

Архитектурный метод ротации ключей авторизации по токену с временной поддержкой старого ключа

Вихляев Дмитрий Романович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье описывается связанная компонентная модель системы позволяющая реализовать периодическую смену ключа шифрования токена авторизации. В реализации учувствуют компоненты системы, организующие связь и синхронизацию настроек по времени. Результатом работы станет описание модулей и их процессов, реализующих временную поддержку истёкшего ключа шифрования.

Ключевые слова: авторизация, токен, ключ шифрования, ротация ключей.

An architectural method for rotating authorization keys by token with temporary support for the old key

Vikhlyayev Dmitry Romanovich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article describes the internal structure of images and containers in Docker, as well as a method for creating your own images. The process of creating images is carried out using the Dockerfile instruction file and the Docker image mount tool. The result of the work will be a description of the process of creating instructions for developing a new image using examples of Python and Nodejs applications.

Keywords: Docker, image, container, containerization.

1 Введение

1.1 Актуальность

Ротация ключа токена авторизации с временной поддержкой старого ключа представляет собой важный метод обеспечения безопасности систем аутентификации и авторизации. Актуальность данного подхода обусловлена необходимостью защиты от угроз, связанных с компрометацией ключей, а также обеспечением устойчивости системы к потенциальным атакам. В условиях растущего числа распределенных систем, взаимодействующих через API, и повышенных требований к защите данных ротация ключей позволяет минимизировать риски несанкционированного доступа и утраты конфиденциальной информации. Особенность ротации ключей заключается в том, что она предполагает периодическую замену ключей, используемых для

подписи и проверки токенов. Этот процесс снижает вероятность успешного использования скомпрометированных ключей злоумышленниками, так как срок их действия ограничен. Однако полная замена ключей может нарушить работу системы, если токены, подписанные устаревшим ключом, утрачивают свою действительность до завершения всех активных сеансов. Для устранения этого недостатка применяется временная поддержка старого ключа, обеспечивающая плавный переход между версиями ключей.

1.2 Обзор исследований

А.Д.Жестков провёл анализ угроз и уязвимостей при хранении закрытого ключа [1]. К.А.Бирюков сравнил средства безопасного хранения ключей [2]. С.С.Агафьин представил функции безопасности внешних аппаратных модулей, используемых для хранения криптографических ключей [3]. К.Ю.Филатов описал использование токенов на примере JWT в spring boot REST api приложении [4]. С.В.Козлов, М.С.Воробьев создал аутентификацию пользователя в web-приложении с использованием стандарта JWT [5].

1.3 Цель исследования

Цель исследования – описать способ периодической ротации ключей шифрования с временной поддержкой старого ключа, во много сервисной системе с поддержкой авторизации.

2 Материалы и методы

В работе приведены механизмы работы брокера сообщений и сервиса синхронизации настроек, приводится описание хранения ключей в таблице базы данных.

3 Результаты и обсуждения

Периодическая замена секретной фразы (ключ) шифрования представляет технологию управления безопасностью, которая функционирует на основе динамически обновляемых ключей и токенов. В описании представлен JWT-токен, но в современных системах ротацию применяют к Refresh-токену, который может быть реализован на основе JWT. Логика направлена на обеспечение безопасного обмена данными путем использования криптографических методов для подписания и верификации данных, а также ротации ключей. Основными компонентами системы являются механизмы работы с токенами, ключами и их синхронизация, которые интегрируются для обеспечения надежности и функциональности. Секретные фразы необходимы как для симметричного так для асимметричного метода шифрования.

Основной компонент управления токенами предоставляет интерфейсы для генерации токенов с заданными данными и их верификации. Токен включает в себя полезную нагрузку и временные параметры, такие как время выпуска и срок действия. Функции подписи и проверки токена зависят от криптографических ключей. Верификация токена требует анализа времени

выпуска и срока действия, что предотвращает использование устаревших или поддельных токенов. Ключи играют центральную роль в технологии. Каждый ключ имеет период действия, в течение которого он используется для подписи и верификации токенов. Система поддерживает хранение ключей и предоставляет методы для их создания, удаления и поиска по времени действия. Это позволяет точно определить, какой ключ использовался при генерации токена, и верифицировать его подлинность.

Хранилище ключей является контейнером для управления ключами. Оно обеспечивает поиск ключей по временным меткам, позволяя определять действующий ключ в конкретный момент времени. Также реализован механизм ротации ключей, который автоматически заменяет устаревшие ключи новыми, минимизируя риски компрометации.

Управляющий компонент синхронизации времени отвечает за согласование сроков действия токенов и периодов ротации ключей. Это необходимо для предотвращения сбоев, связанных с несовпадением временных интервалов между компонентами системы. Он также координирует обновления конфигурации между модулями токенов и ключей.

Система управления токенами взаимодействует с хранилищем ключей для получения ключей, необходимых для выполнения операций подписи и верификации. Управляющий компонент ключей отвечает за добавление новых ключей в хранилище и удаление устаревших. Это взаимодействие обеспечивает актуальность и безопасность системы.

В реализации каждого элемента системы возможны специфические сложности. Для токена необходима защита полезной нагрузки и предотвращения подделки токенов. Это требует эффективной реализации криптографических алгоритмов. Генерация криптографически стойких ключей и обеспечение их безопасного хранения. Хранилище ключей отвечает за эффективную организацию поиска ключей по временным интервалам при минимальных задержках. Ротация ключей узкое место, где нужно управлять сроками действия ключей с учетом времени их использования в токенах. Синхронизация времени должна поддерживать точность времени между компонентами, особенно в распределенных системах (рис.1).

В процессе взаимодействия компонентов системы, ключевым элементом является сервис синхронизации, который инициирует обновление настроек токена и отправку соответствующего сообщения через брокера сообщений. Брокер сообщений реализует внутри себя очередь сообщений, каждое сообщение отправляется всем зарегистрированным подписчикам по протоколу TCP. Менеджеры токенов и ключей прослушивают канал и в случае поступления обновления обновляют внутренние данные. За гарантию доставки отвечает брокер сообщений. Этот процесс гарантирует корректную работу системы, и актуальность параметров токенов в реальном времени (рис.2).

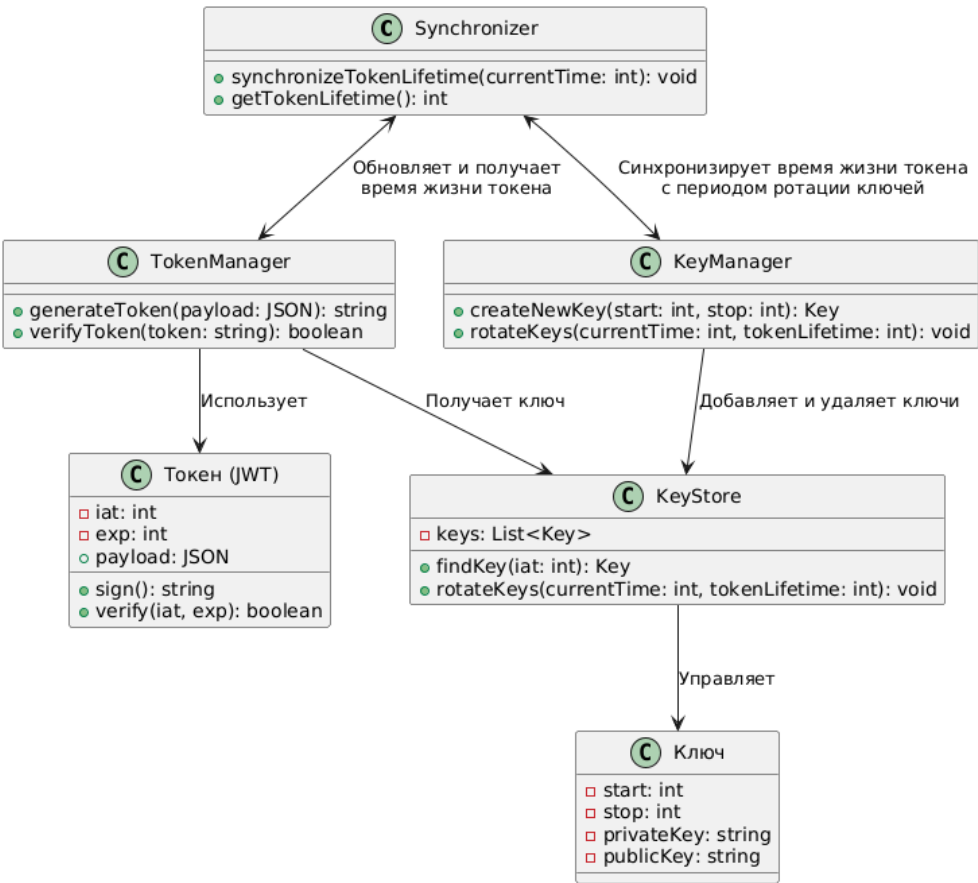


Рис. 1. Связь компонентов системы

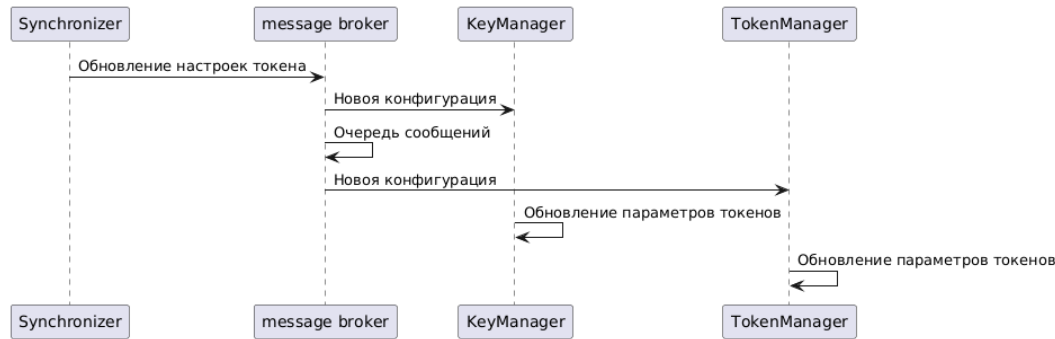


Рис. 2. Передача обновлений сервисам через брокера сообщений

Для работы с ротацией ключей и проверки токенов в зависимости от времени их жизни, используется метод, который извлекает ключ из базы данных на основе времени выпуска токена (iat) и времени истечения действия токена (exp). В системе используется таблица базы данных keys, в которой хранятся данные о ключах, включая их время создания (start), время окончания действия (stop), а также активность ключа (is_active). Вычисляется переходный период как разница между временем истечения токена (exp) и временем его выпуска (iat). Переходный период используется для динамической проверки, чтобы убедиться, что ключ еще действителен в пределах этого периода. Запрос выбирает ключи, которые активны на момент запроса, с учетом переходного периода, и сортирует их по времени создания в убывающем порядке, чтобы сначала получить наиболее актуальные ключи.

Метод возвращает первый найденный ключ, если он соответствует условиям (рис.3,4).

```
CREATE TABLE keys (  
  id SERIAL PRIMARY KEY,  
  key TEXT NOT NULL,  
  is_active BOOLEAN DEFAULT FALSE,  
  stop TIMESTAMP NOT NULL, -- Время, до которого ключ активен  
  start TIMESTAMP DEFAULT CURRENT_TIMESTAMP -- Время создания ключа  
);
```

Рис. 3. Структура таблицы для хранения ключей в базе данных

```
async function getKeyForToken(iat, exp) {  
  // Переходный период = exp - iat  
  const transitionPeriod = exp - iat;  
  
  const query = `SELECT key FROM keys  
    WHERE start <= TO_TIMESTAMP($1) AND stop + ${transitionPeriod} >= TO_TIMESTAMP($1)  
    ORDER BY start DESC LIMIT 1  
  `;  
  const res = await pool.query(query, [iat]);  
  return res.rows.length ? res.rows[0] : null; // Возвращаем ключ, если найден  
}
```

Рис. 4. Поиск ключа соответствующего токену по времени

Ключи хранятся с полями start и stop время выпуска, время окончания активности, старый ключ должен поддерживаться, пока не истечёт время жизни токена (stop+exp-iat) потом удаляется. Менеджер ключей вместе с генерацией нового ключа устанавливает событие на его грядущее удаление. Генерация ключа происходит через установленные промежутки времени, которые равны (stop-start).

4 Выводы

Таким образом, был описан подход для применения ротации ключей с временной поддержкой старого. Приведены конкретные способы гарантии установки равных значений конфигурации в один момент времени. Описаны компоненты возможной системы и их взаимодействие.

Библиографический список

1. Жестков А.Д. Анализ угроз и уязвимостей при хранении закрытого ключа // Научный аспект. 2024. Т. 17. № 5. С. 2287-2293.
2. Бирюков К.А. Средства безопасного хранения ключей // Безопасность информационных технологий. 2013. Т. 20. № 3. С. 50-53.
3. Агафьин С.С. Функции безопасности внешних аппаратных модулей, используемых для хранения криптографических ключей // Безопасность информационных технологий. 2014. Т. 21. № 3. С. 37-43.
4. Филатов К.Ю. Использование токенов на примере JWT в spring boot REST

- арі приложенні // Інформаційні технології моделювання і управління. 2022. Т. 128. № 2. С. 153-157.
5. Козлов С.В., Вороб'єв М.С. Реалізація аутентифікації користувача в web-приложенні з використанням стандарту JWT // Системи комп'ютерної математики і їх застосування. 2022. № 23. С. 362-366.
 6. Девіщина С.Н., Пилькевич П.В., Удод Е.В. Способи покращення захищеності сервісів, використовують JWT-токени // Економіка. Інформатика. 2023. Т. 50. № 1. С. 144-151.
 7. Бабичева М.В. Атаки на jwt токени // В збірнику: Донецькі читання 2022: освіта, наука, інновації, культура і виклики сучасності. Матеріали VII Міжнародної наукової конференції, присвяченої 85-літтю Донецького національного університету. Під загальною редакцією С.В. Беспалової. Донецьк, 2022. С. 176-178.