ПРИКЛАДНІ ПРОБЛЕМИ КОМП'ЮТЕРНИХ НАУК, БЕЗПЕКИ
ТА МАТЕМАТИКИ, 2024, № 3

УДК 004.56

СУЧАСНІ СПОСОБИ АВТЕНТИФІКАЦІЇ ТА ЗАХИСТ НА ОСНОВІ ТОКЕНІВ

Здолбіцька Н.В.[1] (ORCID: 0000-0002-1345-3581),
Жигаревич О.К.[2] (ORCID: 0000-0002-7154-9733),
Бас Д.В.[1]
[1]*Луцький національний технічний університет*
[2]*Волинський національний університет імені Лесі Українки*

MODERN METHODS OF AUTHENTICATION AND TOKEN-BASED SECURITY

Zdolbitska N.V.[1], Zhyharevych O.K.[2], Bas D.V.[1]
[1]*Lutsk National Technical University,*
[2]*Lesya Ukrainka Volyn National University*

Abstract. With the development of digital transformation, new problems have arisen related to the protection of users' own accounts and personal data. The development of business and services has caused the need to use a large number of web applications to support everyday life, and therefore the number of users of these applications has increased. As the number and complexity of web applications grows, developers are faced with the need to provide the latest level of scalability and complexity to ensure user security, using currently popular micro- and nano-authorization services. User identity and access management aims to ensure that people have the right access to the right resources and prevent unauthorized users from entering. To provide single sign-on across multiple accounts and logins, there is a set of authorization, authentication, and single sign-on (SSO) protocols called OpenID Connect. The authentication mechanism is considered, clearly explaining how token-based authentication works and what are the main factors that drive the entire security process. With token-based authentication, users have the option to log into their own accounts using a smartphone or security key, or for passwordless operation. With token-based authentication, the user is checked for access to credentials once in a certain period of time, there is no need to register continuously.

Keywords: security, authorization, authentication, Single Sign-On (SSO), OpenID Connect protocol, JSON, JWT, HTTP, web application.

Вступ. Розвиток цифрової трансформації викликає нові проблеми, що пов'язані з захистом власних акаунтів та особистих даних користувачів. Щороку в середньому зламується близько десяти тисяч облікових записів по даним Norton LifeLock [1]. Основним чинником є злом облікових даних, тобто через використання однакових паролів, розкритих чи зламаних облікових даних, для взлому облікових акаунтів користувачів на різних сайтах чи службах. Отже, необхідне використання системи високого рівня захисту та правила безпеки для автентифікації [2].

Постановка проблеми. Протягом останніх років спостерігається раптовий сплеск розробки та використання вебдодатків. Розвиток бізнесу та послуг спричинив потребу використання більшої кількості вебдодатків для підтримки повсякденного життя, а отже і збільшення кількості користувачів цих додатків. Адже користувачам дуже зручно працювати онлайн: вести електронний комерційний бізнес чи користуватися послугами, наприклад комунальні послуги, сплата рахунків за доставку продукції, медичні консультації, використання різних соціальних мереж тощо.

Розробка вебдодатків має супроводжуватися механізмами авторизації та автентифікації користувачів. По мірі того, як кількість та складність вебдодатків зростає,

для забезпечення безпеки користувачів розробники стикаються з тим, що необхідно забезпечити новітній рівень масштабованості та складності, застосовувавши популярні на даний час мікро- та нано-сервіси авторизації. Так перед розробниками постає проблема дослідити та вибрати той з варіантів різних механізмів автентифікації та авторизації, який найкраще забезпечуватиме потреби захисту користувачів.

Супроводження будь-якого вебдодатка вимагає системи керування доступом великої кількості користувачів відповідно їхнім конкретним потребам. Із збільшенням кількості акаунтів користувачів виникає серйозна проблема запам'ятовування та використання великої кількості логінів та паролів для кожного вебдодатка. Вирішення цієї проблеми можливе при використанні технології єдиного входу (SSO), адже користувач може увійти один раз, а потім відвідувати відповідний вебдодаток без необхідності реєстрації знову.

Аналіз останніх досліджень і публікацій. Основні проблеми захисту мережі – це загрози, засоби протидії, методи автентифікації [3,4]. Для вдосконалення безпеки мережі та моніторингу інформації про безпеку мережі в режимі реального часу, у статті [5] запропоновано метод контролю доступу до інформації автентифікації захисту мережі, що базується на основі алгоритму нечітких міркувань. Систематичний огляд джерел з запитань, що стосуються автентифікації та авторизації у мікросервісах, захисту у мікросервісах проведено автором [6], подано аспекти відповідних викликів, механізмів та технологій. Зв'язок між мікросервісами враховує їхні індивідуальні та надійні характеристики, для забезпечення захисту автентифікації застосовуються механізми OAuth 2.0 [7], OpenID Connect [8], API Gateway, JWT. Протокол автентифікації та авторизації OAuth 2.0 дуже поширений у вебдодатках [7] та серверних програмах [9].

Автори [10] запропонували протокол трифакторної автентифікації (3FA) для додатків IoT, який з точки зору безпеки та функціональних можливостей є надійнішим протоколом, адже базується на біометричних даних користувачів додатків IoT. Серед технологій децентралізованої ідентифікації та контролю доступу для IIoT можна виділити декілька механізмів, зокрема серед них засіб контролю доступу на основі токенів OpenID Connect, OAuth, вебтокенів JSON, модель керування доступом на основі ролей (RBAC) чи атрибутів (ABAC) [11].

Методологія дослідження. У даній статті розглянуто сучасні способи реалізації певного механізму автентифікації та авторизації та захист на основі токена OpenID Connect.

Методи дослідження: аналіз літературних та інформаційних джерел, аналіз способів захисту на основі автентифікації та реалізація захисту вебдодатка на основі токена OpenID Connect.

Результати дослідження та їхнє обговорення. У сучасному динамічному світі, в якому цифрові технології постійно удосконалюються, необхідним також стає і трансформація в безпеці. При збільшенні кіберзагроз на сьогоднішній день традиційних способів автентифікації вже недостатньо. Сучасний працівник може працювати, при цьому використовувати корпоративні ресурси з різних місць чи пристроїв. Використання традиційних паролів стає вразливим до численних загроз, зокрема фішингові атаки, використання викрадених облікових даних користувачів, тощо.

Під аутентифікацією розуміють основу безпеки будь-якої системи, що полягає у перевірці сервером достовірності даних про користувача, що реєструється в програмі чи додатку (рис. 1).

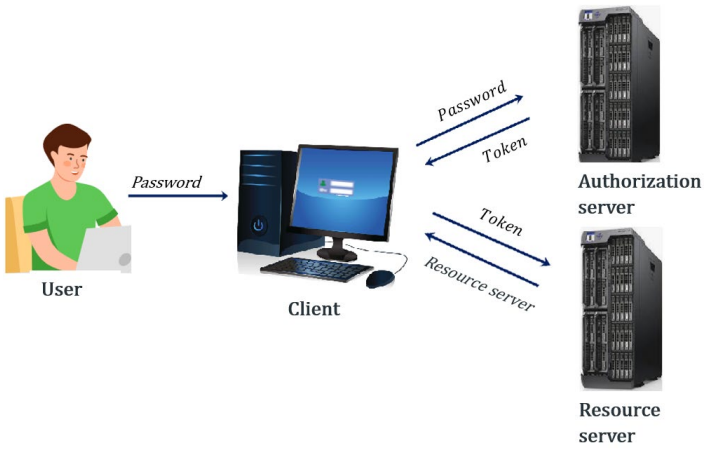

Рис. 1. Аутентифікація

Керування ідентифікацією та доступом користувачів має на меті забезпечення належного доступу людей до потрібних ресурсів та унеможливлення допуску неавторизованих користувачів.

Розглянемо основні типи автентифікації, що використовуються для захисту користувачів:

- надійний пароль;
- система єдиного входу (SSO);
- багатофакторний вхід;
- біометрична;
- токени;
- сертифікати [12].

Застосування аутентифікації на основі пароля базується на основі імені користувача та пароля або PIN-коду. Такий метод автентифікації є найпоширенішим та найпростішим, та його легко зламати, оскільки користувачі досить часто повторно застосовують свої паролі або придумують паролі, що можна легко вгадати, скориставшись загальнодоступною особистою інформацією, наприклад з соцмереж. На багатьох підприємствах працівникам потрібен пароль для користування кількома програми та пристроями, що в свою чергу переобтяжує запам'ятовування та спонукає їх по можливості спрощувати паролі, тоді облікові дані стають вразливими до фішингу та кібератак. Вирішенням цієї проблеми стає необхідність створювати політику паролів, яка не дозволяє повторне використання паролів, регулярну зміну паролів та використання надійних паролів, хоча б певної довжини та в включенням спеціальних символів.

Якщо користувачі використовують двофакторну або багатофакторну аутентифікацію, це посилить безпеку облікових записів, оскільки, щоб отримати доступ зловмисникам необхідні не лише облікові дані, в такому випадку крім пароля користувачам варто додатково використовувати будь-які типи автентифікації вказані вище [12]: хоча б одноразовий пароль, що надсилається користувачеві як текстове повідомлення на мобільний телефон або електронну пошту.

Застосування біометричних даних для аутентифікації на багатьох споживчих пристроїях спрощує вхід до облікових даних (Windows Hello, Apple Face ID, Touch ID), ускладнює злом облікових записів, так як біометричні дані є унікальними, як приклад сканування відбитків пальців, розпізнавання райдужної оболонки ока чи обличчя.

Система єдиного входу (Single sign-on, SSO) дозволяє забезпечити безпеку користувачам, застосовуючи один набір облікових даних, щоб отримати доступ до кількох програм або сайтів. Для цього користувачам створюють обліковий запис

ідентифікаційної інформації (IdP). IdP перевіряє, чи користувач перевірив дані входу через cookie, проте необхідно затратити більше часу, щоб здійснити налаштування та доступ до різних програм та сайтів через SSO. У системі єдиного входу ідентифікаційні дані набувають форми токенів, що містять ідентифікаційні значення інформації про користувача. Провайдер послуг відправляє токен, в якому міститься інформація про користувача (email або ім'я користувача) системі SSO як частина запиту на автентифікацію користувача (рис. 2). У процесі автентифікації необхідним є забезпечення надійного надсилання даних зв'язку між клієнтом та віддаленим сервером.

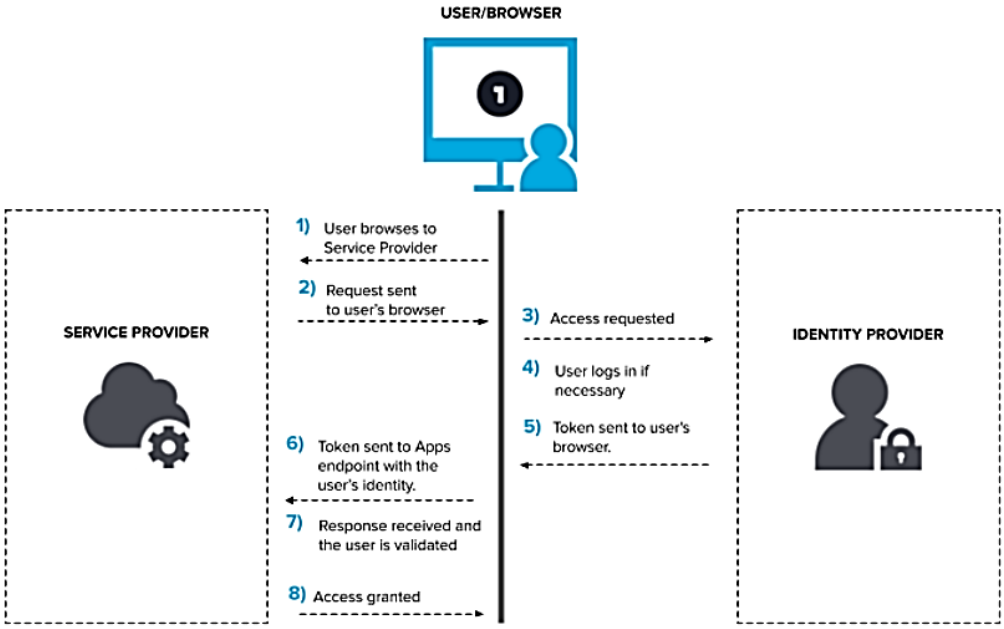

Рис. 2. Налаштування доступу постачальником послуг відповідно до методу SSO

Користувачі, здійснивши аутентифікацію на основі токенів, мають можливість входити у власні облікові записи за допомогою смартфона чи ключа безпеки або для забезпечення роботи без пароля. При автентифікації на основі токенів користувачем перевіряється доступ до облікових даних один раз за певний період періоду часу, немає необхідності здійснювати реєстрацію постійно. Це ускладнює доступ до облікових записів користувачів зловмисникам, щоб зламати обліковий запис, потрібен фізичний доступ до токена та знання облікових даних користувача.

При автентифікації на основі сертифікатів застосовуються цифрові сертифікати для перевірки особи користувача, видані центром сертифікації. Аутентифікацію на основі сертифіката зручно використовувати на підприємствах, що винаймають підрядників для надання тимчасового доступу до мережі. Водночас розгортання автентифікації на основі сертифікатів може бути затратним та трудомістким. Адже необхідно повторно робити нові облікові записи користувачів, якщо ключ вкрали чи вони не можуть отримати доступ до своїх ключів, якщо пристрій зламався.

OpenID – це протокол із відкритим кодом для автентифікації та SSO, для ідентифікації користувачів замість здійснення входу на певний сайт безпосередньо перенаправляють на сайт OpenID для входу. Перевага токенів полягає в тому, що як тільки користувач отримав токен, він може використовувати його знову і знову без необхідності повертатися до постачальника ідентифікаційної інформації. Звичайно, мають бути певні

обмеження, і тому токени мають термін дії, після закінчення цього періоду користувачеві потрібно буде отримати новий токен.

Для забезпечення єдиного входу у кілька облікових записів та входів, великі технологічні корпорації, зокрема Google, Microsoft, Oracle тощо, створили спільну групу OpenID Foundation, сформували набір протоколів авторизації, автентифікації та систему єдиного входу (SSO) під назвою «OpenID Connect Protocol Suit». На рисунку 3 подано схема протоколу OpenID Connect [13]:

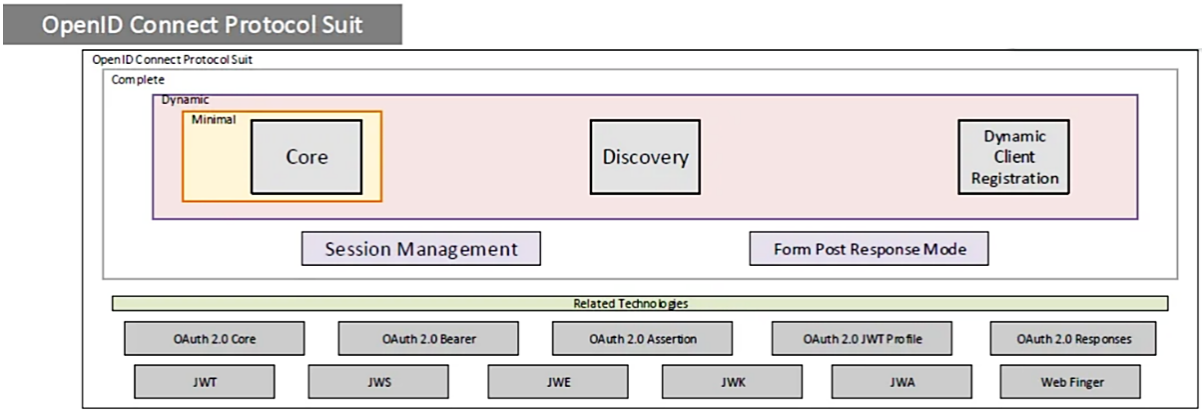

Рис. 3. Протокол OpenID Connect

Аутентифікація на основі токенів стала широко використовуваним механізмом безпеки, який застосовується постачальниками послуг в Інтернет, щоб запропонувати користувачам швидку взаємодію, не ставлячи під загрозу безпеку їх даних. Токен автентифікації – це згенерований комп'ютером код, який використовується для автентифікації користувача для доступу до будь-якого вебдодатку та замінює підпис користувача в цифровому кодуванні. Токени можуть бути двох типів: фізичний та вебтокен, обидва відіграють важливу роль для безпеки. Розглянемо чотири кроки механізму аутентифікації, які чітко пояснюють, як працює автентифікація на основі токенів і які основні чинники керують усім процесом безпеки:

- запит;
- верифікація;
- перевірка токена;
- зберігання;
- термін дії.

При запиті користувач пробує увійти в програму або інтерфейс вебдодатку за допомогою власних облікових даних для входу. Облікові дані користувач містять ім'я користувача, його пароль, біометричні дані, тощо.

Для верифікації інформації для входу з клієнт-сервера надсилаються дані на сервер автентифікації для перевірки діючих користувачів, що намагаються увійти до обмеженого ресурсу. Якщо верифікація облікових даних пройшла успішно, сервер генерує секретний цифровий ключ користувача у вигляді коду через HTTP.

Приклад верифікації сигнатур токена, додавання у контролер User логін обробник подано на рисунку 4.

Токен надсилається у відкритому стандартному форматі JWT, що включає:

- заголовок, у якому вказується тип токена та алгоритм підпису;
- корисне навантаження, що містить інформацію про користувача та інші типи даних;

- підпис (Signature), за допомогою якого перевіряється автентичність користувача та переданих повідомлень.

```
// GET: api/Users
        [HttpGet("Login")]
        public async Task<ActionResult<UserWithToken>> Login([FromBody] User user)
        {
            user = await _context.Users
                          .Where(u => u.EmailAddress == user.EmailAddress
                              && u.Password == user.Password)
                          .FirstOrDefaultAsync();

            UserWithToken userWithToken = new UserWithToken(user);

            if (userWithToken == null)
            {
                return NotFound();
            }

            return userWithToken;
        }
```

Рис. 4. Верифікація токена

При перевірці токена користувач отримує код токена (рис. 5) та представляє його на сервері ресурсів для отримання доступу до ресурсів (рис. 6). При цьому токен доступу зазвичай має обмежений термін дії (30-60 секунд) та існує обмеження щодо кількість спроб для отримання доступу. У випадку, якщо користувач не встиг застосувати його, то він може запросити токен оновлення повторно на сервері автентифікації.

```
https://localhost:44304/api/Users/Login
{"emailAddress" : "john.smith@gmail.com", "password" : "8be7dbd7237e2e0bf90ff81b8ff44333"}
```

Рис. 5 – Код токена

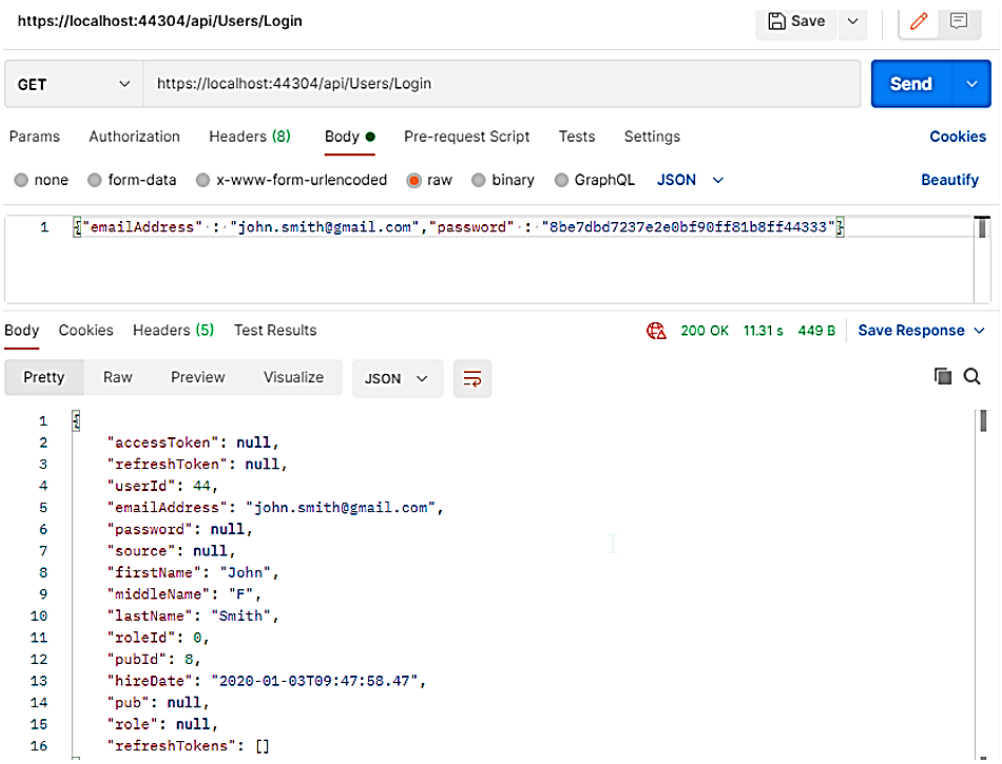

Рис. 6. Перевірка токена на сервері

Додавання секретного ключа для шифрування у файл налаштувань подано на рис. 7. Опис сервісу аутентифікації подано на рис.8.

```
"JWTSettings": {
    "SecretKey": "thisisasecretkeyanddontsharewithanyone"
  },
```

Рис. 7. Приклад секретного ключа для шифрування

```
var jwtSection = Configuration.GetSection("JWTSettings");
services.Configure<JWTSettings>(jwtSection);
```

Рис. 8. Опис сервісу аутентифікації

Додамо в контролер User JWT Settings через конструктор класу (рис.9).

```
public class UsersController : ControllerBase
{
    private readonly BookStoresDBContext _context;
    private readonly JWTSettings _jwtsettings;

    public UsersController(BookStoresDBContext context, IOptions<JWTSettings> jwtsettings)
    {
        _context = context;
        _jwtsettings = jwtsettings.Value;
    }
```

Рис. 9. JWT Settings

Зайдемо на сайт https://jwt.io/ та декодуємо токен (рис. 10) за допомогою секретного ключа (рис. 11).

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bmlxdWVfbmFtZSI6ImpvaG4uc21pdGhA
Z21haWwuY29tIiwibmJmIjoxNjE3MzQ5NzA3LCJleHAiOjE2MzMxNjA5MDcsImlhdCI6MTYxN
zM0OTcwN30.ZaxSe8TGvDRc1kX0nE6q490a6_v9A7XEKyu3DhOGniY
```

Рис. 10. Токен

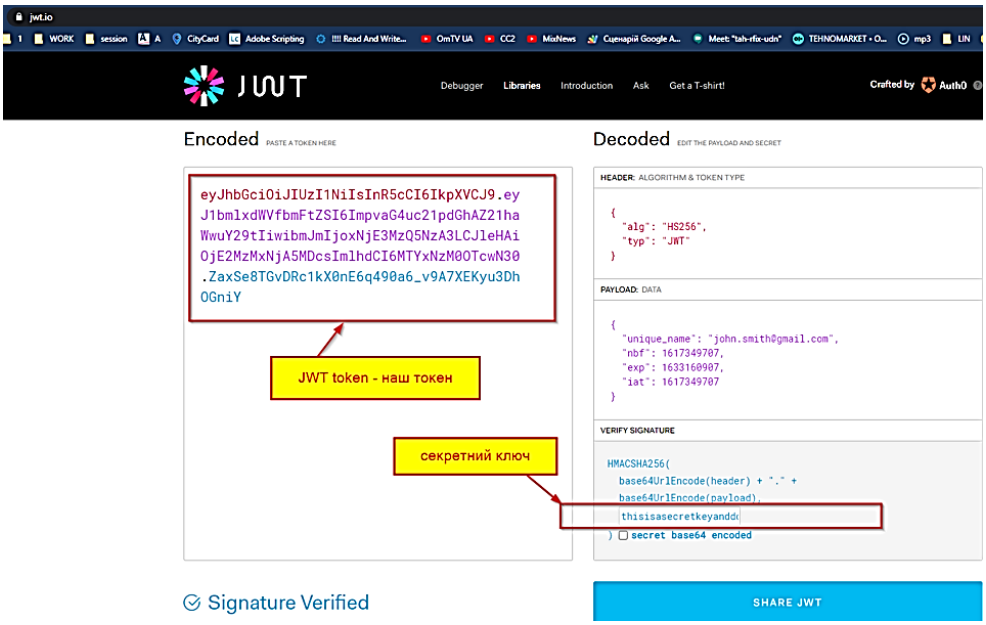

Рис. 11. Токен та секретний ключ

Після перевірки токена сервером ресурсів токен та надання доступу користувачеві, сервер зберігає токен у сховищі даних протягом визначеного користувачем часу сеансу, що може відрізнятися в залежності від типу програми чи вебдодатку. Так, для прикладу, час сеансу для банківських додатків визначено лише кілька хвилин.

Висновки. На сьогоднішній день із зростанням інноваційних технологій, правила забезпечення безпеки повинні ставати більш строгими, щоб надати гарантії доступу до певних ресурсів лише відповідним користувачам. Автентифікація має вирішальне значення з наступних причин:

- зручність, адже користувачі отримують доступ до більшої кількості програм та послуг на власних пристроях, у корпоративних мережах і в хмарних середовищах, їм потрібні ефективні та зручні методи автентифікації. Автентифікація тільки на основі пароля є непрактичною для користувачів, може бути легко скомпрометована;
- інтеграція сторонніх розробників, використання архітектури мікросервісів призвело до різкого збільшення кількості програмних систем, які підключаються одна до іншої в межах однієї організації. Захист механізму автентифікації необхідний для зручності роботи, запобігання випадковому розкриттю даних та захисту від кібератак;
- крадіжка облікових даних і записів, переважна більшість кібератак застосовує методи соціальної інженерії для захоплення облікових записів. Використання надійної автентифікації як для зовнішньої, так і для внутрішньої комунікації має важливе значення для відвертання сучасних кіберзагроз. Парадигма безпеки таких гігантів безпеки, як Microsoft, Google та AWS передбачає в основі безпечну автентифікацію [25].

Токени мають особливе значення у процесі забезпечення безпеки через їх здатність отримувати інформацію в зашифрованому вигляді для користування вебдодатком для підтримки та масштабування взаємодії з користувачем.

References

1. Norton LifeLock says thousands of customer accounts breached. URL: https://techcrunch.com/2023/01/15/norton-lifelock-password-manager-data/
2. Zdolbitska Nina, Bas Dmytro, Zhyharevych Oksana. AUTHORIZATION SERVER FOR A LOCAL NETWORK BASED ON TOKEN TECHNOLOGY. Abstracts of XVIII International Scientific and Practical Conference. Lisbon, Portugal. May 06-08, 2024. Pp. 252-253.
3. Authentication: Methods, Protocols, and Strategies. URL: https://frontegg.com/blog/authentication
4. Kazmi S.H.A., Hassan R., Qamar F., Nisar K., Ibrahim A.A.A. Security Concepts in Emerging 6G Communication: Threats, Countermeasures, Authentication Techniques and Research Directions. Symmetry 2023, 15, 1147p.
5. Ruihong Zhang, Zhihua Hu, Access control method of network security authentication information based on fuzzy reasoning algorithm, Measurement, Volume 185, 2021, 110103,
6. De Almeida, M.G., Canedo, E.D. Authentication and Authorization in Microservices Architecture: A Systematic Literature Review. Appl. Sci. 2022, 12, 3023 p.
7. OpenID Connect. URL: https://openid.net/foundation/
8. Krutika Patil. Authentication and Authorization in Web Applications. Journal of Engineering and Applied Sciences Technology Vol 5(1), 2023. p. 1-2
9. Using OAuth 2.0 for Server to Server Applications. URL: https://developers.google.com/identity/protocols/oauth2/service-account
10. Taher B.H., Liu H., Abedi F., Lu H., Yassin A.A., Mohammed A.J. A Secure and Lightweight Three-Factor Remote User Authentication Protocol for Future IoT Applications. J. Sensors, 2021, pp. 8871204:1-8871204:18.
11. El-Hajj M., Fadlallah A., Chamoun M., Serhrouchni A. A survey of internet of things (IoT) authentication schemes. Sensors 2019, 19, 1141 p.
12. Use these 6 user authentication types to secure networks. URL: https://www.techtarget.com/searchsecurity/tip/Use-these-6-user-authentication-types-to-secure-networks
13. OpenID Connect Authentication and OAuth 2.0 Authorization in Web Application. URL: https://siddhivinayak-sk.medium.com/openid-connect-authentication-and-oauth-2-0-authorization-in-web-application-e7e422eb5223