УДК 629.3.025.7

ІНСТРУМЕНТАЛЬНІ ЗАСОБИ РОЗРОБКИ СИСТЕМИ АВТОРИЗАЦІЇ НА ОСНОВІ СТАНДАРТУ OAUTH 2.0

Дяк А.М., студент
Тарнавський Ю.А., к.ф.м.н.
Національний технічний університет України
«Київський політехнічний інститут ім. І. Сікорського»(Україна, м. Київ)

Анотація *– в статті розглянуто основи авторизації через стандарт OAauth 2.0, способи інтеграпроєкту зі стандартом та аспекти вибору правильного способу інтеграції враховуючи конкретні вимоги.*

Ключові слова *– авторизаційний сервіс, ресурсний сервер, код, авторизація, токен, додаток, доступи.*

Постановка проблеми. Створення авторизаційної системи на основі OAuth 2.0 вимагає хороших інструментальних засобів для ефективної розробки. Проте, може виникнути складність у виборі інструментів, так як наявність широкого вибору варіантів може призвести неправильного або недоцільного використання.

Тому, основна проблема полягає у правильному аналізі інструментів для розробки сервісу так як вони можуть варіюватись від налаштування вже наявних рішень до повноцінної (з нуля) розробки, що включає програмування, використання фреймворків, баз даних та інших інструментів.

Аналіз останніх досліджень. Авторизація користувачів, в веб-середовищі на основі протоколу OAuth 2.0, вивчалась на великій кількості наукових та практичних дослідження. Дані вивчення були проведені групою відкритого міжнародного співтовариства, учених, проєктувальників, мережевих операторів і провайдерів (IETF). [1]

Найбільша кількість досліджень спрямовуються як і на аспекти безпеки протоколу OAuth 2.0 (Ranieri et al., 2022) так і на правильне використання стандарту в певних умовах таких як IoT (Chen et al., 2023) або Cloud Computing (Dong et al., 2023).

Також ще присутні окремі вивчення, що фокусуються на покращенні OAuth 2.0, шляхом додаткових розширень стандарту. Однак, дослідження, що пов’язані з інструментальними засобами розробки систем на основі OAuth 2.0 зустрічаються, відносно досить рідко.

Формулювання цілей (Постановка завдання). Розглянути основні принципи роботи стандарту OAuth 2.0. Проаналізувати вже

наявні рішення на основі OAuth 2.0. Визначити головні аспекти для розробки власного авторизацйного сервісу. Провести пошук оптимального рішення в залежності від потреб.

Основна частина. OAuth 2.0 - це протокол, який дозволяє користувачеві надавати сторонньому веб-сайту або додатку доступ до захищених ресурсів користувача, не обов'язково розкриваючи свої довгострокові облікові дані або навіть свою особу. [2]

Грант авторизації - це обліковий запис, що представляє ресурс власника ресурсу (на доступ до його захищених ресурсів), який використовується клієнтом для отримання токену доступу. Виділяють основні чотири типи грантів в OAuth 2.0 – авторизаційний код, неявний, пароль власника ресурсу облікові дані власника ресурсу та облікові дані клієнта. [3]

Авторизаційний код - це найпоширеніший тип гранту в OAuth 2.0. У цьому гранті клієнт перенаправляє користувача на сервер авторизації, який автентифікує користувача і отримує його згоду на надання доступу клієнту. Потім сервер авторизації надсилає клієнту код авторизації, який клієнт обмінює на токен доступу. Даний потік використовується у веб-додатках або мобільних додатках. [4, с. 236]

Неявний – це спрощена версія потоку, що використовує авторизаційний код. Різниця полягає в тому, що у відповідь повертається відразу токен доступу, без коду авторизації. Здебільшого використовується додатками, які працюють у середовищах, де складно зберігати інформацію безпечно (веб-браузер).

Облікові дані користувача – це спосіб авторизації, який вимагає у користувача передачу свої облікових даних додатку. В такому випадку, використовується додаток до якого є повна довіра. Перевага такого способу авторизації полягає в тому, що він може бути використаний, коли перенаправлення на авторизацйний сервіс є неможливим.

Облікові дані клієнта - це потік, що передбачає обмін облікових даних програми, таких як ідентифікатор клієнта та секрет клієнта, на токен доступу. Використовується для неінтерактивних додатків, наприклад, автоматизованих процесів, мікросервісів тощо.

Для інтеграції проєкту зі стандартом OAuth 2.0 існують три основні способи: використання Oauth2.0 бібліотеки, імплементація OAuth 2.0 вручну та використання вже готових рішень IAM (Identity and Access Management) від певного постачальника. [4, с. 237]

Інтеграція проєктного коду з OAuth 2.0 буде найпростішою, якщо використовувати спеціально розроблені для цього бібліотеки. Практично всі найпопулярніші на ринку мови програмування та фреймворки мають вже створені бібліотеки для підтримки процесу авторизації та автентифікації користувачів. Ці бібліотеки мають набір готових функцій, класів та компонентів, які допомагають простіше проводити авторизацію, використовуючи різні гранти, та отримувати

доступи. Прикладами таких бібліотек є DotNetAuth (.NET), Fosite (GO), Authlib (Python), O2 (QT/C++), Spring Security Oauth (Java), Oauth2 Ruby Gem (Ruby) та інші.

Імплементація OAuth 2.0 вручну в проєкт вимагає хорошого розуміння роботи самого протоколу, вміння захистити авторизаційний сервіс від несанкціонованого доступу (включаючи використання HTTPS), налагодження взаємодії з сервісом використовуючи різні гранти авторизації і захищене збереження токенів та доступів базі. Хоча даний процес вимагає досить багато часу та неабияких зусиль, ми отримаємо дуже гнучке рішення, яке буде можливо налаштувати для своїх конкретних потреб.

Використання вже готових рішень дозволяє значно зменшити час та зусилля, що потрібні для самостійної розробки авторизаційного сервісу, так як вони надають готовий функціонал, забезпечують безпеку сервісу та мають можливість масштабування. [5] Також дані рішення не вимагають глубокого розуміння стандарту та всіх його специфікацій. Найбільш популярними рішеннями на ринку є Okta, Keycloak та Microsoft Azure Active Directory. Розглянемо кожне з них.

Okta є хмарним рішенням для управління ідентифікацією та доступом, яке забезпечує чудовий користувацький досвід і просте адміністрування, безпечно об'єднуючи постачальників, партнерів і клієнтів підприємства в єдиний контур. Серед переваг Окти можна виділити легкість використання, можливість інтеграції з різними додатками та платформами, масштабованість, підтримка стандартів (OAuth 2.0, OIDC, SAML) і хороша документація. Також Окта має недоліки, такі як вартість, тривале навчання та специфікації інтеграції.

Keycloak - це рішення для управління ідентифікацією та доступом з відкритим кодом, яке надає користувачам сучасні додатки та сервіси. Інструмент дозволяє легко захистити сервіси та додатки з мінімальною кількістю коду або взагалі без нього. Серед переваг даного рішення можна виділити відкритий код, гнучкість, наявність багатофакторної авторизації, вбудовані адміністративні інструменти та підтримка стандартів (таких як OAuth 2.0, OIDC, SAML). Недоліками Keycloak є складність налаштування, відсутність більш детального опису всього функціоналу в документації, необхідність самостійного розгортання системи (сервісу і бази даних), ресурсоємність, витрачення великої кількості часу на інтеграцію.

Azure Active Directory - це багатокористувацька хмарна служба каталогів і керування ідентичностями від Microsoft. В Azure AD є ряд переваг, на які варто звернути увагу, такі як легка інтеграція з різними сервісами від Microsoft, масштабованість, надійність, гнучкість, наявність багатофакторної авторизації та можливість керування доступами до ресурсів. Є також і ряд недоліків такі як порівняно велика вартість для великих та середніх організацій, складність налаштування,

складна документація, важкість інтегрування не з продуктами від Microsoft та зав’язаність на цих продуктах.

Висновки. В даній статті було розглянути 3 рішення для створення можливості авторизації через OAuth 2.0.

Використання бібліотек OAuth 2.0 є швидким та зручним способом реалізації протоколу OAuth 2.0. Готові бібліотеки надають готовий функціонал та інструменти, які спрощують розробку авторизаційного сервісу.

Імплементація OAuth 2.0 вручну вимагає більшої розуміння протоколу та ручного написання коду. Цей підхід може бути корисним для розробників, які бажають більшої гнучкості та контролю над реалізацією авторизаційного сервісу.

Готові рішення IAM (Okta, Keycloak, Azure AD) надають повний функціонал для управління ідентифікацією та доступом, включаючи протокол OAuth 2.0. Вони забезпечують високу безпеку, готові інструменти для розробників та можливість інтеграції з іншими сервісами.

Вибір конкретного підходу залежить від конкретних потреб, ресурсів, доступних для розробки, вимог до безпеки та гнучкості. Користування готовими рішеннями IAM може спростити розробку та забезпечити високу безпеку, але вимагає відповідних витрат. Водночас, реалізація вручну дозволяє більшу гнучкість, але вимагає більшої роботи та експертизи у протоколі OAuth 2.0.

Бібліографічний список

1. Учасники проєктів Вікімедіа. IETF – Вікіпедія. Вікіпедія. URL: https://uk.wikipedia.org/wiki/IETF (дата звернення: 28.05.2023).

2. OAuth 2.0 Authorization Framework. Auth0 Docs. URL: https://auth0.com/docs/authenticate/protocols/oauth (дата звернення: 28.05.2023).

3. RFC 6749: The OAuth 2.0 authorization framework. URL: https://www.rfc-editor.org/rfc/rfc6749 (дата звернення: 28.05.2023).

4. Дяк А.М. Інструментальні засоби розробки системи авторизації на основі стандарту OAuth 2.0 / А.М. Дяк, Ю.А. Тарнавський // Сучасні проблеми наукового забезпечення енергетики: Матеріали XVIII Міжнародної науково-практичної конференції молодих вчених і студентів 2023 року. — К.: КПІ ім. Ігоря Сікорського, 2023. — Т. 2.— С. 236-237.

5. Gittlen S., Rosencrance L. What Is Identity and Access Management? Guide to IAM. Security. URL: https://www.techtarget.com/searchsecurity/definition/identity-access-management-IAM-system (дата звернення: 28.05.2023).

УДК 629.3.025.7