УДК 004.223.2

ИСПОЛЬЗОВАНИЕ ПРОТОКОЛА АВТОРИЗАЦИИ OAUTH 2.0 В ПОЛЬЗОВАТЕЛЬСКОМ ПРИЛОЖЕНИИ

А.В. СУББОТИН
(Представлено: канд. физ.-мат. наук., доц. Ю.Ф. ПАСТУХОВ)

В статье рассматривается возможность использования в качестве протокола авторизации протокол OAUTH 2.0.

Ключевые слова: *информационные технологии, авторизация, шифрование, аутентификация.*

OAuth 2.0 – протокол авторизации, позволяющий выдать одному сервису (приложению) права на доступ к ресурсам пользователя на другом сервисе. Протокол избавляет от необходимости доверять приложению логин и пароль, а также позволяет выдавать ограниченный набор прав, а не все сразу.

Принцип работы. Как и первая версия, OAuth 2.0 основан на использовании базовых веб-технологий: HTTP-запросах, редиректах и т. п. Поэтому использование OAuth возможно на любой платформе с доступом к интернету и браузеру: на сайтах, в мобильных и desktop-приложениях, плагинах для браузеров.

Ключевое отличие от OAuth 1.0 – простота. В новой версии нет громоздких схем подписи, сокращено количество запросов, необходимых для авторизации.

Общая схема работы приложения, использующего OAuth, такова:

– получение авторизации;
– обращение к защищенным ресурсам.

Результатом авторизации является access token – некий ключ (обычно просто набор символов), предъявление которого является пропуском к защищенным ресурсам. Обращение к ним в самом простом случае происходит по HTTPS с указанием в заголовках или в качестве одного из параметров полученного access token'a.

В протоколе описано несколько вариантов авторизации, подходящих для различных ситуаций:

– авторизация для приложений, имеющих серверную часть (чаще всего, это сайты и веб-приложения);
– авторизация для полностью клиентских приложений (мобильные и desktop-приложения);
– авторизация по логину и паролю;
– восстановление предыдущей авторизации.

Авторизация для приложений, имеющих серверную часть. Редирект на страницу авторизации:

– На странице авторизации у пользователя запрашивается подтверждение выдачи прав.
– В случае согласия пользователя, браузер редиректится на URL, указанный при открытии страницы авторизации, с добавлением в GET-параметры специального ключа – authorization code.
– Сервер приложения выполняет POST-запрос с полученным authorization code в качестве параметра. В результате этого запроса возвращается access token (рисунок 1).

Это самый сложный вариант авторизации, но только он позволяет сервису однозначно установить приложение, обращающееся за авторизацией (это происходит при коммуникации между серверами на последнем шаге). Во всех остальных вариантах авторизация происходит полностью на клиенте и по понятным причинам возможна маскировка одного приложения под другое. Это стоит учитывать при внедрении OAuth-аутентификации в API сервисов.

Авторизация полностью клиентских приложений.

– Открытие встроенного браузера со страницей авторизации.
– У пользователя запрашивается подтверждение выдачи прав.
– В случае согласия пользователя, браузер редиректится на страницу-заглушку во фрагменте (после #) URL которой добавляется access token.
– Приложение перехватывает редирект и получает access token из адреса страницы (рисунок 2).

Этот вариант требует поднятия в приложении окна браузера, но не требует серверной части и дополнительного вызова сервер-сервер для обмена authorization code на access token.

Авторизация по логину и паролю. Авторизация по логину и паролю представляет простой POST-запрос, в результате которого возвращается *access token*. Такая схема не представляет из себя ничего нового, но вставлена в стандарт для общности и рекомендуется к применению только, когда другие варианты авторизации не доступны.

Восстановление предыдущей авторизации. Обычно, *access token* имеет ограниченный срок годности. Это может быть полезно, например, если он передается по открытым каналам. Чтобы не заставлять

пользователя проходить авторизацию после истечения срока действия *access token*'а, во всех перечисленных выше вариантах, в дополнение к *access token*'у может возвращаться еще *refresh token*. По нему можно получить *access token* с помощью HTTP-запроса, аналогично авторизации по логину и паролю.

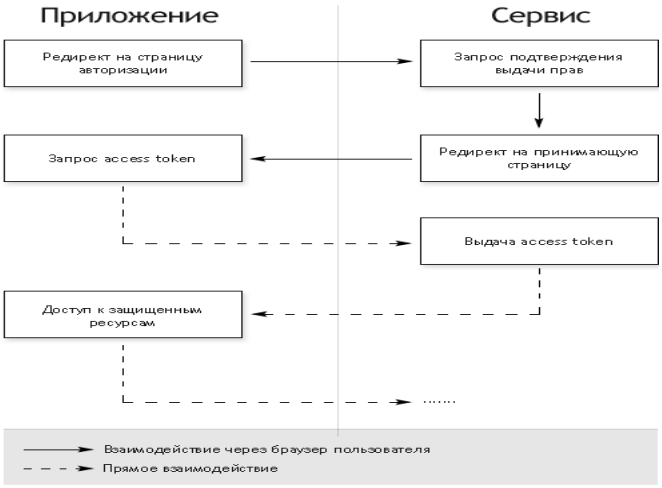

Рисунок 1. – Авторизация для приложений, имеющих серверную часть

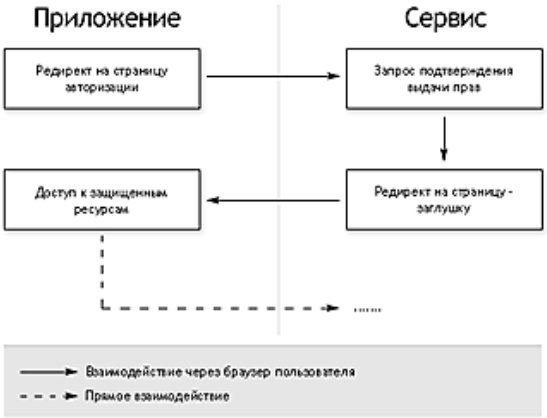

Рисунок 2. – Авторизация полностью клиентских приложений

Заключение. OAuth – простой стандарт авторизации, основанный на базовых принципах интернета, что делает возможным применение авторизации практически на любой платформе. Стандарт имеет поддержку крупнейших площадок и очевидно, что его популярность будет только расти.

ЛИТЕРАТУРА

1. Материал из Википедии – свободной энциклопедии. OAuth [Электронный ресурс]. Режим доступа: https://ru.wikipedia.org/wiki/OAuth – Дата доступа: 23.09.2018.
2. OAuth 2.0 [Электронный ресурс]. Режим доступа: https://oauth.net/2/ – Дата доступа: 23.09.2018.