



10.5281/zenodo.14029864

ГЛУМОВ Константин Сергеевич

ведущий инженер-программист, Альфа-Банк, Россия, г. Пермь

БЕЗОПАСНОСТЬ МИКРОСЕРВИСОВ: УПРАВЛЕНИЕ СЕКРЕТАМИ И БЕЗОПАСНАЯ АУТЕНТИФИКАЦИЯ

Аннотация. Безопасность микросервисов играет ключевую роль в современных распределенных системах, где каждый компонент требует надежной защиты для предотвращения утечек данных и несанкционированного доступа. Основными аспектами безопасности микросервисов являются управление секретами и безопасная аутентификация, позволяющие контролировать взаимодействие между сервисами и доступ к ресурсам. Управление секретами включает в себя методы хранения и ротации чувствительной информации, такие как ключи и токены, что снижает риски атак и утечек данных. Безопасная аутентификация осуществляется через механизмы, такие как JSON Web Tokens (JWT), OAuth 2.0 и Mutual TLS (mTLS), которые обеспечивают защиту от атак на межсервисные взаимодействия. Регулярное обновление методов безопасности, а также внедрение комплексных решений, таких как системы управления доступом на основе ролей (RBAC) и атрибутов (ABAC), являются неотъемлемыми мерами для повышения уровня защиты.

Ключевые слова: микросервисы, управление секретами, аутентификация, JWT, OAuth 2.0, mTLS, безопасность данных.

Введение

Микросервисная архитектура стала важным шагом в развитии современных распределенных систем, обеспечивая гибкость и масштабируемость программного обеспечения. В отличие от монолитных приложений, микросервисы позволяют разрабатывать и разворачивать независимые компоненты, что повышает скорость разработки и упрощает процесс обслуживания. Однако с увеличением количества микросервисов и их взаимодействий значительно возрастают риски, связанные с безопасностью. Каждый микросервис может стать точкой уязвимости, через которую злоумышленники могут получить доступ к критически важной информации или ресурсам системы.

Одной из основных проблем при использовании микросервисов является управление секретами и обеспечение безопасной аутентификации. Микросервисы взаимодействуют друг с другом и с внешними системами, что требует надежного механизма идентификации и защиты передаваемых данных. Без правильного управления секретами, такими как ключи доступа и пароли, системы становятся уязвимыми к атакам. Безопасная аутентификация между сервисами позволяет гарантировать, что каждый запрос исходит от авторизованного

источника, тем самым предотвращая несанкционированный доступ.

Актуальность исследования обусловлена растущими требованиями к безопасности в условиях постоянного увеличения числа микросервисов и их интеграции в критически важные системы. В современных условиях, когда кибератаки становятся всё более изощрёнными, применение эффективных методов управления секретами и аутентификации играет ключевую роль в обеспечении безопасности данных и инфраструктуры.

Цель данной работы – исследовать ключевые аспекты управления секретами и методов безопасной аутентификации в микросервисных архитектурах, а также предложить практические рекомендации по их применению для повышения уровня защиты распределённых систем.

1. Управление секретами в микросервисах

Микросервисные архитектуры представляют собой современный подход к построению сложных приложений, в которых каждое приложение состоит из небольших, независимых сервисов. Важнейшим аспектом их взаимодействия является надежная и безопасная связь между сервисами. Каждый микросервис

выполняет одну или несколько конкретных задач, а для достижения общей цели они обмениваются данными между собой. Для обеспечения безопасности таких взаимодействий необходимо применять механизмы аутентификации и авторизации, которые исключают возможность несанкционированного доступа к данным.

Ключевым этапом этого взаимодействия является установка защищенного канала связи, в рамках которого происходит проверка подлинности участвующих сторон. В контексте микросервисов это обычно реализуется с помощью процедуры service-to-service аутентификации. Этот процесс заключается в проверке того, что каждый сервис в системе действительно тот, за кого себя выдает, что позволяет защитить данные от возможных атак и утечек. Важно отметить, что такой подход требует детального проектирования, так как необходимо учитывать многочисленные факторы, такие как использование различных языков программирования и платформ для разработки микросервисов.

Архитекторы систем сталкиваются с необходимостью выбора безопасного и устойчивого к атакам механизма аутентификации. Одной из самых популярных технологий является mTLS (Mutual TLS), которая обеспечивает взаимную проверку подлинности как на стороне сервера, так и на стороне клиента. Этот механизм предоставляет надежный уровень защиты, что особенно важно для систем, обрабатывающих конфиденциальные данные. Однако настройка такой инфраструктуры может оказаться сложной и потребовать внедрения

сложных процедур управления ключами и сертификатами [1, с. 146-154].

Альтернативным подходом является использование JSON Web Tokens (JWT), который позволяет осуществлять аутентификацию на прикладном уровне. JWT предоставляет возможность передачи утверждений о подлинности между сервисами в формате токенов, что упрощает процесс аутентификации и делает его более гибким. В то время как mTLS работает на уровне транспортного протокола, JWT функционирует на более высоком уровне, что делает его более подходящим для определенных сценариев использования.

Для построения безопасной системы взаимодействия микросервисов также можно использовать подходы на основе доверенных сетей, однако этот вариант требует высокого уровня доверия ко всем компонентам системы. Системы с нулевым доверием, напротив, предполагают, что каждый запрос и каждое взаимодействие должны быть проверены на подлинность и авторизованы перед обработкой. Такой подход значительно повышает безопасность, но усложняет процесс проектирования.

Эффективное управление секретами является одной из ключевых задач кибербезопасности, обеспечивающей контроль над доступом к ресурсам на основании строгих политик. Это касается данных, которые не принадлежат пользователям, и направлено на защиту информации с помощью процессов аутентификации и авторизации [2, с. 267-281]. Далее в таблице 1 будут рассмотрены существующие проблемы, связанные с аутентификацией микросервисов.

Таблица 1

Основные сложности аутентификации в микросервисной архитектуре [3, с. 25-36]

Сложность	Описание
Централизованная зависимость	Каждый микросервис должен иметь свою собственную логику аутентификации и авторизации, что требует унификации подходов и применения одинаковых технологий во всей системе.
Нарушение принципа изоляции	Включение общих механизмов аутентификации и авторизации нарушает принцип независимости микросервисов, добавляя им дополнительные задачи и усложняя их управление.
Увеличение сложности системы	Микросервисы взаимодействуют как с пользователями, так и между собой, а также с внешними системами, что повышает сложность реализации и поддержки механизмов аутентификации.

В таблице 2 будут представлены существующие подходы к аутентификации в микросервисных системах.

Таблица 2

Существующие стратегии аутентификации в микросервисной архитектуре [3, с. 25-36]

Стратегия	Описание
Аутентификация на уровне периферии	Контроль доступа осуществляется через API-шлюз, который управляет аутентификацией и авторизацией для всех микросервисов. Однако, при нарушении безопасности на уровне шлюза возникает угроза доступа ко всей системе.
Аутентификация на уровне сервиса	Каждый микросервис самостоятельно контролирует доступ, предоставляя больше гибкости в настройке правил безопасности. Включает управление политиками доступа и принятие решений внутри каждого сервиса.
Контекстная аутентификация	Учитывает специфическую информацию о пользователе (роль, местоположение, время), передаваемую через токены. Обеспечивает точный контроль доступа, но требует строгой защиты токенов от атак.

В таблице 3 будут описаны основные технические методы аутентификации.

Таблица 3

Основные методы аутентификации [3, с. 25-36]

Метод	Описание
Единый вход (SSO)	Позволяет пользователю или системе единожды аутентифицироваться для получения доступа к нескольким сервисам. В микросервисной архитектуре применяется как для конечных пользователей, так и для сервисов, взаимодействующих друг с другом.
JSON Web Token (JWT)	JWT предоставляет безопасный способ передачи информации между микросервисами в зашифрованном виде. Это эффективный метод обмена данными, такими как идентификаторы пользователей или систем.
OAuth 2.0 и OpenID Connect	OAuth 2.0 защищает взаимодействие между клиентами и серверами API, а OpenID Connect (OIDC) расширяет возможности, обеспечивая работу с федеративными удостоверениями, что упрощает интеграцию с различными поставщиками идентификаций.

Таким образом, выбор подхода к аутентификации микросервисов зависит от архитектуры системы, её масштабов и требований к безопасности. Каждый из методов имеет свои преимущества и ограничения, которые следует учитывать при проектировании системы.

2. Безопасная аутентификация и авторизация микросервисов

Аутентификация и авторизация в контексте микросервисов играют ключевую роль в обеспечении безопасности и правильного функционирования системы. Эти процессы позволяют контролировать доступ к различным сервисам, обеспечивая, что только уполномоченные

пользователи могут взаимодействовать с ресурсами системы.

Аутентификация представляет собой процесс, при котором проверяется личность пользователя или службы. В архитектуре микросервисов каждая служба или пользователь может потребовать проверки подлинности для доступа к ресурсам других сервисов. Это достигается путем предоставления специальных данных, таких как логины, пароли, ключи API или токены безопасности. Ниже на рисунке 1 будет отражена архитектура и стратегии аутентификации.

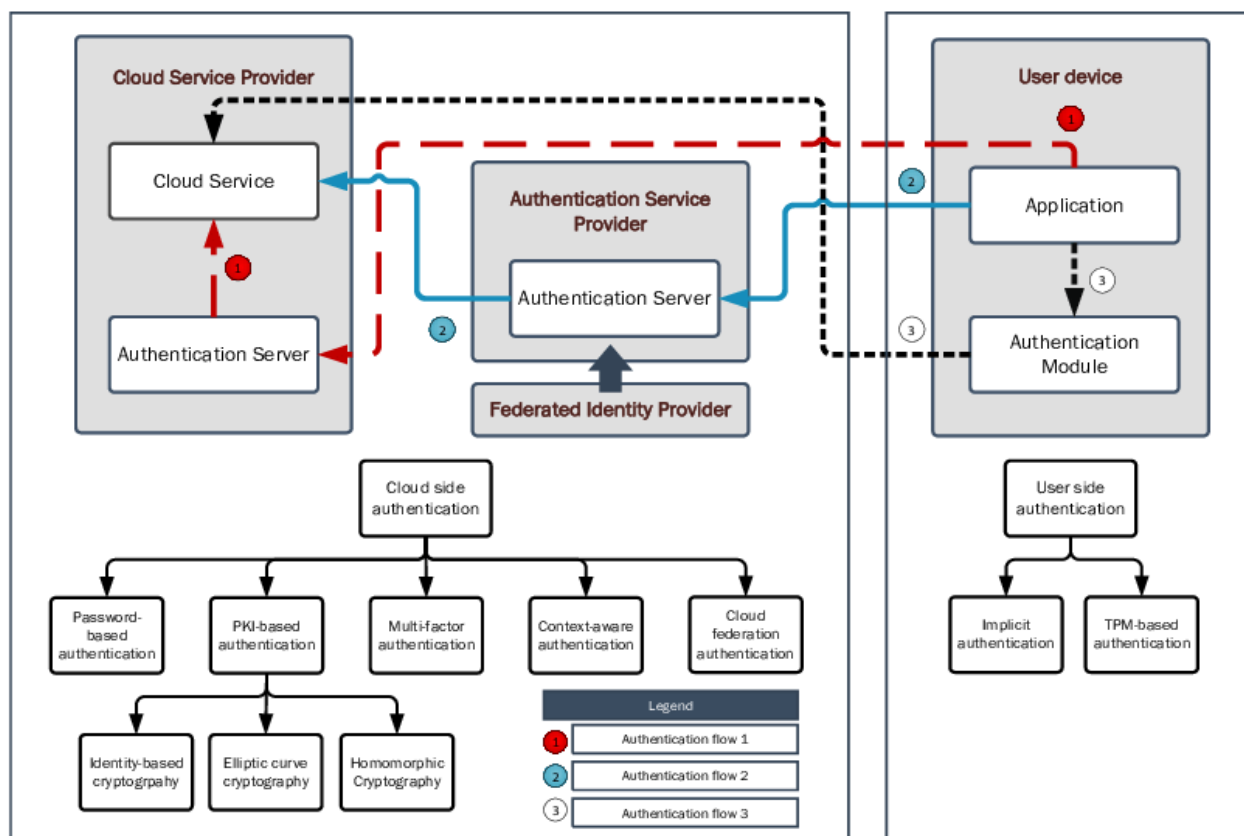


Рис. 1. Архитектура и стратегии аутентификации [7]

Популярными механизмами аутентификации в микросервисных системах являются JSON Web Tokens (JWT), OAuth и OpenID Connect. Эти механизмы позволяют гарантировать, что только авторизованные субъекты могут взаимодействовать с ресурсами системы [4, с. 236-240].

Безопасность в микросервисах особенно важна из-за их распределенной структуры. Каждый сервис может находиться в различных местах и взаимодействовать по сети, что увеличивает потенциальные риски атак. Для минимизации угроз необходимо применять эффективные меры безопасности, такие как шифрование данных, контроль доступа и защита передаваемых данных.

Микросервисные системы предоставляют различные механизмы аутентификации, адаптированные под конкретные задачи. Один из распространенных подходов – использование JWT, который позволяет передавать токены, содержащие закодированные данные о пользователе. Также широко используются OAuth 2.0 для делегирования авторизации и OpenID Connect, который обеспечивает одноразовую проверку подлинности и вход в систему.

Единый вход (SSO) позволяет пользователю один раз пройти аутентификацию для получения доступа ко множеству микросервисов. Это

упрощает управление пользователями и улучшает безопасность, так как все операции проходят через доверенного поставщика удостоверений, который может использовать многофакторную аутентификацию для дополнительной защиты.

При разработке систем аутентификации для микросервисов важно учитывать баланс между безопасностью и масштабируемостью. Центральная аутентификация через единый сервер предоставляет удобство, но может быть узким местом. Децентрализованные методы аутентификации позволяют более гибко распределять нагрузку, но требуют строгих мер безопасности на каждом сервисе [5].

Авторизация, в свою очередь, определяет, какие действия могут выполнять аутентифицированные пользователи или системы. В микросервисной архитектуре этот процесс также требует дополнительных мер, так как каждый микросервис должен иметь четко определенные правила доступа для различных пользователей. Для этого применяются различные подходы, такие как политика контроля на уровне каждого микросервиса или использование централизованных сервисов авторизации, что позволяет унифицировать и упростить управление доступом. Ниже на рисунке 2 будет отражен процесс авторизации на уровне сервиса.

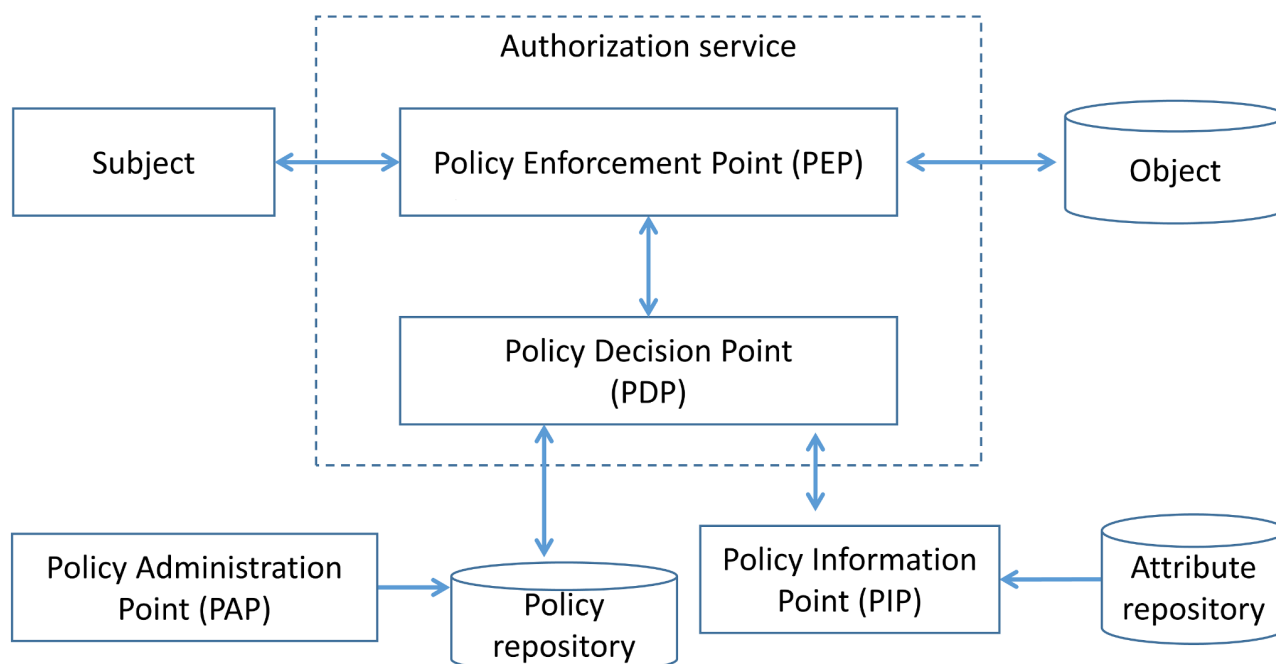


Рис. 2. Авторизация на уровне сервиса [8]

Ключевым фактором безопасной аутентификации и авторизации микросервисов является минимизация рисков, связанных с компрометацией одного из сервисов. Это достигается посредством изоляции данных и ресурсов, распределения прав доступа в соответствии с принципом минимальных привилегий, а также постоянного мониторинга активности и выявления аномалий. Важно, чтобы механизмы безопасности были интегрированы на всех уровнях системы, начиная от взаимодействия микросервисов и заканчивая пользовательским доступом.

Кроме того, внедрение безопасных механизмов аутентификации и авторизации требует соблюдения стандартов безопасности и соответствия нормативным требованиям. Например, такие регламенты, как GDPR и PCI DSS, обязывают компании использовать строгие меры защиты данных, что особенно актуально для систем с микросервисной архитектурой. Реализация безопасной аутентификации и авторизации, таким образом, должна учитывать не только технологические, но и правовые аспекты.

Таким образом, безопасность аутентификации и авторизации микросервисов является важной составляющей общей безопасности системы. Правильное применение современных технологий, таких как JWT и OAuth 2.0, в сочетании с принципами минимальных привилегий и централизованным управлением,

позволяет значительно повысить защиту распределенных приложений.

3. Обеспечение безопасности взаимодействия микросервисов

Обеспечение безопасности взаимодействия микросервисов является одной из ключевых задач при проектировании современных распределённых систем. С увеличением количества микросервисов в архитектуре возрастает риск уязвимостей, что требует внедрения многоуровневых мер безопасности. Этот процесс требует комплексного подхода, охватывающего как сетевые аспекты, так и внутренние механизмы безопасности самих микросервисов.

Одним из фундаментальных принципов обеспечения безопасности взаимодействия является аутентификация и авторизация. Аутентификация обеспечивает идентификацию каждого сервиса, что исключает возможность доступа неавторизованных участников. Авторизация, в свою очередь, гарантирует, что каждый сервис имеет право на выполнение только тех операций, которые ему предписаны. Использование таких протоколов, как OAuth 2.0 и OpenID Connect, позволяет централизованно управлять доступом и упрощает интеграцию с внешними системами.

Шифрование данных, передаваемых между микросервисами, также играет важную роль в обеспечении безопасности. Протоколы TLS (Transport Layer Security) обеспечивают защиту данных на уровне транспортного слоя,

предотвращая их перехват и подмену злоумышленниками. Важно не только шифровать каналы связи между микросервисами, но и управлять ключами шифрования, гарантируя их регулярную ротацию и защиту от компрометации.

Мониторинг и аудит являются дополнительными механизмами для контроля безопасности микросервисной архитектуры. Мониторинговые системы отслеживают активность сервисов, выявляя аномалии, которые могут свидетельствовать о попытках взлома или несанкционированного доступа. Аудит взаимодействий между сервисами помогает в анализе инцидентов, предоставляя детальные логи и временные метки всех операций.

Кроме того, важным аспектом является обеспечение изоляции сервисов. В случае компрометации одного микросервиса необходимо минимизировать возможность распространения угрозы на другие части системы. Этого можно достичь с помощью строгого ограничения сетевых прав доступа между микросервисами, применения принципов минимальных привилегий и контейнеризации с использованием таких инструментов, как Kubernetes [6].

Заключение

В заключение можно отметить, что безопасность микросервисов требует системного подхода, объединяющего методы управления секретами и безопасной аутентификации. Эффективные механизмы защиты, такие как JWT, OAuth 2.0 и mTLS, обеспечивают надежную аутентификацию и предотвращают утечки данных. Управление секретами, включающее ротацию ключей и токенов, позволяет минимизировать риски, связанные с несанкционированным доступом к ресурсам. Регулярное обновление политик безопасности и внедрение систем управления доступом, таких как RBAC и ABAC, помогают повысить уровень защиты системы. Таким образом, только комплексный подход к проектированию и управлению безопасностью микросервисов может гарантировать их стабильное и безопасное функционирование в условиях растущих киберугроз.

Литература

1. Зими́на К.И., Лапо́нина О.Р. Механизмы межсервисной аутентификации в приложениях с микросервисной архитектурой // International Journal of Open Information Technologies. – 2023. – Т. 11. – №. 5. – С. 146-154.
2. Самородских И.Л. Системы управление секретами // E-Scio. – 2020. – №. 3 (42). – С. 267-281.
3. Гольчевский Ю.В., Ермоленко А.В. Актуальность использования микросервисов при разработке информационных систем // Вестник Сыктывкарского университета. Серия 1. Математика. Механика. Информатика. – 2020. – №. 2 (35). – С. 25-36.
4. Прокопенко Д.В. Применение современных методов обучения для разработки микросервисов на примере языка Golang // Трансформация механико-математического и IT-образования в условиях цифровизации. – 2023. – С. 236-240.
5. Аутентификация и авторизация в проекте с микросервисной архитектурой: стратегии, практический пример. [Электронный ресурс] Режим доступа: <https://habr.com/ru/companies/spectr/articles/715290/> (дата обращения 13.09.2024).
6. Best Practices to Secure Microservices with Spring Security. [Электронный ресурс] Режим доступа: <https://www.geeksforgeeks.org/best-practices-to-secure-microservices-with-spring-security/> (дата обращения 13.09.2024).
7. Архитектура и стратегии аутентификации. [Электронный ресурс] Режим доступа: https://www.researchgate.net/figure/Authentication-architecture-and-strategies_fig1_317213681 (дата обращения 24.09.2024).
8. Справочник security-архитектора: обзор подходов к реализации аутентификации и авторизации в микросервисных системах. [Электронный ресурс] Режим доступа: <https://habr.com/ru/companies/huawei/articles/527098/?mobile=yes> (дата обращения 24.09.2024).

GLUMOV Konstantin

Lead Software Engineer, Alfa-Bank, Russia, Perm

SECURITY OF MICROSERVICES: SECRET MANAGEMENT AND SECURE AUTHENTICATION

Abstract. *Microservices security plays a key role in modern distributed systems, where each component requires reliable protection to prevent data leaks and unauthorized access. The main security aspects of microservices are secret management and secure authentication, which allow you to control the interaction between services and access to resources. Secret management includes methods for storing and rotating sensitive information such as keys and tokens, which reduces the risks of attacks and data leaks. Secure authentication is carried out through mechanisms such as JSON Web Tokens (JWT), OAuth 2.0 and Mutual TLS (mTLS), which provide protection against attacks on inter-service interactions. Regular updating of security methods, as well as the introduction of integrated solutions such as role-based access control systems (RBAC) and attributes (ABAC), are essential measures to increase the level of protection.*

Keywords: *microservices, secret management, authentication, JWT, OAuth 2.0, mTLS, data security.*