Пахомова П. В. Современные подходы к обеспечению безопасности в KTOR: JWT, OAUTH, LDAP и KEYCLOAK // Международный журнал информационных технологий и энергоэффективности.– 2025. –Т. 10 № 3(53) с. 67–77

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.056:004.438

СОВРЕМЕННЫЕ ПОДХОДЫ К ОБЕСПЕЧЕНИЮ БЕЗОПАСНОСТИ В KTOR: JWT, OAUTH, LDAP И KEYCLOAK

Пахомова П. В.
ФГБОУ ВО "ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ", Воронеж, Россия (394018, Воронежская область, город Воронеж, Университетская пл., д. 1), e-mail: polinapahomova12@mail.ru

Безопасность программных приложений является важнейшей проблемой в современной разработке программного обеспечения, особенно в условиях преобладания распределённых систем и микросервисов. Ktor выделяется как набирающий популярность фреймворк разработки с поддержкой экосистемы Java, которая предлагает широкий спектр возможностей для реализации надёжных механизмов безопасности. В этой статье основное внимание уделено изучению современных передовых подходов обеспечения безопасности корпоративных сред с использованием Ktor; в частности, будут обсуждаться такие темы, как веб-токен JSON (JWT), OAuth 2.0, облегчённый протокол доступа к каталогам (LDAP) и решения на основе Keycloak. Использование JWT позволяет реализовать аутентификацию без состояния (stateless authentication), что особенно важно в контексте распределённых систем. OAuth 2.0 служит стандартом авторизации, который предоставляет пользователям доступ к общим ресурсам, одновременно защищая конфиденциальные учётные данные пользователя от ненужного раскрытия. LDAP находит практическое применение, облегчая централизованное управление идентификационными данными и привилегированными доступами, что особенно выгодно при работе со сложными организационными структурами большого масштаба. Являясь платформенным решением с открытым исходным кодом, специально разработанным для распознавания личности и управляемой авторизации, Keycloak предоставляет службы поддержки, соответствующие общепринятым протоколам, таким как OpenID Connect или SAML; надёжные решения, необходимые для обеспечения чётко регламентированных конфиденциальных взаимодействий, например, в ситуациях, требующих надёжной проверки, вызванных как внутренними потребностями, так и внешними партнёрами по сети. В рамках Ktor данные механизмы интегрируются с помощью существующих библиотек. В этой статье в рамках Ktor исследовано, каким образом передовые технологии могут быть надлежащим образом использованы для создания безопасных и масштабируемых приложений. В ходе анализа подробно рассматривается каждый из этих механизмов, описываются их преимущества и проблемы, а также предложения по их решению и интеграции при возникновении сложных бизнес-сценариев. В конечном счёте, это исследование предназначено для улучшения понимания прогрессивных мер безопасности, тем самым предоставляя разработчикам расширенные возможности для создания более устойчивых прикладных решений.

Ключевые слова: Кибербезопасность, Ktor, Ktor Authentication, JWT, OAuth, LDAP, Keycloak.

MODERN APPROACHES TO SECURITY IN KTOR: JWT, OAUTH, LDAP AND KEYCLOAK

Pakhomova P. V.
VORONEZH STATE UNIVERSITY, Voronezh, Russia (394018, Voronezh region, Voronezh city, Universitetskaya square, 1), e-mail: polinapahomova12@mail.ru

Software application security is a critical issue in modern software development, especially with the prevalence of distributed systems and microservices. Ktor stands out as a gaining popularity as a development framework with support for the Java ecosystem, which offers a wide range of options for implementing robust security mechanisms. This paper focuses on exploring current best practices for securing enterprise environments using Ktor; in particular, topics such as JSON Web Token (JWT), OAuth 2.0, Lightweight Directory Access Protocol (LDAP),

and Keycloak-based solutions will be discussed. The use of JWT enables stateful authentication, which is especially important in the context of distributed systems. OAuth 2.0 serves as an authorization standard that gives users access to shared resources while protecting sensitive user credentials from unnecessary disclosure. LDAP finds practical applications by facilitating centralized identity and privileged access management, which is especially beneficial when dealing with complex, large-scale organizational structures. As an open source platform solution specifically designed for identity recognition and managed authorization, Keycloak provides support services that are compliant with common protocols such as OpenID Connect or SAML; robust solutions necessary to ensure highly regulated sensitive interactions, such as those requiring strong verification, whether driven by internal needs or external network partners. The Ktor framework integrates these mechanisms using existing libraries. In this paper, the Ktor framework explores how advanced technologies can be appropriately utilized to create secure and scalable applications. The analysis examines each of these mechanisms in detail, describing their benefits and challenges, as well as suggestions for addressing and integrating them when complex business scenarios arise. Ultimately, this research is intended to improve the understanding of progressive security measures, thereby providing developers with enhanced capabilities to create more resilient application solutions.

Keywords: Cybersecurity, Ktor, Ktor Authentication, JWT, OAuth, LDAP, Keycloak.

Введение

Платформа Ktor становится наиболее популярным компонентом в разработке современных приложений. Впервые она была представлена в 2018 году и значительно улучшила развитие Kotlin как языка программирования для серверной разработки. В отличие от более традиционного подхода на основе Spring, Ktor предоставляет более легковесную архитектуру, которая может быть особенно полезна для приложений с высокой нагрузкой, требующих быстрого отклика. Одним из его главных преимуществ является его способность интегрироваться с фреймворком Spring и Java.

Если говорить о безопасности, то ключевой функцией в рамках этой платформы является Ktor Authentication; влиятельная и персонализированная система аутентификации и контроля доступа, которая играет решающую роль в защите приложений от распространенных угроз безопасности.

JSON Web Token (JYVT) представляет а широко распространённую и устоявшуюся среду для безопасного обмена информацией в виде объектов JSON, эти токены выделяются своей компактностью, совместимостью с URL-адресами, поддержкой цифровой подписи, что приводит к улучшенным функциям безопасности, следовательно, является идеальным вариантом в контексте аутентификации без состояния в современных веб-приложениях [1] [10]. JWT обеспечивают бесперебойные механизмы, совместимые с общим решением для обеспечения безопасности несессионных функциональных возможностей, разработанных на основе методологии программирования Spring.

Платформа OAuth 2.0 служит средством авторизации, которое позволяет приложениям получать ограниченный доступ к учётным записям пользователей в службе HTTP. Этот процесс включает делегирование задач аутентификации пользователя службе хостинга [12]. Что касается Ktor Authentication, OAuth 2.0 представляет собой мощный метод защиты RESTful-сервисов и API-интерфейсов за счёт передачи функций аутентификации пользователей на аутсорсинг внешнему серверу авторизации.

Облегчённый протокол доступа к каталогам (LDAP) - широко используемый протокол для доступа к распределённым информационным службам каталогов и их обслуживания по сети Internet Protocol (IP). В Ktor Authentication LDAP играет ключевую роль в управлении идентификациями пользователей и контроле доступа, особенно в обширных корпоративных средах.

Keycloak - это решение с открытым исходным кодом для управления идентификацией и

доступом, которое обслуживает современные приложения и службы. Он обладает широким спектром функций, включая единый вход (SSO), посредничество при идентификации личности, а также возможности входа в систему через социальные сети. Keycloak эффективно интегрируется с платформами, предоставляя разработчикам беспрепятственный доступ к различным механизмам аутентификации наряду с протоколами авторизации, которые повышают параметры безопасности в среде их приложений.

Включение сложных механизмов безопасности, а именно JWT, OAuth, LDAP и Keycloak, в Ktor с помощью Ktor Authentication олицетворяет значительный прогресс в создании безопасных приложений на Kotlin. Такое объединение не только упрощает процесс внедрения сложных требований безопасности, но и гарантирует устойчивость этих приложений к широкому спектру атак.

1. JWT

Использование JWT приобрело значительное значение в современных практиках веб-безопасности, поскольку оно обеспечивает краткий и автономный подход к передаче информации между участниками через объект JSON, который обеспечивает конфиденциальность высокого уровня. JWT предназначены для включения механизмов подписи, что может быть достигнуто путём использования либо криптографии с секретным ключом с использованием алгоритма HMAC, либо публично-частного шифрования с использованием алгоритмов RSA или ECDSA, тем самым обеспечивая целостность данных во время передачи. При наличии таких протоколов аутентификации, которые не зависят от хранилища состояний сеанса, JWT обслуживает подходящие сценарии, такие как RESTful API [1].

JVT обычно состоит из трёх компонентов: заголовка, полезной нагрузки и подписи. Заголовок обычно состоит из двух частей, которые включают тип токена - то есть JWT - и оптимизируемый алгоритм подписи. Полезная нагрузка включает в себя утверждения относительно объекта (обычно пользователя) наряду с дополнительными данными. Наконец, чтобы гарантировать, что после проверки не было внесено никаких изменений, спользуются подписи для обеспечения подлинности с течением времени.

Ktor Authentication предлагает всестороннюю поддержку JWT. Его включение предоставляет разработчикам возможность решать проблемы аутентификации пользователей и авторизации с помощью непостоянного подхода, что оказывается существенно выгодным для RESTful-приложений. С помощью Ktor Authentication процедуры проверки JWT становятся доступными; они гарантируют, что JWT имеют правильное формирование, проверяют их подпись, а также достоверность. При внедрении JWT в приложение Ktor разработчики обычно полагаются на такие известные библиотеки, такие как io.ktor:ktor-auth:2.x.x или io.ktor:ktor-auth-jwt:2.x.x, они содержат основные ресурсы, необходимые для эффективного создания, анализа и аутентификации JWT. Процесс реализации включает в себя настройку JwtTokenStore и JwtAccessTokenConverter с одновременным предоставлением дополнительного TokenEnhancer для дополнения информации в токене. Кроме того, крайне важно, чтобы разработчики настроили менеджер аутентификации в дополнение к описанию ограничений безопасности, налагаемых на конечные точки (endpoints), используемые приложением.

Протокол JWT особенно полезен в ситуациях, когда важно установить подлинность

пользователя и необходимые доступы к определённым ресурсам. Это служит дополнительным преимуществом в архитектуре микросервисов, где безопасная межсервисная коммуникация становится обязательной. Для оптимального использования JWT с Ktor установленные рекомендации включают развёртывание HTTPS для защиты токенов от угроз перехвата, установление реалистичных сроков истечения срока действия токенов и разумное управление информацией, относящейся к разделам полезной нагрузки, чтобы конфиденциальные данные не могли быть случайно раскрыты.

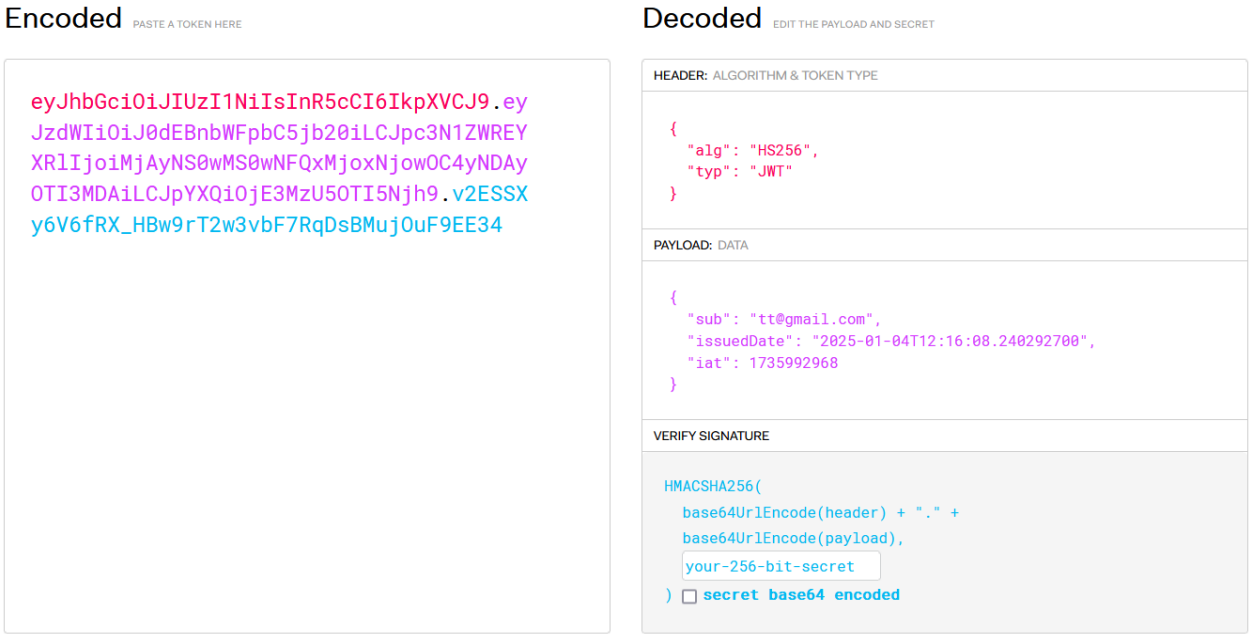

Рисунок 1 - Структура JWT в формате JSON.

Включение JSON в Ktor Authentication обеспечивает надёжный и эффективный подход к управлению аутентификацией и авторизацией в неизменяемом интерфейсе. Его универсальность в сочетании с удобством в использовании делают его оптимальной альтернативой для защиты приложений, основанных на Ktor, особенно тех, которые структурированы вокруг микросервисов, а также услуг RESTful.

2. **OAuth 2.0**

OAuth 2.0 - это платформа авторизации, которая предоставляет сторонним приложениям ограниченный доступ к HTTP-сервису, будь то через владельца ресурса или автономное получение доступа. Его отличие от аутентификации делает его незаменимым в ситуациях, когда пользовательские данные должны запрашиваться у других служб без ущерба для их соответствующих учётных данных [3].

OAuth 2.0 вводит несколько ролей:

- владелец: Пользователь, который разрешает приложению доступ к своей учётной записи;
- сервер ресурсов: На нём хранятся защищённые пользовательские данные;
- клиент: Приложение, запрашивающее доступ к учётной записи пользователя;

Сервер авторизации проверяет личность владельца ресурса и выдаёт токены доступа.

OAuth 2.0 определяет четыре основных типа грантов, подходящих для различных типов

приложений:

- предоставление кода авторизации: Идеально подходит для клиентов, которые могут безопасно хранить клиентские секреты;
- неявное предоставление: Предназначено для клиентов, которые не могут безопасно хранить клиентские секреты;
- предоставление учётных данных с паролем владельца ресурса: Подходит для клиентов с высоким уровнем доверия;
- предоставление учётных данных клиента: Используется для доступа приложений к их собственным ресурсам.

Поддержка OAuth 2.0 в Ktor Authentication упрощает реализацию этих типов грантов:

- конфигурация серверов авторизации и ресурсов: В Ktor для настройки аутентификации через OAuth 2.0 можно использовать установку обработчиков, например, через Authentication с использованием обработчиков типа oauth (используются authorizeUrl, accessTokenUrl, clientId, clientSecret и области доступа);
- ведения о клиенте: Можно настроить клиентские данные, такие как client_id, client_secret, и области доступа (scopes), с помощью параметров в конфигурации OAuth 2.0. Эти данные необходимы для успешной аутентификации и получения токенов от сервера авторизации;
- управление токенами: Внедрение хранилища токенов и службы токенов для управления генерацией, сроком действия и обновлением токенов;
- конфигурация безопасности: Определение ограничения безопасности для различных конечных точек, какие из них защищены, а какие общедоступны.

Так же Ktor Authentication OAuth 2.0 предоставляет несколько расширенных функций, среди них - усилители пользовательских токенов, которые позволяют добавлять дополнительные данные к токенам OAuth, а также обработчики утверждений, предназначенные для управления утверждениями пользователей при выдаче токенов. Доступны конечные точки (endpoints) для обработки перенаправлений пользователя после аутентификации и для предоставления информации о пользователе клиентам. Эти функции значительно расширяют возможности аутентификации и позволяют гибко настраивать процессы безопасности в приложениях.

К числу передовых практик, которые способствуют повышению уровня безопасности, относятся защита клиентских секретов, что требует их безопасного хранения и недопущения раскрытия в клиентском коде. Важно также проверять URI перенаправления, чтобы все перенаправления были предварительно зарегистрированы и проверены, что исключает риск несанкционированных редиректов. Для обеспечения безопасности токенов необходимо использовать HTTPS во всех коммуникациях, связанных с токенами и учетными данными, а также внедрять стратегии отзыва и ротации токенов для предотвращения утечек и атак [16].

Благодаря стратегическому использованию возможностей конфигурации и автоматизации Ktor разработчики имеют возможность адаптировать реализацию OAuth 2.0 для различных требований приложений, обеспечивая при этом оптимальную функциональность и соблюдение мер безопасности.

3. LDAP

Облегчённый протокол доступа к каталогам (LDAP) - широко используемый протокол, предназначенный для доступа к распределённым информационным службам каталогов и поддержания их функциональности в сети по интернет-протоколу (IP). LDAP служит для различных целей, включая, но не ограничиваясь, поиск по электронной почте, процессы аутентификации, а также организацию данных компании. Это оказалось особенно выгодным с точки зрения облегчения управления пользовательской информацией наряду с обеспечением возможностей аутентификации и авторизации в обширных корпоративных средах [4] [13].

В Ktor LDAP функционирует как фундаментальный источник как пользовательских данных, так и аутентификации. Благодаря широкой поддержке он эффективно облегчает бесшовную интеграцию с уже существующими серверами LDAP. Следовательно, эта синергия предоставляет приложениям возможность проверять пользователей, одновременно извлекая соответствующую информацию о роли пользователя, которая была сохранена в независимом каталоге в базе данных LDAP.

Реализация аутентификации LDAP в приложении Ktor обычно включает в себя несколько этапов:

- зависимости: Включите в свой проект зависимости Ktor LDAP;
- конфигурация источника LDAP Context: Настройте LdapContextSource для указания URL-адреса и базового суффикса сервера LDAP;
- провайдер аутентификации: Настройте LdapAuthenticationProvider для обработки всех запросов аутентификации. Это включает в себя указание базы поиска пользователя, фильтра поиска пользователя и, при необходимости, базы группового поиска и фильтра группового поиска;
- сопоставление сведений о пользователе: Это может включать использование собственных классов для сопоставления данных из LDAP с объектами пользователя в Ktor, а также настройку ролей и прав доступа, что можно сделать с помощью специализированных популяторов ролей и мапперов данных;
- конфигурация безопасности: Определите ограничения безопасности в конфигурации Ktor, указав, какие конечные точки (endpoints) защищены, а какие общедоступны.

Так же интеграция с LDAP может включать расширенные функции, такие как реализация службы для более сложного поиска пользовательской информации. Это позволяет улучшить работу с пользовательскими данными, а также внедрить механизмы для настройки политик паролей и обработки исключений, связанных с безопасностью паролей. Дополнительно, использование LDAP-операций может быть реализовано через специализированные шаблоны, такие как LdapTemplate, которые предоставляют более гибкие и сложные возможности для работы с LDAP, помимо базовой аутентификации.

При внедрении LDAP в Ktor важно следовать лучшим практикам безопасности. Это включает использование защищенного канала связи с сервером LDAP (например, через LDAP с SSL), чтобы обеспечить безопасность передаваемых данных. Кроме того, критически важно правильно обрабатывать пароли, избегая их небезопасного хранения или регистрации. Также необходимо защищать приложение от атак с использованием LDAP-инъекций, что достигается проверкой и очисткой входных данных для предотвращения несанкционированных попыток доступа [4].

Включение LDAP в приложение Ktor представляет собой высокоэффективный подход к

управлению аутентификацией пользователей и авторизацией в корпоративных приложениях. Благодаря выгодному использованию встроенной поддержки LDAP в Ktor - разработчики программного обеспечения смогут устанавливать бесперебойную связь с каталогами LDAP, одновременно повышая безопасность и масштабируемость в рамках соответствующих прикладных задач.

4. Keycloak

Keycloak - это современное решение для управления идентификацией и доступом, разработанное Red Hat в виде программного обеспечения с открытым исходным кодом. Его основная цель заключается в упрощении интеграции стандартных протоколов, таких как OpenID Connect и SAML 2.0, в процессы аутентификации при одновременном упрощении процедур авторизации. В дополнение к возможностям централизованной консоли управления, касающимся идентификации пользователей, Keycloak предоставляет функции, обеспечивающие эффективную поддержку единого входа, двухфакторной аутентификации и функций социального входа. Эти расширенные возможности безопасности делают его особенно подходящим для обеспечения целостности современных приложений в различных сервисных средах, где высоко ценятся индивидуальные решения по управлению идентификацией [5] [11].

В контексте Ktor Authentication - Keycloak даёт приложениям Ktor возможность делегировать свои протоколы аутентификации пользователей и авторизации непосредственно Keycloak-динамике, которая впоследствии упрощает усилия по управлению безопасностью. Кроме того, эта интеграция предоставляет указанным приложениям доступ к расширенным функциям, эксклюзивным для Keycloak; примеры включают единый вход, меры аутентификации на основе токенов в дополнение к возможностям объединения пользователей.

Внедрение Keycloak в приложении Ktor обычно включает в себя несколько этапов:

- зависимости: Включите зависимость Keycloak в свой проект;
- настройка сервера перехвата ключей: Настройте сервер перехвата ключей, определив области, клиентов, роли и пользователей;
- конфигурация приложения: Настройте приложение Ktor на использование Keycloak для аутентификации и авторизации. Это включает в себя настройку свойств скрытия ключей в application.conf или application.yml файле;
- конфигурация безопасности: Настройте Ktor на использование адаптера Keycloak для аутентификации. Это включает в себя определение ограничений безопасности и указание защищенных ресурсов в приложении;
- управление пользователями и ролями: Используйте консоль администрирования Keycloak для управления пользователями и ролями, которые могут быть сопоставлены с полномочиями Ktor Authentication.

Так же Keycloak предоставляет расширенные возможности настройки, такие как добавление и управление пользовательскими атрибутами, что позволяет гибко управлять данными пользователей. Можно настроить посредничество при идентификации, чтобы Keycloak выполнял роль промежуточного звена для аутентификации между различными поставщиками идентификационных данных, обеспечивая единую точку входа. Вдобавок, существует возможность настройки темы для Keycloak, это позволит персонализировать внешний вид страниц входа и электронных писем, улучшая пользовательский интерфейс и

опыт.

При интеграции следует придерживаться ряда рекомендаций для обеспечения безопасности. В первую очередь, необходимо обеспечить безопасную коммуникацию между приложением Ktor и сервером Keycloak, используя HTTPS для защиты данных. Также важно безопасно управлять клиентскими секретами, избегая их утечек и несанкционированного доступа. Наконец, следует внедрить надёжную проверку токенов, чтобы убедиться, что доступ к защищенным ресурсам имеет только авторизованный пользователь, предотвращая возможность несанкционированного доступа [18].

Такая интеграция предлагает мощное и гибкое решение для управления аутентификацией и авторизацией в приложениях. Используя Keycloak, разработчики могут повысить безопасность своих приложений, используя преимущества таких функций, как единый вход, аутентификация на основе токенов и федерация пользователей.

5. Обзор литературы

JSON Web Tokens (JWT) остаются важнейшим инструментом в обеспечении безопасности веб-приложений. В статье, опубликованной в 2020 году, подчёркивается, что использование JWT для аутентификации и авторизации позволяет значительно повысить защиту от атак, таких как фальсификация токенов и их повторное использование. В частности, исследование указывает на важность использования JWT вместе с механизмами мониторинга активности пользователей, что улучшает защиту путём обнаружения аномальных действий, таких как попытки несанкционированного доступа. Это, в свою очередь, повышает общую безопасность системы [1].

В научной статье 2023 года обсуждается использование JSON Web Token (JWT) для аутентификации между сервисами в микросервисных приложениях. Отмечается, что JWT позволяет реализовать безсессионный механизм аутентификации, что особенно важно для высоконагруженных систем, требующих масштабируемости и эффективности. Авторы также рассматривают преимущества использования JWT в таких сценариях и приводят примеры его применения [2].

Другим важным аспектом является интеграция OAuth 2.0 в микросервисные архитектуры. Согласно статье 2023 года, использование OAuth 2.0 в таких приложениях позволяет обеспечить высокий уровень безопасности, а также гарантировать контроль над доступом через различные типы грантов. Предлагается модель управления доступом на основе атрибутов для кросс-доменных API, включающая архитектурные решения и принципы ABAC и OAuth. ABAC-сервис авторизации рассматривается как микросервис или набор микросервисов, что обеспечивает совместимость с приложениями, построенными на микросервисной архитектуре. Системы, использующие Ktor Authentication, могут гибко адаптировать авторизацию в зависимости от нужд приложения и пользователя. В этом контексте важно отметить, что OAuth 2.0 помогает интегрировать токенизацию и права доступа на уровне отдельных микросервисов, что особенно полезно для распределённых систем [3].

Параллельно с OAuth 2.0 стоит рассмотреть и возможности LDAP (Lightweight Directory Access Protocol) для управления аутентификацией и авторизацией пользователей. Статья 2023 года показала, что LDAP интегрируется с различными веб-фреймворками, в том числе с Ktor, для обеспечения централизованного управления учётными записями в крупных организациях

[4]. Важной деталью является возможность интеграции LDAP с другими системами безопасности, такими как SAML и OpenID Connect, что расширяет возможности для настройки гибкой и безопасной авторизации в многослойных инфраструктурах [17].

Для более сложных сценариев аутентификации и авторизации в корпоративных системах, Keycloak становится ключевым компонентом. В статьях 2023 года утверждается, что Keycloak позволяет обеспечить управление пользователями, автоматическое распределение ролей и реализацию политики безопасности в реальном времени, что делает его отличным выбором для предприятий, которым необходимы надёжные механизмы защиты API и пользовательских данных [5] [11].

Наконец, использование JWT в легковесных протоколах обмена сообщениями, таких как MQTT, подтверждается исследованием 2019 года, в котором анализировались возможности аутентификации и авторизации для устройств в IoT-средах. JWT, будучи компактным и быстрым для обработки, идеально подходит для работы с протоколами с ограниченными ресурсами, такими как MQTT, и может использоваться для безопасного обмена сообщениями в реальном времени. Исследование показало, что использование JWT совместно с MQTT позволяет создавать высокозащищенные IoT-системы, что актуально для приложений, работающих в облачных и распределённых средах [6].

Заключение

Благодаря своей модульной архитектуре и возможностям настройки, Ktor позволяет разработчикам интегрировать различные механизмы безопасности, при этом каждый компонент обладает своими преимуществами и недостатками. В частности, JWT может похвастаться функциональностью без сохранения состояния, а также возможностью масштабирования, что делает его подходящим для современных веб-приложений; однако тщательный мониторинг безопасности токенов имеет решающее значение для предотвращения любой потенциальной уязвимости или риска кражи. OAuth 2.0 служит обширной, но способной к адаптации структурой авторизации, подходящей для различных типов приложений; тем не менее сложность может представлять проблемы во время внедрения, в то время как строгое соблюдение рекомендаций по передовой практике должно постоянно поддерживаться на протяжении всей работы. LDAP превосходен в управлении идентификациями пользователей в обширных операционных средах с помощью централизованных механизмов аутентификации, но настройка может создавать значительные логистические препятствия, особенно когда сталкиваются с быстро меняющимися наборами данных, требующими постоянной корректировки по сравнению с имеющимися альтернативными решениями. Наконец, интеграция Keycloak в микросервисные архитектуры позволяет проще обрабатывать комплексные функции управления доступом к идентификаторам, значительно сокращая потребности в администрировании, хотя одновременно предъявляет дополнительные требования к конфигурации сервера, возможно, приводя к проблемам снижения производительности, без уделения тщательного внимания оптимизации и определения эффективных компромиссов относительно требуемых конкретных ограничений пропускной способности инфраструктуры. Keycloak, предоставляемый посредством интеграции, позволяет эффективно запускать все эти методы с использованием Ktor Authentication, обеспечивает надёжную общую защиту системы, обеспечивающую максимальное снижение негативных уязвимостей, возникающих в

результате оптимального развертывания, следуя исчерпывающему пониманию фундаментальных принципов, определяющих надежное безопасное управление операциями экосистемы, широко применимых во многих отраслевых вертикалях, извлекающих из этого немалую выгоду после успешного завершения внедрения, достижения стратегических бизнес-целей, нацеливания бизнеса на получение прибыльных результатов, получения конкурентного преимущества перед аналогами, не использующими инновационные подходы для соответствующей защиты своих информационных технологических систем в будущем.

Список литературы

1. Pooja M., Uma P. Insights of JSON Web Token. 2020.
2. Зимина К.И. и Лапонина О.Р. Механизмы межсервисной аутентификации в приложениях с микросервисной архитектурой. 2023.
3. А.В. Беловодов, О.Р. Лапонина Использование управления доступом на основе атрибутов в протоколе OAuth 2.0. 2023.
4. Balaji V. Andres S. Advanced Spring LDAP. 2023.
5. Danso S. D., Yin C. API Security: Protecting APIs With Keycloak. 2023.
6. Krishna S. JSON Web Token (JWT) based client authentication in Message Queuing Telemetry Transport (MQTT). 2019.
7. Ж. Стоянов, И. Христоский Направления будущих исследований и рекомендации по развитию микросервисной архитектуры. 2024.
8. Ł. Wyciślik, Ł. Latusik, A. M. Kamińska A comparative assessment of jvm frameworks to develop microservices. 2023.
9. R. Hat Keycloak-open source identity and access management. 2021.
10. A. Bucko, K. Vishi, B. Krasniqi, B. Rexha, Enhancing JWT Authentication and Authorization in Web Applications. 2023.
11. A. Chatterjee, A. Prinz Applying Spring Security Framework with Keycloak-based OAuth2. 2022.
12. D. Hardt The OAuth 2.0 Authorization Framework. 2012.
13. M. Rouse LDAP (Lightweight Directory Access Protocol). 2019.
14. M. G. de Almeida, E. D. Canedo Authentication and Authorization in Microservices Architecture: A Systematic Literature Review. 2022.
15. T. Sylla, L. Mendiboure, M. A. Chalouf, F. Krief Blockchain-based Context-Aware Authorization Management as a Service in IoT. 2021.
16. A. Hoffman Web Application Security: Exploitation and Countermeasures for Modern Web Applications. 2020.
17. S.Thorgersen, P. I. Silva Keycloak - Identity and Access Management for Modern Applications: Harness the power of Keycloak, OpenID Connect, and OAuth 2.0 protocols to secure applications. 2021.

References

1. Pooja M., Uma P. Insights of JSON Web Token. 2020.
2. K.I. Zimina, O.R. Laponina Cross-Service Authentication Mechanisms in Applications with Microservice Architecture. 2023.
3. A.V. Belovodov, O.R. Laponina Using attribute-based access control in OAuth 2.0. 2023.

4. Balaji V. Andres S. Advanced Spring LDAP. 2023.
5. Danso S. D., Yin C. API Security: Protecting APIs With Keycloak. 2023.
6. Krishna S. JSON Web Token (JWT) based client authentication in Message Queuing Telemetry Transport (MQTT). 2019.
7. Z. Stojanov, I. Hristoski Research Trends and Recommendations for Future Microservices Research. 2024.
8. Ł. Wyciślik, Ł. Latusik, A. M. Kamińska A comparative assessment of jvm frameworks to develop microservices. 2023.
9. R. Hat Keycloak-open source identity and access management. 2021.
10. A. Bucko, K. Vishi, B. Krasniqi, B. Rexha, Enhancing JWT Authentication and Authorization in Web Applications. 2023.
11. A. Chatterjee, A. Prinz Applying Spring Security Framework with Keycloak-based OAuth2. 2022.
12. D. Hardt The OAuth 2.0 Authorization Framework. 2012.
13. M. Rouse LDAP (Lightweight Directory Access Protocol). 2019.
14. M. G. de Almeida, E. D. Canedo Authentication and Authorization in Microservices Architecture: A Systematic Literature Review. 2022.
15. T. Sylla, L. Mendiboure, M. A. Chalouf, F. Krief Blockchain-based Context-Aware Authorization Management as a Service in IoT. 2021.
16. A. Hoffman Web Application Security: Exploitation and Countermeasures for Modern Web Applications. 2020.
17. S.Thorgersen, P.I.Silva Keycloak - Identity and Access Management for Modern Applications: Harness the power of Keycloak, OpenID Connect, and OAuth 2.0 protocols to secure applications. 2021.