УДК 681.3

ПОБУДОВА БЕЗПЕЧНИХ ІНТЕРФЕЙСІВ ДЛЯ ВЗАЄМОДІЇ МІЖ МІКРОСЕРВІСАМИ У ХМАРНИХ СИСТЕМАХ

Яковів В.І.[1)], Пришляк О.В.[2)], Воробйов С.К.[3)]
Західноукраїнський національний університет
[1)]ст. викладач,[2)]аспірант, [3)]магістрант

I. Вступ

Впровадження мікросервісної архітектури у сучасні хмарні системи дозволило розробникам розділити складні застосунки на компактні функціональні модулі, що працюють незалежно один від одного[1,2]. Цей підхід покращує гнучкість розробки, прискорює розгортання нового функціоналу та полегшує масштабування. Однак, разом із зростанням популярності мікросервісної архітектури, постають важливі питання щодо безпеки цих розподілених систем [3-10].

Однією з ключових аспектів безпеки в мікросервісних додатках є забезпечення безпечної взаємодії між окремими сервісами через їхні інтерфейси [1,2,6]. Недостатньо лише захищати окремі сервіси; також потрібно забезпечити цілісність, конфіденційність та аутентифікацію даних, які передаються між сервісами. Для цього необхідно розробляти безпечні інтерфейси для взаємодії, які враховують специфіку мікросервісної архітектури.

II. Мета роботи

Метою роботи є дослідження кращих практик побудови безпечних інтерфейсів для взаємодії між мікросервісами у хмарних системах.

III. Особливості побудови безпечних інтерфейсів для взаємодії між мікросервісами

Мікросервісна архітектура надає безліч переваг у розробці, розгортці та масштабуванні додатків. Однак для забезпечення безпеки у взаємодії між мікросервісами виникають виклики, які потребують уваги. Один з ключових аспектів безпеки - це побудова безпечних інтерфейсів для взаємодії між мікросервісами.

Побудова безпечних інтерфейсів для взаємодії між мікросервісами вимагає врахування декількох важливих вимог [1,2]:

- Кожен мікросервіс повинен бути здатний перевіряти ідентифікацію та авторизацію інших мікросервісів та користувачів.
- Важливо забезпечити захист конфіденційності даних під час їх передачі між сервісами. Використання шифрування та безпечних каналів є обов'язковим.
- В разі виникнення помилок або відмови мікросервісу, інтерфейс повинен бути здатний обробити ці ситуації та відновити роботу системи в нормальний стан.

Враховуючи ці вимоги у роботі запропоновано метод автентифікації, який використовує JWT токен та протокол TOTP та дозволяє мікросервісам підтверджувати свою ідентичність перед іншими сервісами, що дає змогу забезпечити надійну автентифікацію.

Запропонований метод можна описати наступним етапами:

1. Налаштування параметрів:
   - Кожному мікросервісу та користувачу присвоюються унікальний секретний ключ для протоколу TOTP.
   - Сервер аутентифікації налаштовується для генерації JWT токенів та перевірки секретних ключів мікросервісів і користувачів.
2. Аутентифікація мікросервісу:
   - Коли мікросервіс намагається здійснити доступ до іншого мікросервісу або ресурсу, то генерується запит на аутентифікаціюу який включається ідентифікатор мікросервісу та час.
3. Створення JWT токена:
   - Сервер аутентифікації створює JWT токен з інформацією про ідентифікатор мікросервісу,його права доступу, алгоритм шифрування та час життя токена.
4. Включення TOTP одноразового пароля:

- Мікросервіс генерує одноразовий пароль з використанням свого секретного ключа TOTP, який додається до JWT токена або запиту на аутентифікацію.

5. Надсилання запиту на аутентифікацію:

- Запит на аутентифікацію, який містить JWT токен та TOTP одноразовий пароль, надсилається до сервера, який потребує аутентифікації.

6. Перевірка JWT токена:

- Сервер перевіряє цілісність автентичність та час життя JWT токена за допомогою секретного ключа сервера аутентифікації.

7. Перевірка TOTP одноразового пароля:

- Сервер перевіряє, чи введений одноразовий пароль збігається з тим, який був згенерований на стороні мікросервісу.
- Якщо паролі збігаються, а JWT токенвалідний, то запит на аутентифікацію вважається успішним.

8. Відповідь сервера:

- Сервер відповідає на запит мікросервісу, надаючи доступ до ресурсу або функціональності.

9. Завершення сеансу:

- Після завершення взаємодії, сервер може вимагати повторної автентифікації для майбутніх запитів.

Цей метод дозволяє забезпечити безпеку взаємодії між мікросервісами у мікросервісній архітектурі, використовуючи комбінацію JWT токенів для ідентифікації та TOTP одноразових паролів для додаткового захисту.

Висновок

У роботі розглянуто важливі аспекти побудови безпечних інтерфейсів для взаємодії між мікросервісами в мікросервісній архітектурі. Мікросервіси надають численні переваги у розробці, розгортці та масштабуванні додатків, але забезпечення безпеки в їхній взаємодії вимагає ретельного планування та впровадження заходів.

Важливо підкреслити, що побудова безпечних інтерфейсів для взаємодії між мікросервісами є критично важливою для забезпечення безпеки та надійності мікросервісної архітектури. Для цього запропоновано метод автентифікації, який використовує JWT токени та протокол TOTP. Цей метод дозволяє мікросервісам підтверджувати свою ідентичність перед іншими сервісами, забезпечуючи надійну автентифікацію.

Реалізація запропонованого методу може сприяти покращенню безпеки та довіри у взаємодії між компонентами мікросервісних систем, роблячи їх більш стійкими до потенційних загроз.

Список використаних джерел

1. Hannousse, A.; Yahiouche, S. Securing microservices and microservice architectures: A systematic mapping study. Comput. Sci. Rev. 2021, 41, 100415.
2. Washizaki, H.; Xia, T.; Kamata, N.; Fukazawa, Y.; Kanuka, H.; Kato, T.; Yoshino, M.; Okubo, T.; Ogata, S.; Kaiya, H.; et al. Systematic Literature Review of Security Pattern Research. Information 2021, 12, 36.
3. Pinheiro, D.; Oliveira, J.; Figuereido, E. Microservice Smells and Automated Detection Tools: A Systematic Literature Review. In Proceedings of the 4th International Conference on Microservices, (Microservices 2022), Paris, France, 10–12 May 2022
4. Zhou, X.; Wu, X.; Chen, Y.; Deng, D. High-Concurrency and High-Performance Application of Microservice Order System Based on Big Data. Sec. Commun. Netw. 2022, 2022, 3424283.
5. Lyu, C.; Zhang, X.; Liu, Z.; Chi, C.H. Selective Authentication Based Geographic Opportunistic Routing in Wireless Sensor Networks for Internet of Things Against DoS Attacks. IEEE Access 2019, 7, 31068–31082.
6. Zhang, P.; Wang, Y.; Kumar, N.; Jiang, C.; Shi, G. A Security- and Privacy-Preserving Approach Based on Data Disturbance for Collaborative Edge Computing in Social IoT Systems. IEEE Trans. Comput. Soc. Syst. 2022, 9, 97–108.
7. Комар М.П. Нейросетевой подход к обнаружению сетевых атак на компьютерные системы / М.П. Комар, И.О. Палий, Р.П. Шевчук, Т.Б. Федысив // Інформатика та математичні методи в моделюванні. – 2011. – Т. 1, №2. – С. 156–160.
8. V. Cheshun, I. Muliar, V. Yatskiv, R. Shevchuk, S. Kulyna, and T. Tsavolyk, “Safe decentralized applications development using blockchain technologies,” in Proceedings of the 10th International Conference on Advanced Computer Information Technologies, pp. 800–805, Deggendorf, Germany, September 2020.
9. O. Kovalchuk, M. Karpinski, S. Banakh, M. Kasianchuk, R. Shevchuk and N. Zagorodna, "PredictionMachineLearningModelsonPropensityConvictstoCriminalRecidivism", Information, vol. 14, no. 3, pp. 161, 2023.
10. O. Kovalchuk, M. Kasianchuk, M. Karpinskiand R. Shevchuk, "Decision-MakingSupportingModelsConcerningtheInternalSecurityoftheState", INTL Journal of Electronics Telecommunications, vol. 69, no. 2, pp. 301-307, 2023.