БЕЗОПАСНОЕ ХРАНЕНИЕ JWT-ТОКЕНА АВТОРИЗАЦИИ В WEB-ПРИЛОЖЕНИЯХ

Т.Е. Козляк

Развитие методов авторизации в web-приложениях породило создания нового стандарта авторизации, основанного на формате JSON, позволяющий создавать JSON Web Token (JWT) токены доступа. JWT – это открытый стандарт для создания токенов доступа, основанный на формате JSON. Токены создаются сервером, подписываются секретным ключом и передаются клиенту, который в дальнейшем использует токен для подтверждения своей личности. При использования данного типа токенов возникает вопрос о том, как безопасно хранить токены в публичной части web-приложениях.

В работе рассматриваются основные способы хранения JWT-токена, его виды и способы применения при разработке web-приложений. JWT-токены бывают двух видов. 1. Токены доступа (access-token). Для авторизации запросов и предоставляет

доступ его владельцу к защищенным ресурсам сервера. Имеют короткий срок жизни и может нести в себе дополнительную информацию (например, такую как IP-адрес стороны, запрашивающей данный токен). 2. Токены обновления (refresh-token). Для получения нового токена доступа при истечении срока действия предыдущего токена.

Основные способы хранения JWT Access-токена и проблемы безопасности, которые приходится для них решать, следующие. 1. Local Storage / Session Storage (локальное браузерное хранилище). Преимущества заключаются в его юзабилити, т. к. вся работа с хранилищем происходит довольно просто и на чистом JavaScript. Подвержено XSS-атакам, если подключаются сторонние скрипты, которые могут получить доступ к локальному хранилищу. 2. Cookies (небольшой фрагмент данных, до 4Кб). Преимущества заключаются в более гибкой настройке. Простое хранения Access токена в cookie допускает атаки типа CSRF и XSS. Для защиты, можно воспользоваться параметром Cookie SameSite в режиме Strict, который поможет добиться защиты от CSRF-атаки, путем сокрытия ваших cookie при обращении к api других сайтов. Также есть возможность защититься от XSS-атак путем использования флага httpOnly, а добавление флага Secure поможет защититься от Сниффинга (Sniffer).

Хранение токена в Local Storage было использовано автором при разработке веб-приложения для проведения соревнований по программированию искусственного интеллекта «AI Cup Battle», проводимых кафедрой системного программирования и компьютерной безопасности ГрГУ им. Янки Купалы с 2021 г.