

УДК 004.056.53

Ребриков М.М.¹, Куцак С.В.²

¹студ. гр. РТ-818 НУ «Запорізька політехніка»

²старш. викл. НУ «Запорізька політехніка»

БЕЗПЕЧНЕ ЗБЕРІГАННЯ ТОКЕНА АВТОРИЗАЦІЇ (JWT) НА КЛІЄНТСЬКІЙ СТОРОНІ ВЕБДОДАТКУ

Зі стрімким зростанням кількості мережевих сервісів, таких як соціальні мережі, онлайн-банкінг, сервіси, що забезпечують хмарне зберігання інформації та кругообіг документів у електронному форматі, питання авторизації виникає практично в кожному з них. Для авторизації в сучасних вебдодатках часто використовують JSON Web Token [1] (далі JWT), який офіційно з'явився та широко використовується з 2015 року.

JWT - це відкритий стандарт (RFC 7519 [2]) для створення токенів доступу, які, як правило, використовуються для передачі даних аутентифікації в клієнт-серверних додатках. Токени створюються сервером, підписуються алгоритмом HMAC (з використанням секретного ключа) або алгоритмами шифрування, такими як RSA або ECDSA (з використанням пари відкритого і закритого ключів) і передаються на клієнтську сторону додатку. В подальшому користувач використовує даний токен для підтвердження своєї особи.

В даній роботі розглядається питання безпечного зберігання JWT на клієнтській стороні додатку. Існує два основні методи зберігання токенів в пам'яті браузера: в локальному сховищі і в файлах cookie [3]. Також є третій спосіб (зберігання токена в змінній JavaScript), але він практично не використовується, так як зломисник може без труднощів звернутися до цієї змінної з використанням JavaScript і заволодіти токеном. Також, використовуючи цей метод, не є можливим передавати токен між двома або більше відкритими вкладками браузера. Тому в даній роботі розглядаються тільки двох перших методів зберігання токена, а саме: в локальному сховищі браузера і в файлах cookie.

Переваги зберігання JSON Web токена в локальному сховищі браузера:

- якщо в API (програмний інтерфейс додатку) є необхідність передавати токен авторизації саме в заголовок запиту, то зручно використовувати локальне сховище, тому що ним можна керувати за допомогою JavaScript;

- якщо в API, який використовується в додатку, немає можливості додати функціонал встановлення cookie файлів, то єдиним місцем зберігання токена є локальне сховище.

Недоліки зберігання JSON Web токена в локальному сховищі браузера:

- локальне сховище браузера вразливе до атак типу "Впровадження коду", а саме "XSS-атака" (Cross-Site Scripting). Джерелом такої атаки може бути скрипт сторонньої JavaScript бібліотеки, підключеної до вашого вебдодатку.

Переваги зберігання JSON Web токена в файлах cookie:

- при кожному запиті клієнтської сторони додатку до сервера, cookie будуть відправлятися автоматично. Це є перевагою лише в тому випадку, якщо API додатку не вимагає знаходження токена авторизації в заголовку запиту. В іншому випадку неможливо зберігати токен авторизації в файлах cookie;

- якщо при встановленні cookie файлів вказати прапор "HttpOnly", то зломисник не зможе отримати доступ до токена авторизації використовуючи JavaScript.

Недоліки зберігання JSON Web токена в файлах cookie:

- не є можливим зберігати багато даних в токені авторизації, так як розмір файлу cookie обмежений 4Кб.

- файли cookie схильні до атак типу "Міжсайтова підробка запиту" (CSRF-атака), але існує спосіб знизити можливість реалізації такої атаки.

Для реалізації захищеного зберігання токена авторизації в файлах cookie необхідно пам'ятати про наступні правила:

- на сайті повинен бути встановлений SSL (Secure Sockets Layer) – захищений сертифікат;
- файли cookie повинні мати прапор "HttpOnly = true". Це унеможливорює "XSS-атаку";
- файли cookie повинні мати прапор "SameSite = Strict". Це унеможливорює "CSRF-атаку";
- файли cookie повинні мати прапор "Secure = true". При наявності цього прапора, файли cookie будуть передаватися тільки по захищеному каналу зв'язку HTTPS.

Використання JWT (токена авторизації) у файлах cookie на даний момент вважається найбезпечнішим способом зберігання даних на стороні клієнта вебдодатку. Цей спосіб найбільш захищений від можливої реалізації атак злоумисника на дані користувача, але має і недоліки, які потрібно враховувати під час реалізації.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. JSON Web Tokens are an open, industry standard RFC 7519 method for representing claims securely between two parties [Електронний ресурс]. – Режим доступу: <https://jwt.io/>
2. RFC 7519 (JSON Web Token) [Електронний ресурс]. – Режим доступу: <https://tools.ietf.org/html/rfc7519>
3. Using HTTP cookies [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies>