УДК 004.42:004.6
DOI https://doi.org/10.32782/IT/2024-4-30

Андрій ЯРМОЛАТІЙ
викладач кафедри кібербезпеки, інформаційних технологій та економіки, Київський університет інтелектуальної власності та права Національного університету «Одеська юридична академія», Харківське шосе 210, м. Київ, Україна, 02121
***ORCID:** 0009-0004-8655-9928*

Любов ЧЕРЕМІСІНА
викладач кафедри кібербезпеки, інформаційних технологій та економіки, Київський університет інтелектуальної власності та права Національного університету «Одеська юридична академія», Харківське шосе 210, м. Київ, Україна, 02121
***ORCID:** 0009-0005-0719-0745*

Валентин ГАЛУНЬКО
доктор філософії з галузі права, доцент кафедри адміністративного права, інтелектуальної власності та цивільно-правових дисциплін, Київський університет інтелектуальної власності та права Національного університету «Одеська юридична академія», Харківське шосе, 210, Київ, Україна, 02121
***ORCID:** 0000-0002-8133-6766*

Бібліографічний опис статті: Ярмолатій, А., Черемісіна, Л., Галунько, В. (2024). Побудова веб-додатків із сучасними системами аутентифікації на базі OAuth 2.0 PKCE та біометрики для підвищення кіберстійкості цих додатків. *Information Technology: Computer Science, Software Engineering and Cyber Security,* 4, 242–251, doi: https://doi.org/10.32782/IT/2024-4-30

ПОБУДОВА ВЕБ-ДОДАТКІВ ІЗ СУЧАСНИМИ СИСТЕМАМИ АУТЕНТИФІКАЦІЇ НА БАЗІ OAUTH 2.0 PKCE ТА БІОМЕТРИКИ ДЛЯ ПІДВИЩЕННЯ КІБЕРСТІЙКОСТІ ЦИХ ДОДАТКІВ

Сучасний світ бізнесу в своїй технічній основі базується на множинній інтеграцій бізнес платформ різного призначення, зокрема веб-додатків та веб-платформ, що поєднує не лише передачу та синхронізацію даних, але і синхронізацію бізнес процесів і процесів прийняття рішень, які цими даними управляються. Відповідно цілісність і кіберзахист таких даних є ключовим аспектом убезпечення від ризиків і створення підґрунтя для прийняття правильних рішень. В умовах стрімкого розвитку цифрових технологій та значного зростання кіберзагроз, питання забезпечення безпеки веб-додатків бізнесу набуває особливої актуальності а подекуди є важливою частиною корпоративної стратегії при виборі або створенні нових веб-систем. Одним із основних інструментів забезпечення безпеки інтеграцій між веб-платформами Протокол OAuth 2.0 з механізмом Proof Key for Code Exchange (PKCE) є одним з основних інструментів для захисту таких інтеграцій.

***Мета роботи.** Метою даної роботи є аналітичний розгляд технічних аспектів практичного впровадження протоколу OAuth 2.0 і створення пропозиції щодо його вдосконалення через застосування біометричних технологій алгоритму Proof Key for Code Exchange (PKCE) що особливо важливо розуміти під час створення веб-додатків (веб-програмування). Зокрема запропонований підхід комбінованої біометрії під час етапу делегування користувачем прав до клієнтського додатку.*

***Методологія.** У дослідженні застосовано комплексний підхід, що включав аналіз принципів роботи протоколу OAuth 2.0, моделювання потенційних загроз під час делегування користувачем прав клієнтському додатку. Методологія дослідження охоплювала теоретичний аналіз літератури з кібербезпеки та технологій біометричної верифікації, теоретичне моделювання можливих сценаріїв атак та впровадження додаткового шару автентифікації.*

***Наукова новизна.** У цій роботі вперше досліджено потенційну вразливість протоколу OAuth 2.0 PKCE до соціального інжинірингу на етапі делегації прав користувачем клієнтському додатку. Запропоновано інноваційний підхід до підсилення цього етапу шляхом впровадження біометричної верифікації, яка базується на унікальному алгоритмі, що забезпечує підвищений рівень захисту від атак з використанням методів соціальної маніпуляції.*

***Висновки.** Забезпечення безпеки інтеграцій веб-платформ є критичним аспектом для підприємств у сучасному бізнес-середовищі. Особливо важливим є захист процесу делегування прав користувачем клієнтському додатку, оскільки цей етап є вразливим до атак, зокрема соціального інжинірингу. Запропоно-*

ване вдосконалення протоколу OAuth 2.0 з використанням механізму Proof Key for Code Exchange (PKCE) через інтеграцію біометричної верифікації на етапі делегування прав надає додатковий рівень захисту, що дозволяє знизити ризики, пов'язані з соціальними маніпуляціями. Таким чином, запропонований підхід може бути застосований у практиці веб-програмування для забезпечення більш високого рівня безпеки інтеграцій, де високий рівень кіберстійкості має надзвичайно важливе значення.

Ключові слова: *OAuth 2.0, PKCE, веб-програмування, кібербезпека, аутентифікація, біометричні системи, веб-платформи, кіберстійкість.*

Andrii YARMOLATII
Lecturer at the Department of Cybersecurity, Information Technology and Economics, Kyiv University of Intellectual Property and Law of the National University «Odessa Law Academy», 210, Kharkivske highway, Kyiv, Ukraine, 02121, ayarmolatii@gmail.com
ORCID: *0009-0004-8655-9928*

Liubov CHEREMISINA
Lecturer at the Department of Cybersecurity, Information Technology and Economics, Kyiv University of Intellectual Property and Law of the National University «Odessa Law Academy», 210, Kharkivske highway, Kyiv, Ukraine, 02121, cheremisina1112@gmail.com
ORCID: *0009-0005-0719-0745*

Valentyn HALUNKO
PhD in law, Associate Professor at the Department of Administrative Law, Intellectual Property and Civil-Law Disciplines, Kyiv University of Intellectual Property and Law of the National University «Odessa Law Academy», 210, Kharkivske highway, Kyiv, Ukraine, 02121, valentinvalentin0987@gmail.com
ORCID: *0000-0002-8133-6766*

To cite this article: Yarmolatii, A., Cheremisina, L., Halunko, V. (2024). Pobudova veb-dodatkiv iz suchasnymy systemamy autentyfikatsiyi na bazi OAuth 2.0 PKCE ta biometryky dlya pidvyshchennya kiberstijkosti tsikh dodatkiv [Building web applications with modern authentication systems based on oauth 2.0 pkce and biometrics to enhance the cyber resilience of these applications]. *Information Technology: Computer Science, Software Engineering and Cyber Security,* 4, 242–251, doi: https://doi.org/10.32782/IT/2024-4-30

BUILDING WEB APPLICATIONS WITH MODERN AUTHENTICATION SYSTEMS BASED ON OAUTH 2.0 PKCE AND BIOMETRICS TO ENHANCE THE CYBER RESILIENCE OF THESE APPLICATIONS

The modern business world is technically founded on the integration of various business platforms, including web applications and web platforms. This integration encompasses not only data transfer and synchronization but also the synchronization of business processes and decision-making processes governed by these data. Therefore, the integrity and cybersecurity of such data are key aspects in mitigating risks and laying the groundwork for making sound decisions. Amid the rapid development of digital technologies and the significant rise in cyber threats, ensuring the security of business web applications has become particularly relevant and is often an essential part of corporate strategy when selecting or creating new web systems. One of the primary tools for securing integrations between web platforms is the OAuth 2.0 protocol with the Proof Key for Code Exchange (PKCE) mechanism.

Objective. *The aim of this work is to provide an analytical review of the technical aspects of the practical implementation of the OAuth 2.0 protocol and to create a proposal for its improvement through the application of biometric technologies using the Proof Key for Code Exchange (PKCE) algorithm, which is particularly important to understand when developing web applications (web programming). Specifically, the proposed approach involves combining biometrics during the user delegation phase of granting rights to a client application.*

Methodology. *The study applies a comprehensive approach, which includes an analysis of the principles of OAuth 2.0 protocol operation, modeling potential threats during the user's delegation of rights to the client application. The research methodology encompassed a theoretical analysis of literature on cybersecurity and biometric verification technologies, theoretical modeling of possible attack scenarios, and the implementation of an additional authentication layer.*

Scientific novelty. *This work investigates for the first time the potential vulnerability of the OAuth 2.0 PKCE protocol to social engineering during the delegation of user rights to a client application. An innovative approach is proposed to strengthen this stage by implementing biometric verification based on a unique algorithm that provides an enhanced level of protection against attacks using social manipulation techniques.*

Conclusions. *Ensuring the security of web platform integrations is a critical aspect for businesses in the modern business environment. The protection of the user rights delegation process to the client application is particularly important, as this stage is vulnerable to attacks, including social engineering. The proposed improvement of the OAuth 2.0 protocol with the use of the Proof Key for Code Exchange (PKCE) mechanism through the integration of biometric verification at the rights delegation stage provides an additional layer of protection, reducing risks associated with social manipulation. Therefore, the proposed approach can be applied in web programming practices to ensure a higher level of integration security, where a high level of cyber resilience is of utmost importance.*

Key words: *OAuth 2.0, PKCE, web programming, cybersecurity, authentication, biometric systems, web platforms, cyber resilience.*

Актуальність проблеми. У сучасному бізнес-середовищі інтеграція різних веб-платформ і сторонніх спеціалізованих веб-додатків є необхідною умовою для досягнення ефективності та гнучкості в управлінні інформацією. Згідно дослідницької платформи Gartner Magic Quadrant серед сучасних лідерів і найпоширеніших веб-платформ на ентерпрайз рівні що створюють можливість розробляти веб-додатки і інтегруватися з ними є такі веб-платформи як ServiceNow і Salesforce. Обидва ці інструменти дозволяють організаціям швидко автоматизувати бізнес-процеси та забезпечити ефективну взаємодію між різними компонентами корпоративних IT-систем. Ці платформи активно використовуються для управління IT-сервісами, взаєминами з клієнтами, підтримки співробітників та іншими критичними завданнями організацій, включаючи навіть операції кібербезпеки.

Проте ефективний і стійкий до атак обмін даними через створення інтеграцій веб-додатків з такими платформами потребує надійного механізму аутентифікації та авторизації, що одночасно забезпечував би і захист від несанкціонованого доступу і зменшення ризику викрадення зловмисниками атрибутів аутентифікації (паролі, токени тощо) що зазвичай передаються незахищеними каналами зв'язку або зберігаються в незахищених додатках клієнтської сторони інтеграції. Для забезпечення такого захисту сьогодні широко використовується протокол OAuth 2.0, спеціально підсилений механізмом PKCE (Proof Key for Code Exchange), що значно підвищує рівень безпеки особливо при інтеграціях з відкритими API та незахищеними додатками але має потенційні слабкі місця в алгоритмі, які потребують підсилення в сферах критично важливого надвисокого рівня кіберстійкості.

Аналіз останніх досліджень і публікацій. Серед останніх досліджень (Бодак, Дорошенко, 2022) досить детально і послідовно розглядаються протокол OAuth 2.0 і алгоритм PKCE відносно відкритих клієнтів в контексті проблеми авторизації таких клієнтів, приводиться набір потенційних атак і пропонується рішення через використання моделі BFF (Backend For Frontend). З іншої сторони в цій та інших (Білодід, 2024, с. 76–90) роботах мало уваги приділяється передумовам і причинам створення підсиленого флоу OAuth 2.0 PKCE, адже розуміння необхідності застосування того чи іншого механізму аутентифікації при побудові захищеної інеграції ефективної передачі даних між системами(платформами) базується на порівнянні знань про попередні або альтернативно існуючі механізми. Детальний розгляд таких механізмів не є фокусом даної статті, але слід зауважити, що більшість попередньо існуючих механізмів (серед них Базова Аутентифікація – Basic Authentication, використання API Key в заголовку надісланих пакетів інформації тощо) виконувались в один крок і вимагали від користувача ділитися своїми паролями зі сторонніми додатками та веб-сервісами для отримання доступу до їх облікових даних, а оскільки можливості 100% гарантувати, що при передачі паролей інформація не буде отримана і використана зловмисником, виникла необхідність в новому підході до алгоритму аутентифікації, що і призвело до появи OAuth 2.0.

Метою даної статті є аналіз алгоритму роботи OAuth2.0 та PKCE і потенційно вразливих моментів та пропозиція по їх підсиленню при веб програмуванні додатків та платформ для отримання високого рівня їх кіберстійкості.

Архітектура веб-додатків і веб-додатків і платформ. На початку слід згадати загальну структуру більшості веб-додатків яка має клієнт-серверну архітектуру (рис. 1):

Як вказано на рис. 1. – весь додаток – це поєднання(інтеграція) окремих систем і механізмів як на клієнтській та і на серверній частині, але серверна частина завжди складніша і має бути потужніша, адже своїм основним призначенням має обробку великої кількості запитів від множини клієнтських додатків та процессінг даних з бази даних.

Якщо серверну частину умовно можна вважати більш захищеною по причині віддаленості від зловмисника і складності фізичного доступу, то клієнтська частина і сама взаємодія між сервером і клієнтом потребує створення додаткового захисту – кіберстійкої системи аутентифікації.

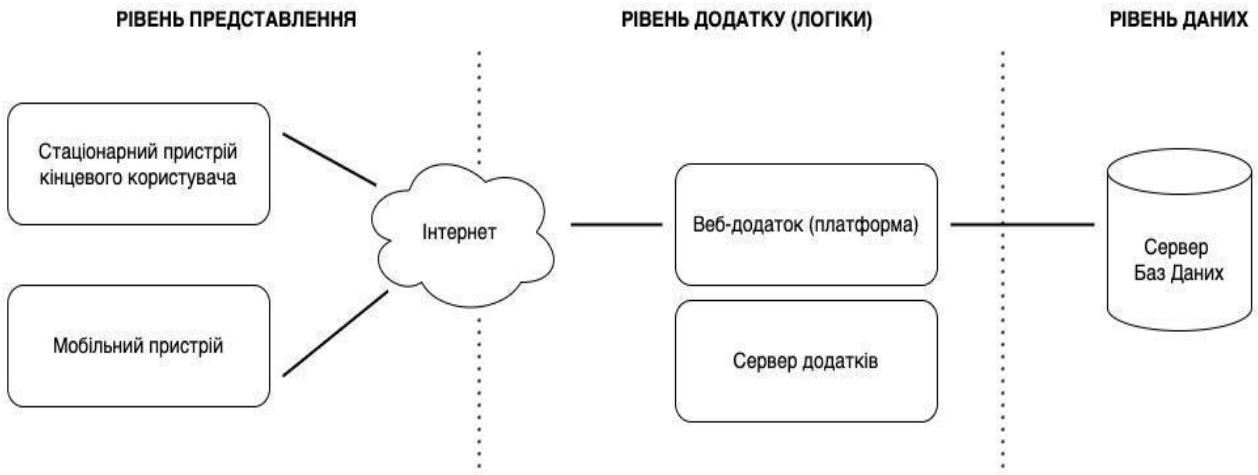

Рис. 1. Загальна схема веб-додатків

При цьому підхід до створення веб-додатків в веб-програмуванні може різнитися в залежності від багатьох факторів, в тому числі від середовища і предметної області для яких цей додаток розробляється. Веб-програмування додатків до вже існуючої платформи значно спрощений, але розробка веб-додатку так само часто може розпочинатися з нуля і вимагатиме більше часу на вирішення нетривіальної задачі правильного вибору компонент. Проте оскільки більшість веб-додатків пишуться розробниками на сьогоднішній день на основі JavaScript та Node.js, це автоматично звужує набір варіантів та задає шаблон розробки – використання бібліотек і модулів, які значною мірою скорочують час розробки.

На сьогоднішній день в веб-програмуванні серед модулів і бібліотек JavaScript/Node.js для реалізації системи аутентифікації OAuth 2.0 найчастіше використовуються: passport, passport-oauth2, simple-oauth2, express-oauth-server, express-oauth2-bearer, openid-client, oidc-provider, jsonwebtoken тощо. Серед них найчастіше використовують passport, passport-oauth2 та simple-oauth2. Їх широке застосування і популярність супроводжуються підтримкою глобального комьюніті, а серед переваг визначають простоту використання базових сценаріїв і широкий спектр інтеграційних можливостей з іншими системами.

Передумови OAuth 2.0 та Implicit Flow.

На рис. 2. продемонстрована класична та історично перша архітектура інтеграції двох систем з використанням паролю (або API Key) для аутентифікації. Літерами (а) та (б) вказано небезпечні елементи такої архітектури, де а) не захищений канал передачі аутентифікаційних даних – email, sms тощо та б) незахищений додаток, до якого може отримати доступ зловмисник і рано чи пізно отримати паролі що там зберігаються. Звісно, що в кожному окремому випадку величина такої ймовірності різна, але проблемою залишається наявність такої можливості для зловмисника, тому використовувати такий підхід в критично важливих процесах було б дуже ризиковано.

Враховуючи вказані вище ризики, OAuth 2.0 був розроблений як стандарт для так званої делегованої авторизації, що дозволяє стороннім додаткам отримувати обмежений доступ до ресурсів користувача без необхідності передавати свої облікові дані. Тепер, замість одно-крокового алгоритму(очевидний флоу) аутентифікації маємо протокол, який передбачає і додатковий елемент безпеки (крок) і використання так званих «токенів доступу», що генеруються сервером авторизації і використовуються клієнтами для доступу до захищених ресурсів (неявний потік – Implicit Flow). Ключовими елементами протоколу OAuth 2.0 є наступні:

1. Сервер авторизації (Authorization Server) – сервер, який відповідає за аутентифікацію користувачів і видачу токенів доступу
2. Сервер ресурсів (Resource Server) – сервер, що зберігає захищені ресурси (наприклад, дані користувачів або іншу необхідну інформацію), і перевіряє токени доступу
3. Клієнт (Client) або клієнтський додаток – додаток, який звертається до ресурсів користувача за допомогою отриманого токена доступу
4. Тип Авторизації (Authorization Grant) – тип авторизації, який застосовує OAuth для видачі коду доступу.

Розглянемо класичний спенарій роботи OAuth 2.0, що зображено на Рис. 3

Як видно на рис. 3., клієнтський додаток перед запитом до серверу ресурсів спочатку

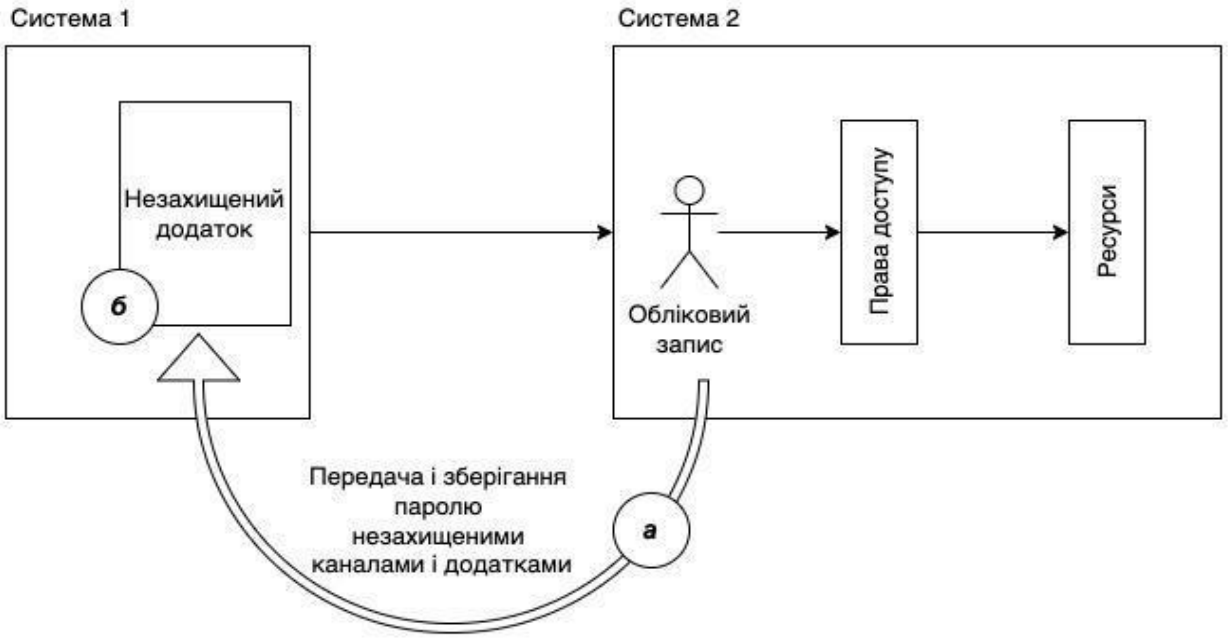

Рис. 2. Передача і зберігання паролей незахищеними каналами та додатками

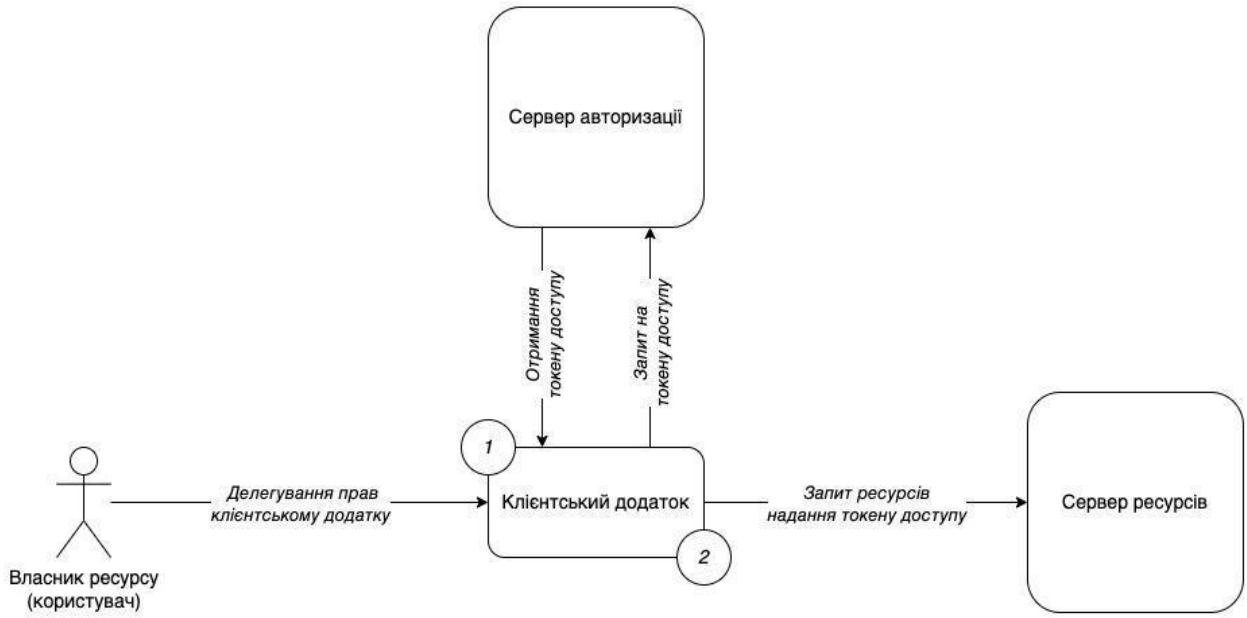

Рис. 3. Класичний сценарій OAuth 2.0

взаємодіє з авторизаційним сервером наступним чином:

1. Отримання коду авторизації: клієнт перенаправляється на авторизаційний сервер, де користувач надає дозвіл на доступ до своїх ресурсів.

2. Обмін коду на токен доступу: після отримання коду клієнт відправляє його на сервер для отримання токена доступу.

Варто зауважити, що все частіше сервер авторизації і сервер ресурсів є технічно двома компонентами однією і тієї ж системи (платформи) і навіть більше того – веб-додаток чи платформа мають універсальні механізми, що дозволяють їм вести себе і як сервер і як клієнтський додаток. Так, наприклад, в веб-платформі ServiceNow зберігаються і інформація як ресурс і налаштування аутентифікації та авторизації. Для цього достатньо зайти в аплікейшн меню System OAuth та обравши пункт Application Registry (реєстр додатків), перейти до конфігураційної форми (рис. 4.)

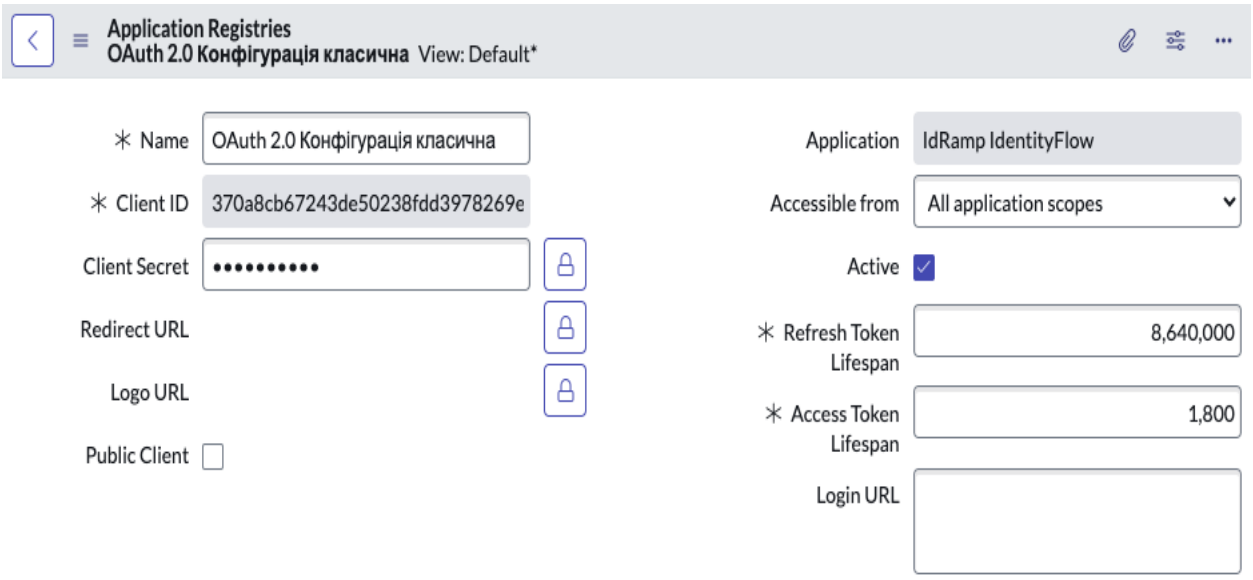

Рис. 4. Конфігурація OAuth 2.0 аутентифікації на веб-платформі ServiceNow

Все, що необхідно зробити для конфігурації серверу аутентифікації на даному етапі – просто додати Redirect URL, адже параметри Client ID та Client Secret генеруються автоматично системою. Серверна частина сконфігурована і можнавикористовувати клієнтський додаток (наприклад Postman) для тестування. За схожим принципом створюється конфігурація OAuth-серверу і в веб-платформі Salesforce, де спочатку необхідно перейти до менеджеру додатків (App Manager) а потім створити нову сутність підключаємого додатку (Connected App). В обох платформах відбувається збереження інформації про підключаємий веб-додаток.

Підсилення аутентифікації OAuth 2.0 механізмом PKCE. При аналізі форми конфігурації OAuth платформи ServiceNow (рис. 4) чекбокс Public Client було навмисне не використано адже розглядався класичний сценарій OAuth 2.0. І як вже було згадано раніше (Бодак, Дорошенко, 2022) – такий механізм має ряд вразливостей при роботі з публічними (відкритими, незахищеними) клієнтськими додаткам, що в свою чергу може понизити рівень кіберстійкості такої системи. На Рис. 5 продемонстрована схема атаки перехоплення коду авторизації:

Як видно на рис. 5., серед можливих вразливостей OAuth 2.0 існує можливість атаки на перехоплення коду авторизації додатком зловмисника на пристрої кінцевого користувача. Інші можливі атаки були розглянуті раніше (Бодак, Дорошенко, 2022). Така ситуація пояснюється архітектурою взаємодії з сервером авторизації, де не сам додаток, а операційна система чи браузер комунікують з сервером для отримання коду авторизації і далі передають його додатку, про існування якого сервер авторизації не знає, тобто немає безпосередньої привязки а значить і контролю до клієнтського додатку. Це в свою чергу призвело до необхідності покращення алгоритму захисту OAuth 2.0 що і призвело до створення механізму PKCE (Proof Key for Code Exchange), який був вперше представлений у жовтні 2014 року як частина RFC 7636, доповнення до OAuth 2.0. PKCE був створений для вирішення проблем безпеки, пов'язаних з аутентифікацією в мобільних додатках та інших клієнтах, які не можуть безпечно зберігати конфіденційні дані, як-от клієнтські секрети.

Серед причин виникнення PKCE виділимо наступні:

- ***Відсутність клієнтського секрету***: У мобільних додатках і браузерних клієнтах немає безпечного способу зберігати клієнтський секрет. PKCE усуває необхідність у клієнтському секреті для захисту процесу авторизації.
- ***Захист від атаки підміни коду (Code Injection)***: Без PKCE зловмисник може перехопити код авторизації та обміняти його на токен доступу, якщо у них є можливість перехопити трафік між клієнтом і сервером. PKCE додає додатковий рівень безпеки, використовуючи динамічно згенерований секрет для підтвердження того, що запит на отримання токена доступу походить від того ж клієнта, що й запит на авторизацію.

Рис. 5. Перехоплення Коду Авторизації

● ***Поліпшення безпеки***: PKCE покращує безпеку OAuth 2.0, забезпечуючи, що тільки клієнт, який ініціював авторизацію, може обміняти код на токен доступу. Це досягається шляхом додавання додаткового параметра code_verifier і його хеш-версії code_challenge до запиту авторизації.

Алгоритм роботи PKCE. На рис. 6 показана загальна схема роботи PKCE алгоритму в контексті OAuth 2.0 аутентифікації.

1. Клієнт(клієнтський додаток) генерує випадковий код перевірки (code_verifier) і обчислює хеш (code_challenge).

2. Клієнт направляє запит на авторизацію разом з code_challenge.

3. Користувач надає дозвіл на доступ до своїх даних.

4. Клієнт отримує код авторизації.

5. Клієнт відправляє код авторизації разом з code_verifier для отримання токена доступу.

6. Авторизаційний сервер перевіряє відповідність code_challenge і code_verifier, і якщо вони співпадають, видає токен доступу.

Таким чином, навіть якщо зловмисник отримає код авторизації, він не зможе використати його для отримання токена без доступу до первісного коду перевірки. Хоча цей механізм і підвищує кіберстійкість OAuth 2.0, особливо вразливим є третій крок, де користувач надає дозвіл на доступ до своїх даних. Адже на практиці надання доступу відбувається наступним чином – користувачу виводиться або діалогове вікно або сторінка, на якій в кращому випадку необхідно ввести логін і пароль користувача а не рідко може бути просто кнопка “Дозволити” без особливої верифікації персони, яка надає доступ, що робить алгоритм вразливим для соціального інжинірингу та потенційних маніпуляцій користувачем.

Удосконалення OAuth 2.0 за допомогою PKCE. Розглянуту вище слабку точку алгоритму PKCE можна підсилити різними методами, але розглянемо три найперспективніші – використання біометричної технології, додаткова верифікація власника ресурсів (користувача) та комбінований підхід.

1. Використання біометричної технології. Біометричні технології, такі як сканування відбитків пальців, розпізнавання обличчя або голосу, можуть бути інтегровані як додатковий рівень автентифікації перед наданням доступу до даних. Це значно ускладнить можливість несанкціонованого доступу, оскільки зловмиснику буде потрібно не лише код авторизації, але й фізичний доступ до біометричних даних користувача.

2. Додаткова верифікація власника ресурсів. Це може включати відправлення одноразового

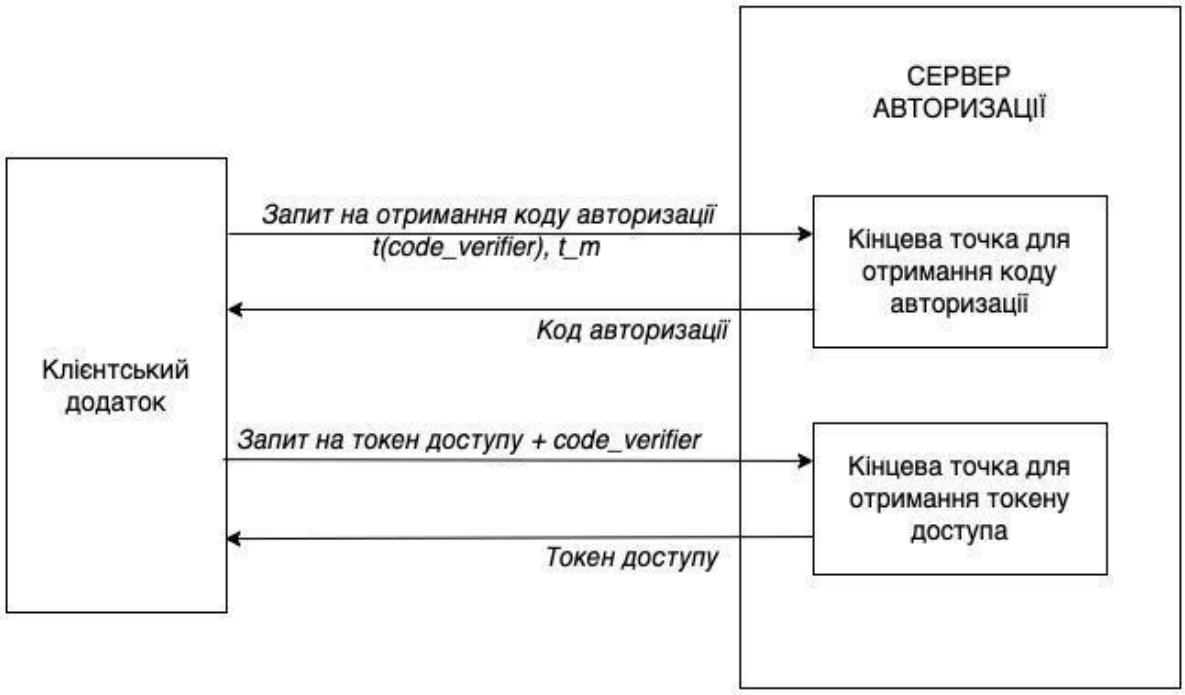

Рис. 6. Загальна схема роботи OAuth 2.0 PKCE

пароля (OTP) на зареєстрований номер телефону або електронну пошту користувача для підтвердження дії. Такий підхід забезпечує додатковий рівень захисту, оскільки користувач має підтвердити своє бажання надати доступ через окремий канал зв'язку.

3. Комбінований підхід. Комбінований підхід передбачає використання біометричної верифікації, яка здійснюється на сторонньому сервісі. Це забезпечує додатковий рівень безпеки, оскільки верифікація проходить поза основною системою, зменшуючи ризик компрометації локальних даних. Таким чином, навіть якщо зловмисник отримає доступ до локальних даних користувача, він не зможе пройти біометричну перевірку на сторонньому сервісі. Комбінований підхід рекомендовано до реалізації де потрібно створити особливо високий рівень кіберстійкості, адже комплексна система захисту завжди використовує поєднання засобів і заходів а не окремий інструмент.

Розглянемо схему комбінованої верифікації користувача OAuth 2.0 PKCE

Як видно на рис. 7, делегування доступу клієнтському додатку (клієнту) супроводжується додатковою верифікацією зі стороннім сервером верифікації та використанням біометричних технологій. В такій схемі, перед тим як делегується доступ, користувачу буде надіслане посилання (через sms або email) з посиланням для проходження верифікації. Користувач повинен довести, що це він делегує доступ а не хтось за його пристроєм. Тому перейшовши за посиланням, користувач ініціює процедуру верифікації і при цьому застосовує біометрику – або сканування відбитку пальця, або сканування обличчя тощо. Якщо через біометричні атрибути користувач довів що це він, то сервер верифікації дає сигнал дозволу на делегування. Окрім цього, слід застосувати комбінацію біометричних перевірок наступним алгоритмом: якщо користувач попередньо не пройшов, наприклад, біометричну верифікацію відбитком пальця, то наступна верифікація буде вже скануванням обличчя або комбінацією обох щоб мінімізувати ризик соціального інжинірингу або дати можливість зловмиснику на повторний однотипний маневр.

Інформування клієнта (клієнтського додатку) про завершення і результати верифікації є нетривіальною задачею, адже користувач не повинен нічого додатково робити в кінці процесу верифікації і при цьому і сервер верифікації і клієнт мають отримати відповідні дані саме стосовно цього користувача. Це досягається за допомогою загальнодоступного веб-сервісу і методу GET протоколу HTTP, що на практиці означає звичайне переведення користувача на іншу сторінку і унікальними параметрами, що передаються в адресному рядку. Така сторінка містить інформацію про завершення верифікації і інформує користувача, що той може закрити вкладку браузера. Так, загальнодоступна реалізація API результатів верифікації може мати наступну послідовність:

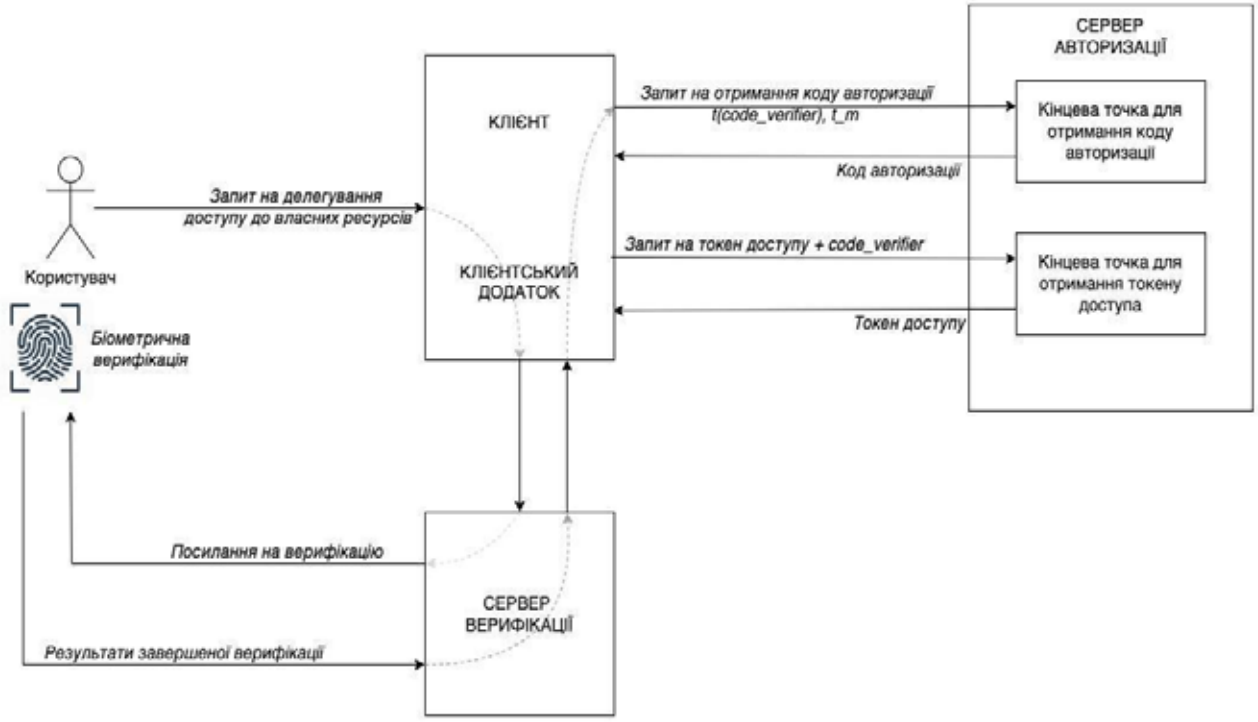

Рис. 7. Архутектура підсилення етапу делегації послідовності OAuth 2.0 PKCE

- Отримання інформації про користувача з URL
- Надсилання запиту до зовнішнього сервера верифікації
- Оновлення статусу перевірки на основі відповіді від сервера
- Парсинг результату з відповіді

Таким чином, запропонована архітектура підсилення PKCE біометричною верифікацію створює новий, більший високий, рівень кіберстійкості веб-додатків чи платформ де він буде застосований.

Висновки. OAuth 2.0 з PKCE є потужним інструментом для створення безпечних інтеграцій між сучасними бізнес веб-платформами, такими як ServiceNow та Salesforce, а також у веб-додатках, де неможливо гарантувати захист клієнтського секрету на пристрої користувача. Ця конфігурація протоколу дозволяє ефективно захищати дані при їх обміні між системами, знижуючи ризик несанкціонованого доступу. Впровадження PKCE у таких інтеграціях підвищує рівень безпеки, гарантуючи цілісність та конфіденційність інформації. Однак PKCE має потенційну вразливість, зокрема в процесі делегації прав доступу, де існує ймовірність соціального інжинірингу. Тому в середовищах, де вимагається високий рівень кіберстійкості, рекомендовано доповнювати цей процес верифікацією користувача з використанням біометричних технологій.

ЛІТЕРАТУРА:

1. The OAuth 2.0 Authorization Framework. Microsoft Internet Engineering Task Force (IETF). URL: https://datatracker.ietf.org/doc/html/rfc6749 (дата звернення 11.10.2024).

2. Proof Key for Code Exchange by OAuth Public Clients. URL: https://datatracker.ietf.org/doc/html/rfc7636 (дата звернення 11.10.2024)

3. Бодак В. В., Дорошенко А. Ю., Захист відкритих клієнтів за допомогою одного алгоритму авторизації. *Проблеми програмування*. 2022. № 3-4. С. 409–416.

4. Радівілова Т., Кіріченко Л., Пантєлєєв В., Мазепа А., Білодід В. «Аналіз методів автентифікації для вебзастосунків та реалізація вебзастосунку з інтегрованою системою автентифікації», СУЧАСНИЙ СТАН НАУКОВИХ ДОСЛІДЖЕНЬ ТА ТЕХНОЛОГІЙ В ПРОМИСЛОВОСТІ, 2024. (3(29), с. 76–90. doi: 10.30837/2522-9818.2024.3.076.

5. Authgear Team. (2024). PKCE in OAuth 2.0: How to Protect Your API from Authorization Code Grant Attacks. *Authgear Blog*. https://doi.org/10.48550/arXiv.2412.07012

6. Authgear Team. (2024). PKCE in OAuth 2.0: How to Protect Your API from Attacks. *Authgear Blog*. https://doi.org/10.48550/arXiv.2412.07012

7. Passport.js Team. (2020). PKCE Support for OAuth 2.0. *Medium*. https://medium.com/passportjs/pkce-support-for-oauth-2-0-e3a77013b278 (дата звернення 14.10.2024)

8. Identity Beyond Borders. (2020). What the heck is PKCE? *Medium*. https://medium.com/identity-beyond-borders/what-the-heck-is-pkce-40662e801a76

REFERENCES:

1. The OAuth 2.0 Authorization Framework. Microsoft Internet Engineering Task Force (IETF). Retrieved from: https://datatracker.ietf.org/doc/html/rfc6749

2. *PKCE Proof Key for Code Exchange by OAuth Public Clients*. Retrieved from: https://datatracker.ietf.org/doc/html/rfc7636

3. Bodak, V. V., Doroshenko, A. Yu. (2022). Zakhyst vidkrytykh kliientiv za dopomohoyu odnoho alhorytmu avtoryzatsii [Protection of open clients using a single authorization algorithm]. *Problemy prohramuvannya*. 3-4, 409–416. [in Ukrainian]

4. Radivilova, T., Kirichenko, L., Pantelieiev, V., Mazepa, A., Bilodid, V. (2024). «Analiz metodiv avtentifikatsii dlia vebzastosunkiv ta realizatsiia vebzastosunku z intehrovanoiu systemoiu avtentifikatsii» [Analysis of Authentication Methods for Web Applications and Implementation of a Web Application with an Integrated Authentication System] SUCHASNYI STAN NAUKOVYKH DOSLIDZHEN TA TEKHNOLOHII V PROMYSLOVOSTI, (3(29), s. 76–90. doi: 10.30837/2522-9818.2024.3.076 [in Ukrainian]

5. Authgear Team. (2024). *PKCE in OAuth 2.0: How to Protect Your API from Authorization Code Grant Attacks*. Authgear Blog. https://doi.org/10.48550/arXiv.2412.07012

6. Authgear Team. (2024). *PKCE in OAuth 2.0: How to Protect Your API from Attacks*. Authgear Blog. https://doi.org/10.48550/arXiv.2412.07012

7. Passport.js Team. (2020). *PKCE Support for OAuth 2.0*. Medium. Retrieved from: https://medium.com/passportjs/pkce-support-for-oauth-2-0-e3a77013b278

8. Identity Beyond Borders. (2020). *What the heck is PKCE?* Medium. Retrieved from: https://medium.com/identity-beyond-borders/what-the-heck-is-pkce-40662e801a76