УДК 004.056.5

БЕЗОПАСНОЕ ХРАНЕНИЕ ДАННЫХ: СПОСОБЫ И ПОДХОДЫ

Васильев Тимур Игоревич[1],
e-mail: timurvasilev@gmail.com,
[1]*Paybis LTD, Шотландия*

В данной работе рассмотрены ключевые аспекты безопасного хранения данных организации в условиях современного мира. Обсуждены наиболее распространенные угрозы, такие как социальная инженерия, уязвимости конфигурации и инсайдерские атаки. Проведен анализ существующих подходов к безопасному хранению данных в аспекте программного и аппаратного обеспечения и их функционального взаимодействия в системе информационной безопасности. По результатам анализа построен прототип приложения с использованием существующих подходов и минимальным риском утечки данных. Представлены практические рекомендации по выбору ключевых способов безопасного хранения данных. Предложена методика проектирования защищённой среды на основе комплексного рассмотрения технических, организационных и регуляторных аспектов. Новизна полученного результата заключается в формировании нового многослойного подхода к защите данных, включающего единый стандарт взаимодействия сервисов. Практическая значимость полученных результатов заключается в возможности непосредственного применения предложенных методов в рамках промышленных систем, что позволит повысить уровень безопасности и надёжности обработки конфиденциальной информации.

Ключевые слова: безопасное хранение данных, информационная безопасность, социальная инженерия, уязвимости конфигурации, инсайдерские атаки, криптографическая защита, многослойная безопасность, прототипирование систем безопасности, методы предотвращения утечек данных

SECURE DATA STORAGE: METHODS AND APPROACHES

Vasilev T.I.[1],
e-mail: timurvasilev@gmail.com,
[1]*Paybis LTD, Scotland*

The article describes the key aspects of the secure storage of organizational data in the modern world. The most common threats such as social engineering, configuration vulnerabilities, and insider attacks are discussed. The analysis of existing approaches to secure data storage in terms of software and hardware and their functional interaction in the information security system is carried out. Based on the results of the analysis, a prototype application was built using existing approaches and minimal risk of data leakage. Practical recommendations on choosing key ways to securely store data are presented. A methodology for projecting a secure environment based on a comprehensive review of technical, organizational and regulatory aspects is proposed. The novelty of the result obtained lies in the forming of a new multi-layered approach to data protection, including a single standard for the interaction of services. The practical significance of the results obtained lies in the possibility of direct application of the proposed methods within industrial systems, which will increase the level of security and reliability of confidential information processing.

Keywords: secure data storage, information security, social engineering, configuration vulnerabilities, insider attacks, cryptographic protection, multilayer security, prototyping of security systems, methods for preventing data leaks

DOI 10.21777/2500-2112-2025-1-68-76

Введение

В современном мире постоянного роста цифровизации и возрастающих объёмов информации остро встает вопрос безопасного хранения данных. Утечки конфиденциальной информации,

от личной переписки до банковских данных, часто ведут к серьёзным финансовым и репутационным потерям.

В истории есть множество громких случаев утечек данных, благодаря которым становится понятна необходимость надёжных решений. Например, в 2017 году компания Equifax подверглась атаке, в результате которой были украдены персональные данных более 147 миллионов человек. Утечка произошла из-за уязвимости в веб-приложении, которая не была своевременно устранена. В 2019 году из-за утечки данных компании Facebook в открытом доступе оказалось более 540 миллионов записей пользователей. Эти данные хранились без должной защиты на серверах Amazon S3, что подчёркивает важность правильной конфигурации облачных хранилищ. Кибератака на Marriott International в 2018 году привела к компрометации данных о 500 миллионах гостей, включая имена, адреса, телефоны и номера паспортов. Это стало возможным из-за несанкционированного доступа к базе данных, который оставался незамеченным в течение нескольких лет.

Крупные компании и пользователи всё чаще сталкиваются с выбором технологий или инструментов, с помощью которых возможно защитить данные от злоумышленников.

Существует множество исследований, которые посвящены безопасному хранению и обработке данных. В работе [1] подробно описаны методы создания безопасных решений для работы с карточными данными. Автор уделяет внимание вопросам авторизации, безопасному взаимодействию между сервисами в микросервисной архитектуре и надежным методам хранения карточных данных. В качестве недостатка можно отметить, что в работе уделяется значительное внимание вопросам авторизации и безопасной работе с карточными данными, однако практические аспекты масштабирования и адаптации данных решений к другим типам приложений остаются нерассмотренными. В исследовании [2] подробно описаны основные виды угроз для информационной безопасности, а также результаты исследований рынка в сфере защиты данных. С помощью данных факторов авторы предлагают собственный метод безопасного хранения и обработки данных в базах данных. Алгоритм построен на многоуровневом подходе к обеспечению конфиденциальности, целостности и доступности. В статье [3] рассматривается метод рассечения-разнесения для защиты данных в облачных хранилищах, который основан на побитовом разделении файлов и их случайном распределении по разным узлам. В данной работе акцентировано внимание на то, что этот подход обеспечивает более высокий уровень безопасности по сравнению с традиционными методами. Однако метод усложняет управление данными, увеличивает нагрузку на сеть, замедляет процесс восстановления информации и зависит от стабильности работы распределённой инфраструктуры. В работе [4] рассмотрены проблемы безопасного хранения паролей веб-сервисов и предлагается метод защиты с использованием хеширования и криптографической соли. Автор анализирует уязвимости хранения паролей в открытом виде, оценивает различные алгоритмы хеширования и рекомендует SHA-256. Подчёркивается важность динамической соли для защиты от атак по словарю и радужных таблиц. Однако метод не защищает от полного перебора, требует значительных вычислительных ресурсов, а безопасность зависит от надёжного хранения соли.

С учётом изложенного, актуальным является дальнейшее исследование и разработка комплексных методов безопасного хранения и обработки данных, которые учитывали бы современные угрозы, требования рынка и особенности распределённых систем.

Целью данной работы является разработка и обоснование нового подхода к безопасному хранению и обработке данных, позволяющего эффективно противостоять актуальным угрозам и обеспечивать требуемый уровень защищённости информации.

В рамках исследования применялись принципы системного анализа и комплексный подход к оценке результатов, что позволило обоснованно сформулировать требования к предлагаемой методике и рекомендации по её внедрению.

Анализ существующих способов и подходов безопасного хранения данных

Одной из главных угроз для безопасного хранения данных являются методы, которые используются для похищения данных. Самыми распространёнными считаются социальная инженерия, к ко-

торой относятся фишинг или мошеннические звонки, а также внедрение вредоносного программного обеспечения. Также опасны уязвимости или ошибки конфигурации, очень часто данные утекают из-за ошибки настройки серверов и сервисов. Отдельную угрозу представляют инсайдеры, это сотрудники, которые имеют доступ к критическим частям системы и могут действовать в интересах злоумышленников. Подробнее о методах похищения данных можно ознакомиться в работах [5–9]. В совокупности эти факторы требуют комплексного подхода к защите, включающего технические меры, обучение персонала и постоянный аудит инфраструктуры.

Существует множество методов и подходов, которые позволяют сделать безопаснее процесс хранения и работы с данными. Но выбор оптимального подхода во многом зависит от множества условий работы системы. Ниже мы рассмотрим ряд ключевых способов и подходов, которые наиболее часто применяются на практике и заслуживают особого внимания при проектировании и эксплуатации систем хранения данных.

Первым и самым важным шагом при построении системы с безопасным хранением данных является выбор места хранения данных, чаще всего рассматриваются два способа: размещение на собственном физическом сервере или использование облачного хранилища. У каждого из подходов есть свои плюсы и минусы.

Физический сервер предоставляет полный контроль над оборудованием и инфраструктурой, в таком подходе администраторы могут гибко настраивать серверы, сетевую архитектуру и физический доступ к ним. При использовании данного подхода повышается безопасность и конфиденциальность, так как хранение данных осуществляется в пределах компании, что существенно уменьшает риск утечки информации. Также локальное размещение данных часто является единственным способом соблюдения внутренних стандартов и регуляций, особенно если отрасль или законодательство требуют, чтобы информация хранилась в конкретной юрисдикции или под полным контролем организации. Но у данного решения есть и ряд минусов: высокая совокупная стоимость обслуживания и владения, так как приходится закупать и обслуживать серверы, а также оплачивать электроэнергию. Для масштабирования в случае роста нагрузки на систему требует дополнительных финансовых вложений. Наконец, нужны дополнительные сотрудники для работы с физическими серверами, так как такая работа требует специфической экспертизы.

Облачные хранилища выгодны своей масштабируемостью, при необходимости можно быстро увеличивать или уменьшать объём доступного хранилища и вычислительных ресурсов. Для поддержания проекта достаточно небольшой команды инженеров, что является существенным плюсом. Но хранение данных в облаке часто вызывает вопросы конфиденциальности и контроля, особенно если речь идёт о критически важной информации. К тому же корпоративные или законодательные нормы могут запрещать хранение данных за пределами определённой страны либо предъявлять дополнительные требования к шифрованию и управлению ключами.

Следующим важным шагом является внедрение чёткого разграничения прав доступа. Для этого обычно используются такие механизмы, как ролевое управление доступом (RBAC) или атрибутивная модель (ABAC). В основе этих подходов лежит принцип минимизации привилегий, благодаря чему сотрудники получают доступ только к тем блокам системы, которые необходимы для выполнения прямых обязанностей. Например, полный доступ к данным следует предоставлять только ограниченному кругу доверенных лиц, обычно это администраторы или “DevOps”-инженеры. Для сотрудников, которые работают строго в определенном доменном контексте, например, команде, отвечающей за платежи, права должны быть ограничены до чтения исключительно их доменной зоны. То есть только чтение платежных данных без модификации.

Использование микросервисной архитектуры может значительно повысить уровень безопасности при хранении данных. Благодаря разделению приложения на независимые сервисы каждый компонент отвечает лишь за ограниченный набор функций, что позволяет четко разграничивать доступ к данным. Если злоумышленнику удастся взломать один из микросервисов, то он не получит доступ ко всем данным приложения, так как остальные сервисы имеют собственную базу данных. Также данная архитектура позволяет гибко внедрять и обновлять механизмы безопасности. Помимо этого, микросервисы упрощают реализацию подхода Zero Trust, при котором каждый запрос между сервисами проверяется, а каналы связи шифруются. При этом не требуется изменять всю систему при изменении или ужесточе-

нии политик безопасности – достаточно настроить отдельные сервисы или коммуникационные шины. Такой подход даёт возможность легко масштабироваться, эффективно анализировать трафик и вводить дополнительные меры защиты при росте нагрузки или появлении новых микросервисов. Подробнее о способах внедрения микросервисов в приложении можно ознакомиться в работах [10–15].

Резервное копирование данных и наличие продуманного плана восстановления является решающим фактором в критических ситуациях, когда данные могут быть повреждены или потеряны. Важно создавать регулярные бэкапы для сохранения нескольких версий данных, таким образом создавая «страховочную сетку». Также копии важно хранить не только физически раздельно (разные дата-центры), но и в логически разных средах (облако и локальный сервер). Такой географический и технический «разнос» снижает риск единовременного выхода из строя всех копий в случае катастрофы или массовой атаки. Помимо резервного копирования, необходимо убедиться, что данные в копиях пригодны для восстановления, нужно настроить автоматизированное тестирование бэкапов. Это может включать как периодическую проверку целостности, имитацию инцидентов в тестовой среде, так и осуществление восстановления, для оценки времени, необходимого для возвращения системы в рабочее состояние. Такой подход поможет выявить и устранить проблемные места в планах восстановления. В результате компания получит уверенность в надёжности резервов и сократит потенциальный простой при сбоях.

Главным аспектом безопасного хранения данных является использование механизмов анонимизации и псевдонимизации данных. При анонимизации все персональные данные будут преобразованы таким образом, что идентифицировать конкретного пользователя станет невозможно. Например, вместо имени и фамилии хранится только статистическая информация, которая не позволяет установить связь с конкретным пользователем. В случае псевдонимизации данные также изменяются, но связь с пользователем доступна через уникальный идентификатор. Такого рода способ обеспечения безопасности данных полезен при обработке, когда необходимо сохранять их функциональность (например, в аналитике), но при этом необходимо обеспечить высокий уровень конфиденциальности.

Токенизация является одним из методов псевдонимизации. С использованием токенизации данные защищаются с помощью замены конфиденциальной информации на уникальные токены, в которых не содержится исходной информации, а также данный токен не может быть обратно преобразован без использования секретного ключа. В случае платежных систем это значит, что информация о банковских картах или банковских счетах будет заменена токенами, которые в дальнейшем могут передаваться между компаниями без риска кражи или утечки конфиденциальных данных пользователя. Одним из главных преимуществ токенизации является то, что в случае кражи токенов злоумышленниками они не смогут воспользоваться ими для проведения платежных операций, так как токены сами по себе не имеют значительной ценности. Благодаря этому токенизация и стала важным инструментом для защиты данных, особенно это актуально для высоконагруженных систем, где большой объем транзакций и риск утечки данных высок.

Внешним системам передаётся уникальный токен, который создаётся специально для конкретной интеграции или одной попытки оплаты. Этот токен привязан исключительно к заданному контексту и времени использования, что делает его абсолютно бесполезным для злоумышленников в случае утечки. Более того, даже при успешной компрометации токена его использование невозможно без соблюдения ряда дополнительных условий, таких как проверка источника запроса, временные ограничения и валидация дополнительных параметров, привязанных к токену. Такой подход значительно снижает риски несанкционированного доступа и повышает общий уровень безопасности системы.

Шифрование является одним из ключевых инструментов для обеспечения конфиденциальности данных. Оно превращает исходную информацию в зашифрованный вид, который невозможно прочитать без использования специального ключа. Современные методы делятся на две основные категории: симметричное и асимметричное.

Симметричное шифрование основано на использовании одного и того же ключа для шифрования и расшифровки данных. Примерами таких алгоритмов являются AES (Advanced Encryption Standard) и Triple DES. Симметричное шифрование отличается высокой скоростью работы, что делает его подходящим для задач, требующих шифрования больших объёмов данных, например, при шифровании жёстких дисков или сетевого трафика. Основным вызовом здесь является безопасное управление ключами, так как компрометация ключа может привести к утечке всех зашифрованных данных.

Асимметричное шифрование использует пару ключей: открытый и закрытый. Открытый ключ используется для шифрования данных, а закрытый – для их расшифровки. Примерами асимметричных алгоритмов являются RSA и ECC (Elliptic Curve Cryptography). Этот метод широко применяется для защиты данных при передаче (например, в HTTPS), а также для цифровых подписей. Асимметричное шифрование обеспечивает высокий уровень безопасности, но его производительность ниже по сравнению с симметричными методами, поэтому на практике их часто комбинируют.

Дополнительно применяются хэш-функции, такие как SHA-256, которые преобразуют данные в уникальный фиксированный отпечаток. Хотя хэширование не является шифрованием, оно широко используется для проверки целостности данных и аутентификации. Для повышения безопасности применяется солирование (добавление случайных данных перед хэшированием), чтобы предотвратить атаки на основе предварительно вычисленных значений (таблицы радужных хэшей).

Важным аспектом шифрования является управление ключами. Специальные системы, такие как HSM (Hardware Security Module) и программные решения (например, AWS KMS или HashiCorp Vault), обеспечивают безопасное хранение, ротацию и управление ключами. Такие инструменты помогают предотвратить их утечку и автоматизировать управление шифрованием в масштабах компании.

Реализация многоуровневых методов хранения данных в реальной среде

Рассмотрим практический пример проектирования приложения, которое соответствует всем принципам безопасности и опирается на вышеприведенные правила. Для примера можно разработать приложение, которое создается для крупной медицинской организации, которая оказывает широкий спектр услуг пациентам и взаимодействует с различными партнёрами. Допустим, система, предназначенная для записи на приём, управления персоналом, обработки результатов анализов и ведения электронных медицинских карт. Ввиду высокой конфиденциальности данных и строгих требований регуляторов (например, законодательство о персональных данных пациентов), организация стремится обеспечить максимальную безопасность при хранении и обработке полученной информации. Первая задача, которую необходимо решить, – выбрать подходящую инфраструктуру для размещения серверов и баз данных. В зависимости от специфики компании и её финансовых возможностей специалисты рассматривают два варианта: использование собственных серверов, которые расположены в отдельном дата-центре компании, и обращение к облачному решению. Технический комитет после изучения требований по хранению персональных данных в конкретной юрисдикции принимает решение о гибридном подходе: наиболее чувствительная информация (полные медицинские карты, паспортные данные, финансовые документы) будут храниться на локальных серверах в зашифрованном виде, а менее критичная часть, связанная, к примеру, с записью на консультации, push-уведомлениями и частью аналитических отчётов, будет развернута в облачной среде. Это позволит масштабировать определённые сервисы, не подвергая риску все записи пациентов, а также сохранять внутренние стандарты и регламенты безопасности.

Следующим этапом становится проработка архитектуры приложения. Приложение будет разделено на несколько микросервисов, каждый из которых отвечает за собственный участок функционала: сервис управления личными кабинетами, сервис медицинских карт, сервис выгрузки результатов анализов, сервис записи и оплаты услуг, сервис аналитики и статистики, сервис уведомлений. Такое разделение позволяет ввести принцип минимизации привилегий: у каждого микросервиса есть отдельная база данных, а общий доступ между ними жёстко контролируется на уровне сетевых политик и проверки подлинности запросов. Каждый сервис не доверяет остальным по умолчанию, а любая попытка взаимодействия требует аутентификации и, при необходимости, авторизации. Для этого в системе предусмотрен сервис авторизации, который выдаёт каждому участнику взаимодействия уникальные токены. При обращении к сервису медицинских карт, например, сервис авторизации проверяет роль пользователя, необходимое действие и выдает JWT-токен, ограниченный по времени и связанному контексту. Все запросы между микросервисами проходят через зашифрованные каналы.

Хранение данных организовано таким образом, чтобы каждое хранилище было зашифровано, и доступ к нему осуществлялся только при наличии соответствующих прав. Например, базу данных

с медицинскими картами обслуживает PostgreSQL, настроенный на работу с аппаратными модулями безопасности, где генерируются и хранятся ключи. Если по тем или иным причинам компрометация физического носителя данных станет реальностью, злоумышленники не смогут расшифровать зашифрованные файлы без соответствующего ключа. В облачной части приложения хранятся второстепенные данные (очередь на приём, перечень кабинетов, расписание врачей), при этом для обеспечения безопасности используется аналогичная схема шифрования, только с применением решения KMS (Key Management Service) от облачного провайдера. Все пароли и секреты, необходимые для внутреннего взаимодействия сервисов, хранятся в специальных хранилищах (например, HashiCorp Vault или AWS Secrets Manager) с учётом принципа ротации ключей и уведомлений при подозрительных операциях.

Для персональных данных, таких как номера полисов ОМС, страховые документы, реквизиты банковских карт при оплате услуг, активно применяется псевдонимизация и токенизация. Допустим, при оплате консультации пациент вводит реквизиты карты: система создания токена заменяет реальные данные на уникальный идентификатор, который хранится во внутренней защищённой базе. В дальнейшем внешний сервис видит лишь сгенерированный токен, который не раскрывает никаких реальных сведений о клиенте и используется только в строго заданном контексте: при повторном выставлении счёта или возврате средств. Если недобросовестный сотрудник или злоумышленник получит такой токен, он не сможет напрямую провести операцию, поскольку потребуется верификация этого токена дополнительными параметрами, включая принадлежность к конкретному аккаунту, ограничение по времени и источнику запроса. Подобная схема не только снижает риски финансовых утечек, но и упрощает соблюдение требований стандартов вроде PCI DSS, поскольку реальные банковские данные не фигурируют в открытом виде ни в одном из микросервисов, кроме того, который непосредственно отвечает за связь с банком или платёжной системой.

Организация резервного копирования строится на многоуровневой модели, которая включает регулярные бэкапы. Внутри локального дата-центра приложения формируются полные и инкрементные архивы, которые ежедневно сохраняются на выделенных серверах, расположенных в другом помещении или даже географически удалённом центре обработки данных. Параллельно шифрованные копии выгружаются в облако, где они хранятся в отдельной среде и не сопряжены непосредственно с продуктивными серверами. Такая избыточность помогает справиться со сбоями, будь то выход из строя локальной инфраструктуры, физическое повреждение серверов или случайное удаление данных администраторами. При этом тщательно организована система проверок, которая регулярно запускает процедуру восстановления резервных копий в тестовом окружении, проверяя целостность и корректность. Если на каком-то этапе будут выявлены повреждения или несостыковки в данных, ответственные сотрудники компании немедленно получат уведомление и могут в сжатые сроки устранить проблему.

Немаловажным аспектом для подобной платформы выступает обучение персонала и аудит инфраструктуры. Все сотрудники компании, которые работают с системой, должны проходить курсы по информационной безопасности. Администраторы и разработчики должны регулярно проверять журналы доступа к критическим базам, анализировать информацию в системах мониторинга. Например, если вдруг будет выявлена серия запросов, которые приходят на сервис медицинских карт из непредвиденных сегментов сети, система безопасности способна автоматически заблокировать такой трафик и уведомить ответственных специалистов. Параллельно должны проводиться регулярные тесты и сканирование уязвимостей, на основании которых формируются отчёты и планы по повышению надёжности приложения.

Рассмотрим преимущества такой схемы работы приложения. Во-первых, за счёт разделения приложения на микросервисы каждый компонент не хранит больше данных, чем ему действительно необходимо. Во-вторых, применение токенизации при работе с платёжными или персональными данными ограничивает область, где потенциальная утечка может нанести урон. В-третьих, симметричное и асимметричное шифрование данных и каналов передачи делает перехват информации бессмысленным, поскольку злоумышленник не сможет расшифровать похищенные записи. В-четвёртых, постоянный аудит логов и резервное копирование облегчат оперативное восстановление и реакцию на возможные инциденты. Наконец, гибридная архитектура даёт возможность гибко распределять нагрузку: наиболее затратные по ресурсам процессы выполняются в облаке, а конфиденциальные сведения остаются под полным контролем организации.

Таким образом, на примере приложения можно продемонстрировать целый комплекс мер и методов, о которых шла речь в обзорной части данной статьи: от правильного выбора места размещения данных и принципов разграничения прав до токенизации, резервного копирования и шифрования на разных уровнях. Каждая из этих мер помогает добиться максимальной защищённости и устойчивости к многочисленным современным угрозам, в том числе связанным с социальной инженерией, инсайдерскими атаками и уязвимостями в программном обеспечении. Дополнительной гарантией надёжности становится постоянная работа над системой безопасности, включая обновления микросервисов, инфраструктурных компонентов и регулярные тренинги для сотрудников. Этот комплексный подход к построению и эксплуатации приложения позволяет соответствовать как внутренним стандартам компании, так и внешним требованиям регулирующих органов, при этом сохраняя высокую доступность и удобство эксплуатации для всех участников процесса.

Заключение

В ходе исследования были рассмотрены ключевые способы и подходы к безопасному хранению и обработке данных, включая вопросы инфраструктуры, разграничения прав доступа, резервного копирования, шифрования, токенизации и псевдонимизации. Проведённый анализ подтвердил актуальность темы, обусловленную ростом числа кибератак и непрерывным усложнением систем хранения информации. Поставленная в работе цель – разработка и обоснование нового подхода, позволяющего эффективно противостоять современным угрозам и обеспечивать требуемый уровень защищённости, – была достигнута за счёт комплексного рассмотрения технических, организационных и регуляторных аспектов.

Новизна полученных результатов заключается в комплексной методике проектирования защищённой среды, которая учитывает реалии распределённых систем и микросервисной архитектуры. Важным элементом является гибкое применение способов шифрования, псевдонимизации и разграничения прав, что позволяет избежать избыточного доступа и минимизировать воздействие потенциальных атак. Кроме того, в работе подчёркивается значение регулярного аудита, мониторинга, обучения персонала и корректной конфигурации инфраструктуры в целом.

Практическая значимость представленных решений заключается в том, что описанные методы могут быть внедрены в различные сферы – от финансового сектора и телекоммуникаций до здравоохранения и электронного документооборота. Сочетание микросервисной архитектуры с многоуровневым шифрованием и токенизацией подходит для высоконагруженных систем, где необходимы гибкость, масштабируемость и соответствие строгим требованиям регуляторов по защите персональных данных. При этом уделяется внимание как внешним угрозам (вредоносному ПО, взлому, утечкам из облачных хранилищ), так и инсайдерским факторам, наиболее критичным в крупных организациях.

Результаты исследования свидетельствуют об эффективности сочетания данных технологий: при внедрении разграничения прав доступа по принципу наименьших привилегий, регулярном резервировании и проверке целостности бэкапов, использовании шифрования на различных уровнях и применении токенизации, за счет чего уменьшается вероятность несанкционированных действий и последствий возможных инцидентов. Разработанная методика может стать основой для дальнейших исследований и совершенствования практик безопасного хранения данных. Это особенно актуально в условиях, когда технологиям приходится постоянно адаптироваться к появлению новых видов угроз и к изменениям в законодательстве, регулирующем обращение с конфиденциальной информацией.

Таким образом, можно сделать вывод, что представленное решение и изложенные в работе рекомендации успешно подтверждают поставленную цель и позволяют повысить безопасность информационных систем, сохранив при этом их производительность и гибкость.

Список литературы

1. *Васильев Т.И.* Разработка безопасных решений для обработки платежных данных // Восточно-Европейский научный журнал. – 2024. – № 8 (105). – С. 8–13.

2. *Емельянов А.В., Емельянова С.С.* Анализ угроз информационной безопасности и разработка алгоритма безопасного хранения данных в базе данных // Научно-технический вестник Поволжья. – 2019. – № 6. – С. 135–137.

3. *Аршинский Л.В., Шурховецкий Г.Н.* Особенности применения метода рассечения-разнесения для безопасного хранения данных во внешних хранилищах // Информационные технологии. – 2021. – Т. 27, № 5. – С. 259–266.
4. *Фатьянова А.А., Золотухин Д.А.* Хеширование пользовательских данных для безопасного хранения // Математические исследования и информационные технологии в цифровой экономике: сборник научных трудов по итогам Межвузовской научно-практической конференции, Саратов, 25 февраля 2020 года. – Саратов, 2020. – С. 55–58.
5. *Копылова О.П., Заусалин П.А.* Способы и методы кражи персональных данных // Уголовно-процессуальное и технико-криминалистическое обеспечение для борьбы с киберпреступлениями в современном обществе: материалы II Всероссийской научно-практической конференции, Тамбов, 07 декабря 2023 года. – Тамбов, 2023. – С. 111–118.
6. *Маныч Е.Г.* Кража финансовых данных или данных банковских карт – один из видов киберпреступления // Цифровые технологии в борьбе с преступностью: проблемы, состояние, тенденции: сборник материалов I Всероссийской научно-практической конференции, Москва, 27 января 2021 года. – Москва, 2021. – С. 156–158.
7. *Закиев Е.Г., Дашкевич А.А.* Технологии защиты данных от краж // Цифровая экономика глазами студентов: материалы Всероссийской научной конференции, Казань, 2024 год. – Казань, 2024. – С. 385–388.
8. *Филенко Е.С., Соболев П.Ю.* Кража данных с помощью CSS: атака и защита // Интеллектуальный потенциал XXI века: материалы Международной (заочной) научно-практической конференции, София, 17 мая 2018 года / под общ. ред. А.И. Вострецова. – Нефтекамск: Мир науки, 2018. – С. 49–55.
9. *Минченко В.С.* Современные методы кражи персональных данных и способы противодействия им // Наука и инновации в XXI веке: актуальные вопросы, открытия и достижения: сборник статей XVIII Международной научно-практической конференции, Пенза, 05 апреля 2020 года. – Пенза: Наука и Просвещение, 2020. – С. 20–23.
10. *Караханова А.А.* Анализ микросервисной архитектуры, монолитных приложений, архитектуры SOA // Синергия наук. – 2020. – № 46. – С. 255–262.
11. *Луценко Д.Ю., Полякова Л.П.* Разбиение монолитного приложения на микросервисы с использованием паттерна Strangler // Информационные технологии в управлении и экономике. – 2021. – № 3 (24). – С. 82–87.
12. *Lokhvitsky V.A., Goncharenko V.A., Levchik E.S.* A scalable microservice model based on a queuing system with “cooling” // Intellectual technologies on transport. – 2022. – No. 4 (32). – P. 46–51.
13. *Никитин И.В., Гриценко Т.Ю.* Сравнение подходов монолитной архитектуры и микросервисной архитектуры при реализации серверной части веб-приложения // Дневник науки. – 2020. – № 3. – URL: http://dnevniknauki.ru/images/publications/2020/3/technics/Nikitin_Gritsenko.pdf (дата обращения: 30.01.2025). – Текст: электронный.
14. *Козин А.А., Харченко А.В.* Архитектура микросервисов для небольших команд // Прикладная математика: современные проблемы математики, информатики и моделирования: материалы VI Всероссийской научно-практической конференции молодых ученых, Краснодар, 15–21 апреля 2024 года. – Краснодар, 2024. – С. 402–405.
15. *Корниенко Д.В., Никулин А.В.* Архитектурные паттерны проектирования микросервисов в JAVA // Инновационные технологии современной научной деятельности: стратегия, задачи, внедрение: сборник статей по итогам Международной научно-практической конференции, Иркутск, 24 апреля 2024 года. – Стерлитамак, 2024. – С. 150–160.

References

1. *Vasil’ev T.I.* Razrabotka bezopasnyh reshenij dlya obrabotki platezhnyh dannyh // Vostochno-evropejskij nauchnyj zhurnal. – 2024. – № 8 (105). – S. 8–13.
2. *Emel’yanov A.V., Emel’yanova S.S.* Analiz ugroz informacionnoj bezopasnosti i razrabotka algoritma bezopasnogo hraneniya dannyh v baze dannyh // Nauchno-tekhnicheskij vestnik Povolzh’ya. – 2019. – № 6. – S. 135–137.

3. *Arshinskij L.V., Shurhoveckij G.N.* Osobennosti primeneniya metoda rassecheniya-razneseniya dlya bezopasnogo hraneniya dannyh vo vneshnih hranilishchah // Informacionnye tekhnologii. – 2021. – T. 27, № 5. – S. 259–266.
4. *Fat'yanova A.A., Zolotuhin D.A.* Heshirovanie pol'zovatel'skih dannyh dlya bezopasnogo hraneniya // Matematicheskie issledovaniya i informacionnye tekhnologii v cifrovoj ekonomike: sbornik nauchnyh trudov po itogam Mezhvuzovskoj nauchno-prakticheskoj konferencii, Saratov, 25 fevralya 2020 goda. – Saratov, 2020. – S. 55–58.
5. *Kopylova O.P., Zausalin P.A.* Sposoby i metody krazhi personal'nyh dannyh // Ugolovno-processual'noe i tekhniko-kriminalisticheskoe obespechenie dlya bor'by s kiberprestupleniyami v sovremennom obshchestve: materialy II Vserossijskoj nauchno-prakticheskoj konferencii, Tambov, 07 dekabrya 2023 goda. – Tambov, 2023. – S. 111–118.
6. *Manych E.G.* Krazha finansovyh dannyh ili dannyh bankovskih kart – odin iz vidov kiberprestupleniya // Cifrovye tekhnologii v bor'be s prestupnost'yu: problemy, sostoyanie, tendencii: sbornik materialov I Vserossijskoj nauchno-prakticheskoj konferencii, Moskva, 27 yanvarya 2021 goda. – Moskva, 2021. – S. 156–158.
7. *Zakiev E.G., Dashkevich A.A.* Tekhnologii zashchity dannyh ot krazh // Cifrovaya ekonomika glazami studentov: materialy Vserossijskoj nauchnoj konferencii, Kazan', 2024 god. – Kazan', 2024. – S. 385–388.
8. *Filenko E.S., Sobolev P.Yu.* Krazha dannyh s pomoshch'yu CSS: ataka i zashchita // Intellektual'nyj potencial XXI veka: materialy Mezhdunarodnoj (zaochnoj) nauchno-prakticheskoj konferencii, Sofiya, 17 maya 2018 goda / pod obshch. red. A.I. Vostrecova. – Neftekamsk: Mir nauki, 2018. – S. 49–55.
9. *Minchenko V.S.* Sovremennye metody krazhi personal'nyh dannyh i sposoby protivodejstviya im // Nauka i innovacii v XXI veke: aktual'nye voprosy, otkrytiya i dostizheniya: sbornik statej XVIII Mezhdunarodnoj nauchno-prakticheskoj konferencii, Penza, 05 aprelya 2020 goda. – Penza: Nauka i Prosveshchenie, 2020. – S. 20–23.
10. *Karahanova A.A.* Analiz mikroservisnoj arhitektury, monolitnyh prilozhenij, arhitektury SOA // Sinergiya nauk. – 2020. – № 46. – S. 255–262.
11. *Lucenko D.Yu., Polyakova L.P.* Razbienie monolitnogo prilozheniya na mikroservisy s ispol'zovaniem patterna Strangler // Informacionnye tekhnologii v upravlenii i ekonomike. – 2021. – № 3 (24). – S. 82–87.
12. *Lokhvitsky V.A., Goncharenko V.A., Levchik E.S.* A scalable microservice model based on a queuing system with "cooling" // Intellectual technologies on transport. – 2022. – No. 4 (32). – P. 46–51.
13. *Nikitin I.V., Gricenko T.Yu*. Sravnenie podhodov monolitnoj arhitektury i mikroservisnoj arhitektury pri realizacii servernoj chasti veb-prilozheniya // Dnevnik nauki. – 2020. – № 3. – URL: http://dnevniknauki.ru/images/publications/2020/3/technics/Nikitin_Gritsenko.pdf (data obrashcheniya: 30.01.2025). – Tekst: elektronnyj.
14. *Kozin A.A., Harchenko A.V.* Arhitektura mikroservisov dlya nebol'shih komand // Prikladnaya matematika: sovremennye problemy matematiki, informatiki i modelirovaniya: materialy VI Vserossijskoj nauchno-prakticheskoj konferencii molodyh uchenyh, Krasnodar, 15–21 aprelya 2024 goda. – Krasnodar, 2024. – S. 402–405.
15. *Kornienko D.V., Nikulin A.V.* Arhitekturnye patterny proektirovaniya mikroservisov v JAVA // Innovacionnye tekhnologii sovremennoj nauchnoj deyatel'nosti: strategiya, zadachi, vnedrenie: sbornik statej po itogam Mezhdunarodnoj nauchno-prakticheskoj konferencii, Irkutsk, 24 aprelya 2024 goda. – Sterlitamak, 2024. – S. 150–160.

Статья поступила в редакцию: 23.01.2025
Статья принята в печать: 10.03.2025
Received: 23.01.2025
Accepted: 10.03.2025