Трокоз Є.М., ст. викладач,
Сірик А.Ю., магістрант
Державний університет «Житомирська політехніка»

ЗАСТОСУВАННЯ БЕЗПЕЧНИХ API ДЛЯ ОБМІНУ ДАНИМИ: МЕТОДИ ТА ТЕХНОЛОГІЇ

В сучасному світі, де інформаційні технології визначають нові стандарти розвитку, обмін даними між програмами та сервісами виявляється ключовим аспектом розробки програмного забезпечення. Процес обміну даними потребує не лише ефективності та швидкості, але й високого рівня безпеки.

API, або інтерфейс програмування додатків, є механізмом, що визначає взаємодію між різними програмами чи сервісами. Це забезпечує ефективну комунікацію та обмін даними між різними компонентами програмного забезпечення. Серед різних архітектур API можна виділити RESTful, SOAP, GraphQL, WebSocket та Open API (Swagger).

RESTful API (Representational State Transfer) орієнтований на ресурси. Кожен ресурс має свій унікальний URI. В RESTful API використовуються стандартні методи HTTP. Також він спрощує роботу з даними та підтримує швидку та ефективну взаємодію. SOAP API (Simple Object Access Protocol) заснований на обміні повідомленнями у форматі XML. Використовується для обміну структурованими та типізованими даними. Підтримує більш жорсткі стандарти в порівнянні з RESTful і частіше використовується у великих корпоративних системах. GraphQL API дозволяє клієнтам запитувати саме ті дані, які їм потрібні, замість отримання всього об'єму даних. Визначає структуру запитів та формат повернених даних на стороні клієнта. Також GraphQL API забезпечує більш гнучку та ефективну взаємодію для клієнтів. WebSocket API забезпечує двосторонню взаємодію між клієнтом та сервером в режимі реального часу. Використовує постійне підключення для передачі даних в обидві сторони. WebSocket API ідеальний для відстеження подій, онлайн-чатів та інших сценаріїв, де необхідна миттєва взаємодія. Open API (Swagger) визначає стандарти та специфікації для документації та розробки API. Забезпечує чітку специфікацію для структури та взаємодії з API. Крім того, він дозволяє автоматично генерувати код для взаємодії з API на різних мовах програмування [1].

Один із важливих аспектів безпеки в обміні даними – це використання протоколу HTTPS. Він забезпечує захищену передачу даних за допомогою шифрування, зменшуючи ризик перехоплення чи модифікації інформації. Однією з ключових функцій HTTPS є використання шифрування для захисту даних під час їхньої передачі. Для цього використовуються криптографічні протоколи, такі як TLS (Transport Layer Security) або його попередник SSL (Secure Sockets Layer). Сертифікати видаються авторитетними сертифікаційними центрами та підтверджують ідентичність сервера. Шифрування унеможливлює зловмисникам читання чи модифікацію даних, які передаються між клієнтом та сервером. HTTPS використовує різні криптографічні алгоритми для забезпечення конфіденційності та цілісності даних. Сайти, які використовують протокол HTTPS, можуть отримати позитивний вплив на їхній рейтинг у пошукових системах. Крім того, браузери можуть позначати незахищені сайти як небезпечні, тоді як захищені сайти відзначаються символом замку чи іншими позначками безпеки.

OAuth 2.0 є важливим інструментом для забезпечення безпеки під час обміну авторизаційними даними між різними системами. Він дозволяє надавати обмежений доступ до ресурсів без передачі облікових даних. Токени в OAuth 2.0 використовуються для ідентифікації та авторизації користувачів під час доступу до ресурсів. Вони використовуються для забезпечення безпеки обміну даними та відокремлення від передачі справжніх облікових даних.

Застосування безпеки на рівні даних. Шифрування та хешування грають важливу роль у забезпеченні безпеки обміну даними через API. Шифрування забезпечує конфіденційність, а хешування допомагає перевірити цілісність отриманих даних [2, 3].

Методи та технології обміну даними з використанням безпечних API є визначальним елементом сучасного програмування. Вони забезпечують не тільки ефективну комунікацію, а й гарантують безпеку обміну даними відповідно до найвищих стандартів. Розуміння та використання цих методів та технологій стає важливим завданням для розробників у всіх сферах індустрії.

Список використаних джерел

1. Jacobson, Daniel. APIs: A Strategy Guide. O'Reilly Media, 2012. – 148 p.
2. API Security. URL: https://www.imperva.com/learn/application-security/api-security
3. What is API Security? URL: https://nonamesecurity.com/learn/api-security/