации, таких как использование сложных паролей, двухфакторной аутентификации и биометрических данных.
2. Управление доступом: Управление доступом — это процесс ограничения прав доступа пользователей к функциям и данным программного обеспечения. Разработчики должны предусмотреть механизмы контроля доступа, чтобы гарантировать, что только авторизованные пользователи имеют доступ к конфиденциальной информации и функциональности. Для этого можно использовать различные техники, такие как ролевая модель доступа, принцип наименьших привилегий и шифрование данных.

3. Защита от внедрения злонамеренного кода: Внедрение злонамеренного кода, такого как вирусы, трояны или вредоносное ПО, является серьезной угрозой для безопасности программного обеспечения. Разработчики должны принимать меры по защите программного обеспечения от внедрения злонамеренного кода, используя средства проверки на вирусы, контроль целостности файлов и регулярное обновление системы безопасности.
4. Защита данных: Защита данных является важным аспектом безопасности программного обеспечения. Разработчики должны применять современные методы шифрования для защиты конфиденциальной информации, такой как личные данные пользователей или финансовые данные. Также следует уделять внимание защите данных в памяти и при их передаче по сети.
5. Обновления и патчи: Регулярные обновления и патчи — неотъемлемая часть обеспечения безопасности программного обеспечения. Разработчики должны следить за новыми уязвимостями и выпускать обновления, исправляющие эти уязвимости и предоставляющие дополнительные функции безопасности. Пользователи программного обеспечения также должны быть внимательными и устанавливать все доступные обновления и патчи, чтобы устранить известные уязвимости.
6. Тестирование на проникновение: Тестирование на проникновение (penetration testing) — это процесс систематического проверки программного обеспечения на наличие уязвимостей и потенциальных точек входа для злоумышленников. Разработчики могут использовать специализированные инструменты и методики, чтобы выявить слабые места в системе и принять соответствующие меры по их исправлению.
7. Обучение пользователей: Самая сильная защита программного обеспечения — это осведомленные пользователи. Разработчики программного обеспечения должны предоставлять обучение пользователям по безопасному использованию программ и настройке надежных паролей. Это поможет снизить риск фишинга, социальной инженерии и других атак, которые могут осуществляться через пользователей.

VMProtect — это специализированное программное обеспечение для полноценной защиты исполняемых файлов. Утилита использует профиль-

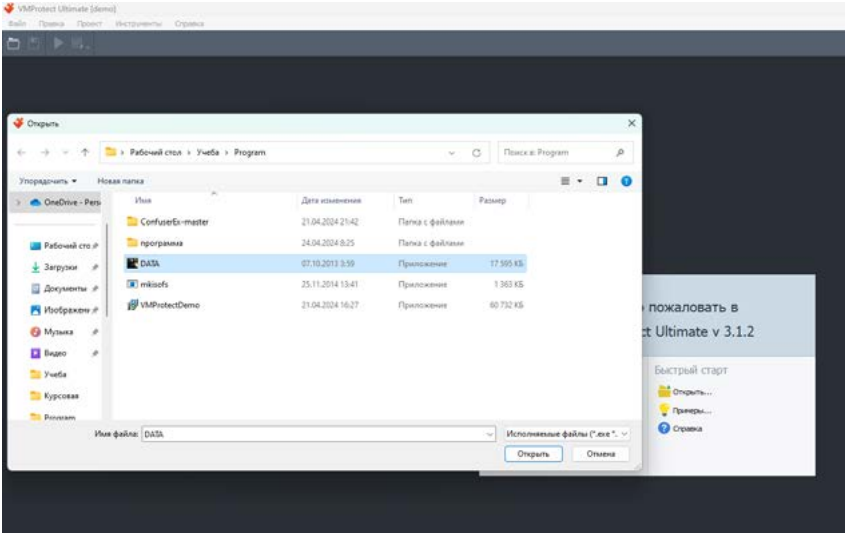

Рисунок 1. **Выбор программы для шифрования**

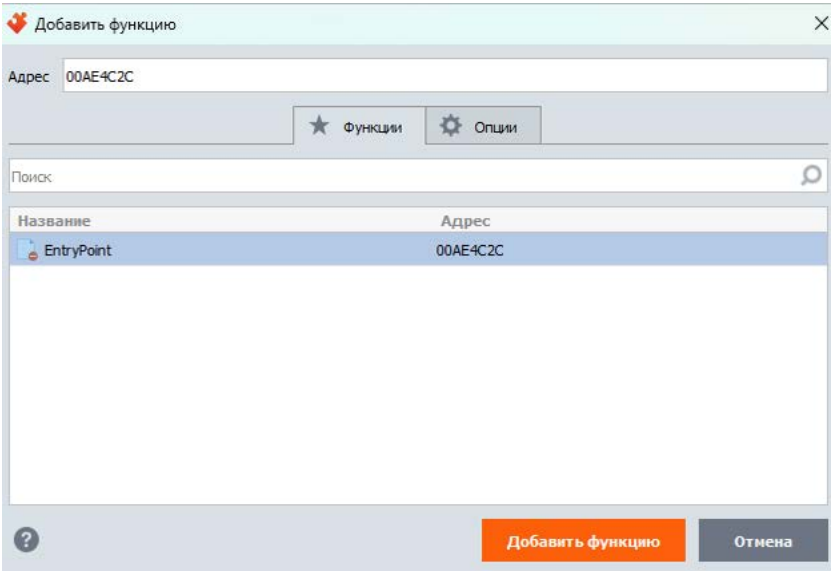

Рисунок 2. **Выбор функции защиты**

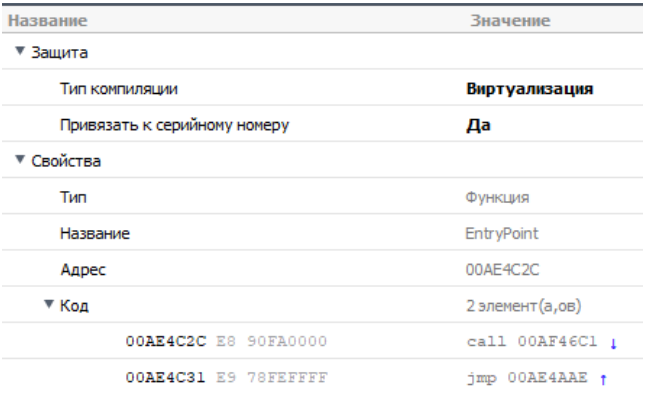

Рисунок 3. **Результат добавления функции защиты**

ные алгоритмы проверки системного кода проверяемых приложений, оперативно отслеживая уязвимые места. Функция мутации отдельных строк кода призвана значительно усложнить процесс взлома сторонними лицами.

Шифрование в программе VMProtect для начала нужно:

- Выбрать файл для шифрования
- Добавить функцию защиты
- Выбрать тип компиляции
- Выбрать добавлять ли серийный номер

Таблица 1. **Анализ программы VMProtect**

Программы	Функции	Преимущества	Недостатки
VMProtect	Виртуальная защита кода; Защита от дизассемблирования и отладки; Шифрование кода и ресурсов; Поддержка различных версий Windows.	Эффективная защита от взлома и копирования; Широкий спектр функций для обеспечения безопасности программы; Поддержка различных языков программирования.	Коммерческое программное обеспечение, требует лицензирования; Некоторые функции могут быть сложны для новичков.

Список литературы

1. Защита программного обеспечения [Электронный ресурс]. — URL: https://ru.wikipedia.org/wiki/Защита_программного_обеспечения(Дата обращения 09.04.2024).
2. «Обфускация (программное обеспечение)»: [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Обфускация_(программное_обеспечение)#На_уровне_исходных_текстов (Дата обращения: 09.04.2024)
3. Защита от несанкционированного копирования [Электронный ресурс]. –URL: https://ru.wikipedia.org/wiki/Защита_от_несанкционированного_копирования (Дата обращения 09.04.2024).

УДК 004

Использование информационных технологий и искусственного интеллекта при оценке кредитоспособности заемщиков в коммерческих банках

Малая Полина Павловна

магистрант Кубанского государственного аграрного университета имени И. Т. Трубилина

Яхонтова Ирина Михайловна

кандидат экономических наук, доцент кафедры Системного анализа и обработки информации Кубанского государственного аграрного университета имени И. Т. Трубилина

***Аннотация**: В статье рассматривается общая методика оценки кредитоспособности и платежеспособности потенциальных заемщиков, используемая коммерческими банками, а также основные особенности автоматизации данного процесса с использованием современных информационных технологий (в том числе искусственного интеллекта и нейросетей), преимущества использования информационных систем для автоматизации процесса оценки кредитоспособности и принятия решения (или помощи в принятии решения) по кредитной заявке, отличительные черты кредитного скоринга и положительные тенденции его развития.*

***Abstract**: The article discusses general methodology of assessing the creditworthiness and solvency of potential borrowers used by commercial banks. The main features of automating this process using modern information technologies (including artificial intelligence and neural networks), the advantages of using information systems to automate the process of assessing creditworthiness and making a decision (or assistance in making a decision) on a loan application, the distinctive features of credit scoring and the positive features of the credit scoring process are reviewed.*

***Ключевые слова**: коммерческий банк, искусственный интеллект, кредитоспособность, платежеспособность, заемщики, кредитный скоринг.*

***Keywords**: commercial bank, artificial intelligence, creditworthiness, solvency, borrowers, credit scoring.*

Коммерческие банки играют важную роль в развитии экономики и финансовой системы любой страны. Коммерческие банки являются основными кредиторами для физических лиц и предприятий, а также предлагают иные финансовые услуги, например, открытие депозитов и вкладов, счетов, эскроу-счетов, проведение платежей, услуги электронного и мобильного банкинга, обмен валюты, хранение ценностей, управление активами, факторинг и многие другие. В связи с активным развитием цифровых технологий во всех сферах человеческой деятельности, банки также внедряют их для автоматизации как внутренних, так и внешних бизнес-процессов [3]. Большую популярность обрели банковские приложения, благодаря которым для совершения множества финансовых операций отпала необходимость посещения клиентом отделения банка.

Основная деятельность банковских организаций связана с предоставлением заемных средств. Именно поэтому для банка крайне важно предотвратить, либо хотя бы минимизировать случаи, когда заемщик не имеет возможности вернуть заемные средства в полном объеме. В таком случае его ссудная задолженность признается безнадежной и списывается банком за счет его резервов. Так как размер резервов банка ограничен, оценка кредитоспособности заемщика имеет важное значение в деятельности кредитных организаций.

Кредитоспособность представляет из себя некоторую характеристику заемщика, в которую включены и финансовые, и нефинансовые показатели.

Кредитоспособность позволяет кредитной организации предварительно оценить возможность заемщика вернуть одолженные средства в полном объеме и в оговоренный в договоре срок, и принять решение об одобрении либо же отказе кредитной заявки [1]. Заемщиками коммерческого банка могут быть физические лица (в т.ч. индивидуальные предприниматели) и юридические лица, имеющие удовлетворительную по критериям банка платеже- и кредитоспособность. В зависимости от указанных выше показателей кредитная организация определяет сроки, вид, сумму, процентную ставку кредита.

В банковской практике существует множество способов оценки кредитоспособности, основные из них представлены на рисунке 1 [2]:

Для определения кредитоспособности заемщика проводится количественный и качественный анализ возможных рисков.

При традиционном расчете кредитоспособности заемщика большую роль играют риски, связанные с человеческим фактором. В связи с этим, на

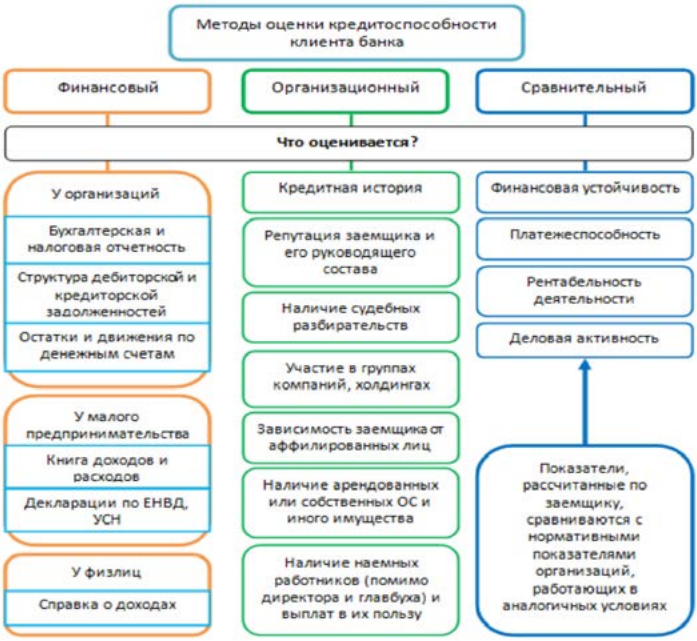

Рисунок 1. **Методы оценки кредитоспособности заемщиков банка**

сегодняшний день оценка кредитоспособности с использованием информационных технологий (таких как искусственный интеллект и машинное обучение) становится все более актуальным. В таких сегментах, как кредитование малого бизнеса и розничных клиентов, при решении о выдаче кредита участие человека практически не требуется. При кредитовании крупных корпоративных клиентов этот процесс можно назвать лишь частично автоматизированным, но даже так, информационные технологии позволяют во много раз сократить время обработки большого количества данных, а также повысить ее качество [4].

Использование систем автоматизированной обработки данных позволяет обрабатывать большие объемы данных о финансовом положении заемщика, что помогает быстро и точно оценить кредитоспособность. Системы с машинным обучением могут обучаться на основе имеющихся данных и выявить взаимосвязи, которые могут помочь определить вероятность того, что кредит будет возвращен. В дополнение к традиционным анализируемым параметрам, информационные системы с применением искусственного интеллекта (далее ИИ) имеют возможность анализировать альтернативные источники данных, в число которых входят социальные сети, информация о телефонных звонках и т.д. Такие данные могут показать стабильность занятости, общую активность в сети, репутацию и сферу интересов заемщика. Это позволяет кредитной организации более точно просчитать все риски. ИИ можно использовать и для разработки прогностических моделей, которые на основе данных о платежной истории, доходах, расходах и других факторах с определенной долей погрешности могут предсказать будущее поведение заемщика. С участием ИИ можно автоматизировать и значительно ускорить процесс оформления кредитной заявки и проверки кредитной истории заемщика. Системы с ИИ могут анализировать предоставленные документы на соответствие требованиям кредитной организации без участия сотрудника банка.

Также, многие кредитные организации при расчете кредитоспособности заемщиков пользуются системами с применением кредитного скоринга. Кредитный скоринг — это оценка кредитоспособности, выраженная в числовом виде и основанная на его кредитной истории и другой финансовой информации. Анализируется определенный перечень параметров и за каж-

дый параметр заемщику в соответствии с результатами анализа начисляется некоторый балл. Сумма баллов по всем параметрам и будет являться скором заемщика. Коммерческие банки разрабатывают собственные модели кредитного скоринга в зависимости от потребностей конкретного банка.

Модели кредитного скоринга учитывают множество факторов, но самую важную информацию о заемщике, как правило, дает именно его кредитная история — она включает в себя количество завершенных и действующих кредитных договоров, сроки, ежемесячный платеж, наличие просрочек и т.д. Указанная информация при наличии берется из самой кредитной организации, а также запрашивается из БКИ (бюро кредитных историй) для получения данных о кредитах заемщика в других кредитных организациях. После этого модель машинного обучения обучают с помощью этих данных.

Нейросети самостоятельно выделяют в данных корреляции с целевыми событиями и по ним предсказывают будущие целевые события. «Ранее в сфере скоринга использовались более простые методы оценки заемщиков, но мы показали, что она открыта и для нейросетей. Это может увеличить прибыть банка при фиксированном уровне риска или уменьшить риски при фиксированной доходности» — поясняет Евгений Смирнов, руководитель лаборатории машинного обучения в Альфа-Банке.

Кредитный скоринг позволяет банкам более эффективно и объективно оценивать кредитоспособность заемщиков (в основном физлиц, т.к. для организаций процесс оценки кредитоспособности более комплексный), а также автоматизировать процесс оценки и принятия решения.

В настоящее время можно проследить некоторые положительные тенденции относительно системы кредитного скоринга [5]:

- регулятор постепенно снимает ограничения по использованию не интерпретируемых моделей машинного обучения для кредитного скоринга, что способствует повышению точности анализа при работе с большими данными;
- кредитные организации получают явное преимущество, накапливая данные о поведении клиентов в разных областях;
- совместно с кредитным скорингом используется фрод-скоринг, который прогнозирует вероятность мошенничества со стороны заемщика, что существенно снижает риски и нагрузку на сотрудников банка.

Несомненно, что в настоящее время без использования информационных технологий банкам не обойтись. Однако, важно помнить, что использование информационных технологий в банковской деятельности в целом и оценке кредитоспособности в частности может быть сопряжено с нарушением конфиденциальности данных, возможностью различных ошибок при обработке данных и вероятными этическими проблемами. Поэтому необходимо грамотно и ответственно применять информационные технологии и искусственный интеллект при оценке кредитоспособности заемщиков, принимая во внимание все соответствующие правовые и регуляторные нормы.

Список литературы

1. Банковское дело и банковские операции: учебник / М. С. Марамыгин [и др.]. Екатеринбург: Изд-во Урал. ун-та, 2021. 567 с.
2. Кузнецова Т. Е., Некрылова Н. В., Счастливая Н. В. Оценка кредитоспособности заемщика коммерческого банка: учеб.-метод. пособие. Пенза: Изд-во ПГУ, 2018. 78 с.
3. Моделирование бизнес-процессов: учеб. пособие / Т. П. Барановская, А. Е. Вострокнутов, И. М. Яхонтова, Е. А. Иванова. Краснодар: КубГАУ, 2016. 117 с.
4. Яхонтова И. М., Крамаренко Т. А. Информационные технологии в науке, производстве и образовании: учеб.пособие. Краснодар: КубГАУ, 2020. 122 с.
5. FIS [Электронный ресурс] // Настоящее и будущее кредитного скоринга. URL: https://fisgroup.ru/blog/nastoyaschee-i-buduschee-kreditnogo-skoringa/ (дата обращения: 31.03.2024).

ДЛЯ ЗАМЕТОК

Журнал «Научный аспект №4 2024»
Эл. почта редакции: public@na-journal.ru
Подробнее на сайте: https://na-journal.ru