OAuth та OpenID connect

План

- Призначення OAuth
- Ролі в OAuth
- Надання авторизації
- Типи клієнтів та їх профілі
- OpenID Connect
- Схема взаємодії в OpenID Connect

Аутентифікації клієнта в HTTP

- BASIC автентифікація
- DIGEST автентифікація

Проблеми та обмеження

- Стороннім програмам потрібно зберігати облікові дані власника ресурсу для подальшого використання, зазвичай це пароль у явному вигляді.
- Серверам потрібна підтримка автентифікації за паролем, незважаючи на їх недоліки.
- Сторонні програми отримують надмірно великий доступ до захищених ресурсів власника, не дозволяючи власнику обмежити доступ.
- Власник ресурсу не може скасувати доступ до окремої сторонньої програми, не скасувавши доступ для всіх сторонніх учасників.
- Компрометація будь-якої сторонньої програми призводить до компрометації пароля кінцевого користувача та всіх даних, що захищаються цим паролем.

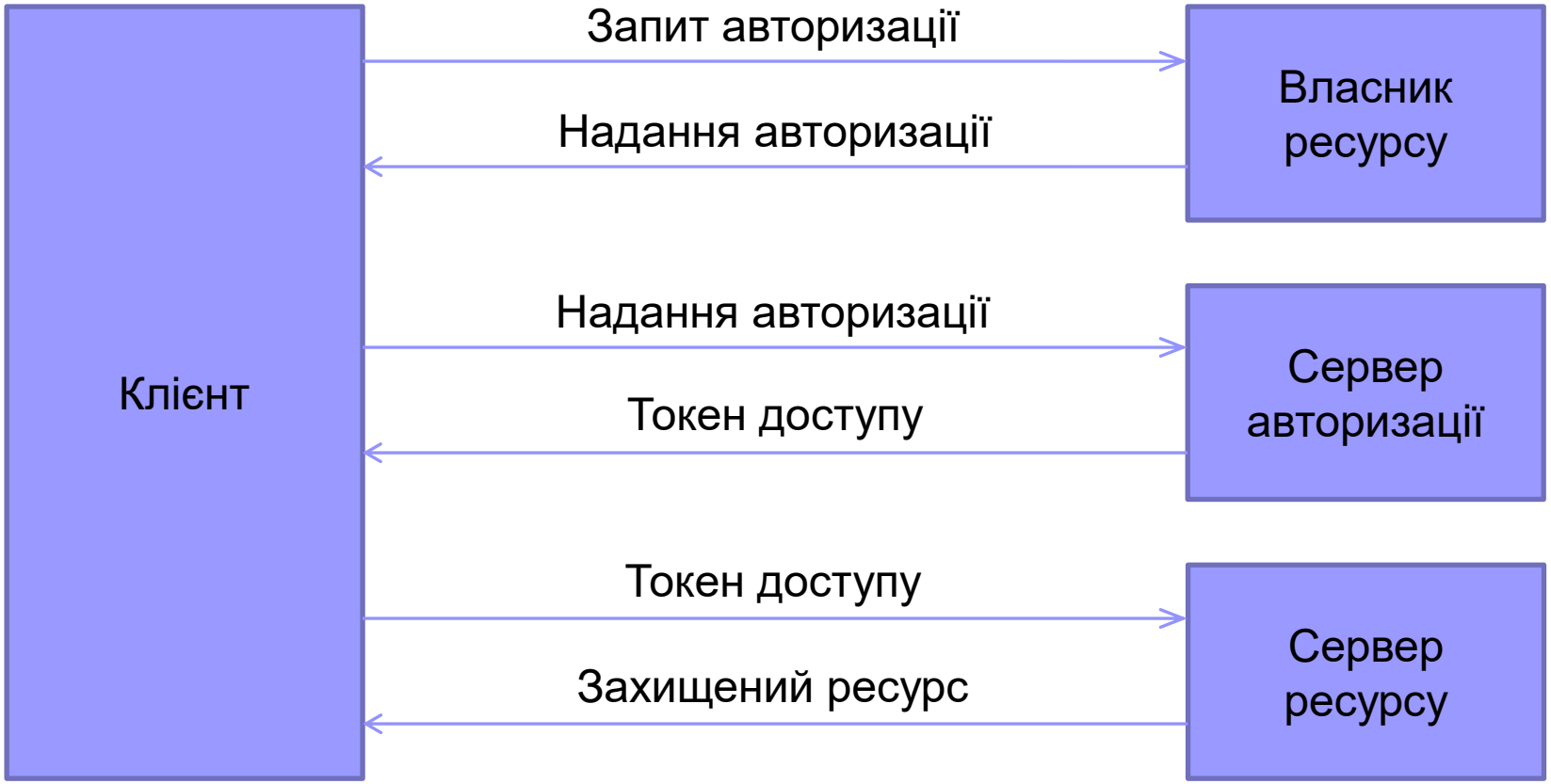
Призначення OAuth

- OAuth являє собою фреймворк для авторизації, що дозволяє програмам здійснювати обмежений доступ до облікових записів користувачів на HTTP сервісах, наприклад, на Facebook, GitHub. Він працює за принципом делегування автентифікації користувача сервісу, на якому знаходиться обліковий запис користувача, дозволяючи сторонньому додатку отримувати доступ до облікового запису користувача.
- Визначає рівень авторизації, розділяє ролі клієнта та власника ресурсу
- В OAuth запити клієнта на доступ до ресурсу управляються власником ресурсу та обробляються сервером ресурсу, і для клієнта випускається набір облікових даних, відмінний від облікових даних власника ресурсу.

Ролі в OAuth

- Власник ресурсу
  - Сутність, яка може надати доступ до захищеного ресурсу. Якщо власник ресурсу є людиною, він називається кінцевим користувачем.
- Сервер ресурсу
  - Сервер, на якому розміщені ресурси, що захищаються, здатний приймати запити до ресурсів і відповідати на них, використовуючи токени доступу.
- Клієнт
  - Додаток, що виконує запити до ресурсу, що захищається з отриманим від власника ресурсу дозволом на доступ.
- Сервер авторизації
  - Сервер, що випускає токени доступу для клієнта після успішної автентифікації клієнта власником ресурсу та отримання авторизації.

Схема взаємодії в протоколі



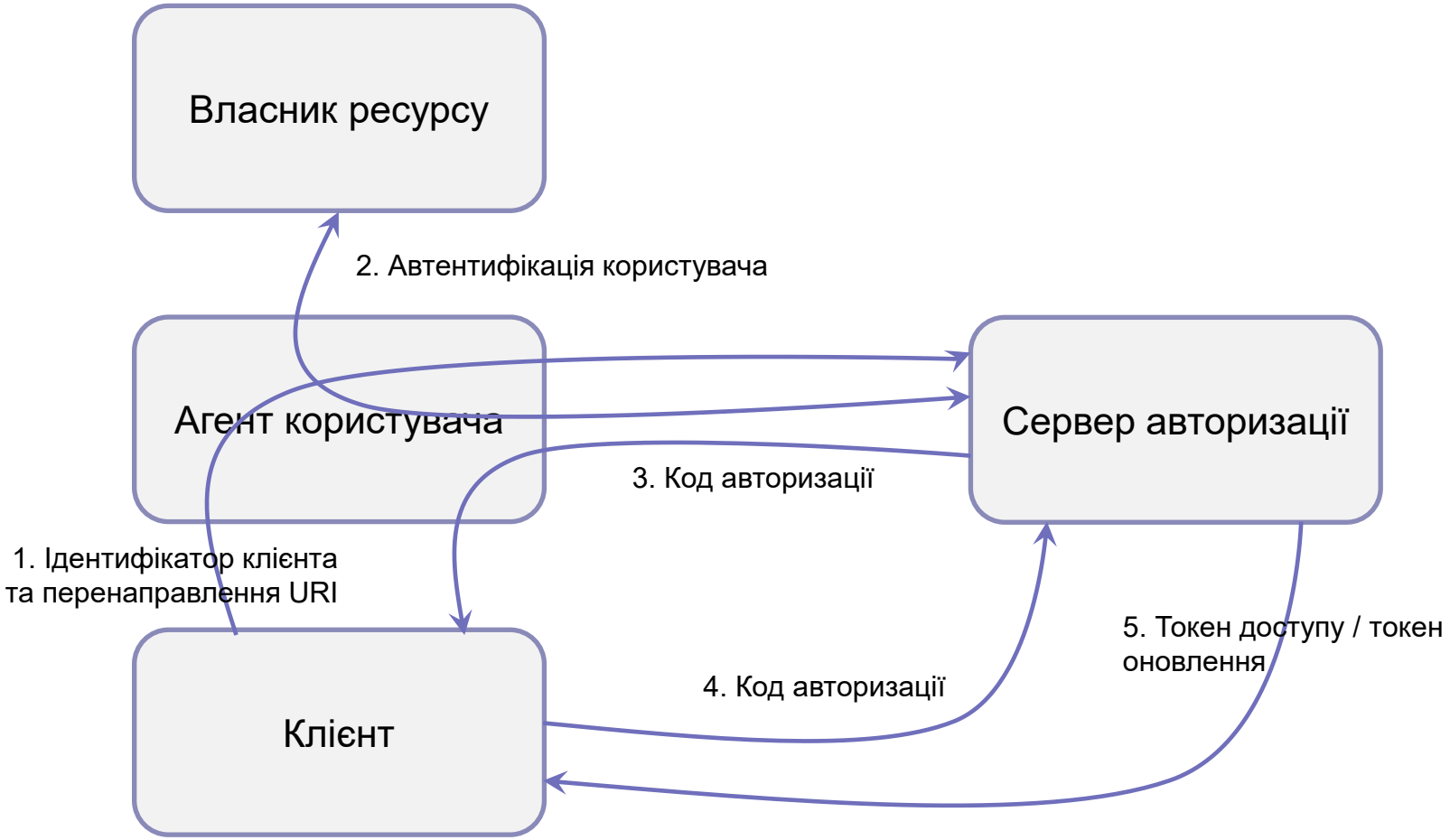
Надання авторизації

- Надання авторизації є обліковими даними, які видаються власником ресурсу для доступу до захищених ресурсів, і використовується клієнтом для отримання токена доступу. Специфікація OAuth визначає чотири основних типи дозволів:
  - код авторизації,
  - неявний дозвіл,
  - облікові дані пароля власника ресурсу,
  - облікові дані клієнта

Код авторизації

- Код авторизації отримують з використанням сервера авторизації як посередника між клієнтом і власником ресурсу. Замість того щоб запитувати авторизацію безпосередньо у власника ресурсу, клієнт перенаправляє власника ресурсу до сервера авторизації (використовуючи його агент користувача, відповідно до RFC 2616), звідки в свою чергу виконується перенаправлення власника ресурсу назад до клієнта з кодом авторизації.
- Перед перенаправленням власника ресурсу назад до клієнта з кодом авторизації сервер авторизації автентифікує власника ресурсу та виконує авторизацію. Оскільки власник ресурсу автентифікується лише на сервері авторизації, облікові дані власника ресурсу ніколи не передаються клієнту.

Отримання коду авторизації



Неявний дозвіл

- Неявний дозвіл є найпростішим пересиланням коду авторизації, оптимізованим для клієнтів, реалізованих у браузері, які використовують скриптову мову, таку як JavaScript. У разі неявного пересилання замість коду авторизації клієнта безпосередньо клієнту випускається токен доступу (як результат авторизації власника ресурсу). Тип дозволу є неявним, так як не випускається проміжних облікових даних (таких як код авторизації), які в подальшому використовуються для отримання токена доступу.
- При випуску токена доступу при використанні неявного дозволу сервер авторизації не автентифікує клієнта. У деяких випадках ідентифікація клієнта може бути перевірена за допомогою перенаправлення URI, що використовується для доставки токена доступу клієнту. Токен доступу може бути доступний власнику ресурсу або іншим програмам, які мають доступ до користувача агенту власника ресурсу.

Схема роботи неявного дозволу

Облікові дані пароля власника ресурсу

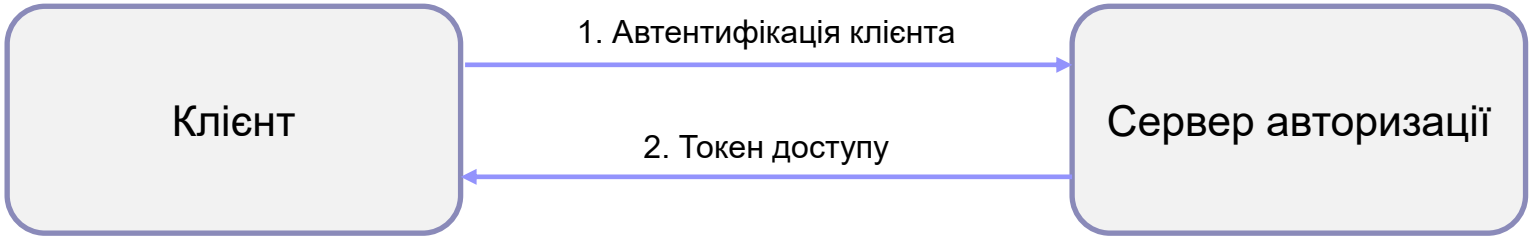
- Облікові дані на основі пароля власника ресурсу (наприклад, ім'я користувача та пароль) можуть бути безпосередньо використані як код авторизації для отримання токена доступу. Облікові дані повинні використовуватися лише тоді, коли є високий рівень довіри між власником ресурсу та клієнтом (наприклад, клієнт є частиною ОС або високо привілейованої програми), і коли інші типи дозволів авторизації не доступні (такі як код авторизації).
- Хоча цей тип дозволу вимагає прямого доступу клієнта до облікових даних власника ресурсу, облікові дані власника ресурсу використовуються для єдиного запиту та обмінюються на токен доступу. Даний тип доступу дозволяє ліквідувати необхідність клієнта зберігати облікові дані власника ресурсу для подальшого використання, обмінюючи облікові дані на токени доступу з великим терміном життя або на токени оновлення.

Облікові дані пароля власника ресурсу

Облікові дані клієнта

- Облікові дані клієнта (або інші форми автентифікації клієнта) можуть використовуватися як авторизаційний дозвіл, коли область авторизації обмежена ресурсами, що керуються клієнтом, або ресурсами, що попередньо узгоджуються з сервером авторизації.
- Зазвичай облікові дані клієнта використовуються як авторизаційний дозвіл, коли клієнт діє від імені власника (клієнт також є власником ресурсу) або запитує доступ до захищених ресурсів, ґрунтуючись на попередньо отриманій авторизації від сервера авторизації.

Облікові дані клієнта



Типи клієнтів

- Конфіденційний
  - Клієнти мають можливість зберегти конфіденційність своїх облікових даних (наприклад, клієнт, реалізований на захищеному сервері з обмеженим доступом до облікових даних клієнтів), або мати можливість виконувати безпечну автентифікацію клієнтів за допомогою інших способів.
- Публічний
  - Клієнти, які не можуть підтримувати конфіденційність своїх облікових даних (наприклад, клієнти, що працюють на пристрої, що використовуються власником ресурсу, наприклад, встановлений додаток або програму на основі веб-браузера), та не здатні виконувати безпечну аутентифікацію клієнта.

Профілі клієнтів

- Веб-застосунок
  - Веб-застосунок - це конфіденційний клієнт, який працює на веб-сервері. Власники ресурсу отримують доступ до клієнта через інтерфейс HTML, оброблений у користувацькому агенті на пристрої, який використовується власником ресурсу. Облікові дані клієнта, а також будь-який доступ до токену, випущений для клієнта, зберігаються на веб-сервері.
- Застосування на основі агента користувача
  - Застосунок на основі агента користувача є публічним клієнтом, в якому клієнтський код завантажується з веб-сервера та працює у користувацькому агенті (наприклад, веб-браузер) на пристрої, який використовується власником ресурсу. Облікові дані, передані через протокол, є легко доступними (і часто видимими) власником ресурсу.
- Застосунок
  - Застосунок - це публічний клієнт, який встановлений на пристрої, який використовується власником ресурсу. Облікові дані, передані протоколом, доступні власнику ресурсу.

OpenID Connect

- Забезпечує ідентифікацію та автентифікацію поверх OAuth 2.0
- Дозволяє довіряючим сторонам (RPs) підтвердити особу кінцевого користувача
- Дозволяє RP отримувати основну інформацію про профіль
- Описано на https://openid.net/connect/

Схема взаємодії в OpenID Connect