Список литературы

1. *Синадский Н. И.*, *Сушков П. В.* Модификация методов анализа социальных графов на основе применения атрибутивных компонентов учетных записей для идентификации сообществ пользователей социальных сетей // Вестн. УрФО. Безопасность в информационной сфере. 2017. № 2 (23). С. 32–40.

2. *Сушков П. В.*, *Синадский Н. И.* Модифицированный метод оценки частичного изоморфизма социальных графов // XV Всерос. науч.-практ. конф. студентов, аспирантов и молодых ученых : сб. материалов. Курган : Курган. ГУ, 2016. С. 207–210.

3. *Безуглая М. В.*, *Патрушева О. М.*, *Синадский Н. И.*, *Сушков П. В.* Расчет показателя сходства учетных записей пользователей социальных сетей на основе анализа атрибутов и структуры социальных связей // Безопасность информационного пространства : сб. тр. XIV Всерос. науч.-практ. конф. студентов, аспирантов и молодых ученых / сост. А. А. Захаров. Тюмень : Изд-во Тюмен. гос. ун-та, 2016. С. 19–23.

4. Модель Барабаши — Альберт [Электронный ресурс]. Режим доступа: https://ru.wikipedia.org/wiki/Модель_ Барабаши_ —_Альберт.

УДК (004.056.53)

Д. А. Лазуков
Научный руководитель: канд. физ.-мат. наук, доц. О. В. Ниссенбаум
Тюменский государственный университет, Тюмень

ОБЗОР МЕТОДОВ АУТЕНТИФИКАЦИИ В REST WEB API

Аннотация. В статье обозреваются методы аутентификации в REST API и дается краткое представление о принципах их работы.

Ключевые слова: аутентификация; REST API; Web; контроль доступа.

На сегодняшний день множество web-сервисов предоставляют доступ не только через web-интерфейс, десктоп или мобильное приложение, но и через Web API (Application Programming Interface). Существует множество способов реализации API, например, SOAP, XML-RPC и т. д., однако самым популярным на данный момент является REST. REST (Representational State Transfer — передача состояния представления) — стиль архитектуры для разработки распределенных систем в сети [1]. Сервисы, которые следуют принципам REST, называют RESTful. REST не привязан к какому-либо протоколу передачи данных и не является стандартизированным: он является архитектурным стилем,

и разработчики вольны отступать от него в зависимости от требований разрабатываемой системы, однако из-за того, что он получил широкое распространение в web-сфере, большинство REST сервисов используют стандарты HTTP, URL, XML и JSON.

В связи с тем что RESTful сервисы доступны в Сети (в частности REST WEB API), остро встает вопрос контроля доступа к ресурсам сервиса. Для этого были разработаны следующие протоколы аутентификации пользователей, позволяющие реализовывать защиту веб-сервисов от несанкционированного доступа.

HTTP Basic аутентификация [2]. Является самым простым протоколом аутентификации. В нем используется специальный HTTP заголовок Authorization, значение которого равно комбинации логина пользователя и пароля закодированной в base64 формате. В итоге заголовок выглядит следующим образом: «Authorization: Basic base64("username: password")». Очевидным плюсом данного метода является простота и очевидность работы. Но хотя комбинация логина и пароля закодированы, они не являются зашифрованными. Такой вид аутентификации может быть использован только с протоколом HTTPS.

HMAC (Digest) аутентификация [2]. Большим минусом Basic аутентификации является то, что нужно отправлять пароль пользователя в каждом запросе. Также метод не защищен от подделки заголовка или тела запроса, уязвим к атаке повторного воспроизведения.

Решением этой проблемы является использование HMAC (Hash Based Message Authentication). Суть метода заключается в том, что каждый запрос пользователя хэшируется на его секретном ключе. Сформированный хэш называют дайджестом (digest). После формируется HTTP заголовок вида «Authentication: hmac имя_пользователя: base64(дайджест)». Дайджест формируется в зависимости от требований безопасности системы: он может содержать в себе название HTTP метода, URL запрашиваемого ресурса, хэш тела запроса. Если требуется дополнительная защита от подделки запроса или атаки повторного воспроизведения дайджест может включать в себя nonce (одноразовый код) или метку времени. После того как дайджест сформирован и запрос отправлен, сервер, зная имя пользователя и его секрет формирует дайджест на своей стороне по тому же принципу, что и пользователь. И если дайджесты сервера и пользователя равны, то запрос считается аутентичным.

OAuth 1.0 и OAuth 2.0. OAuth — открытый протокол, позволяющий предоставить доступ третьей стороне к защищенным ресурсам пользователя без передачи пользователем пароля третьей стороне. На данный момент самый популярный метод предоставления доступа к API, который используют почти все крупные веб-сервисы (VK, Facebook, Telegram, Twitter).

Хотя и протоколы не имеют обратной совместимости, общий принцип работы у них похож. Для разъяснения принципа их работы введем понятия пользователя *U*, веб-сервиса *A*, веб-сервиса *B*, хранящего защищенные данные пользователя и предоставляющий доступ к ним по API. Общий упрощенный механизм работы заключается в следующем:

1. Сервис *A* запрашивает разрешение *U* на доступ к его информации, хранящейся на сервисе *B*.

2. *U* переходит на сервис *B*, где подтверждает свое разрешение на использование информации сервисом *A*.

3. Сервис *B* передает сервису *A* токен доступа к данным пользователя *U* [3].

Зачастую токен содержит в себе информацию об уровне доступа и полномочиях, которыми наделен сервис *A* (JSON Web Token, JWT). Также возможно хранение прав доступа токена на стороне сервиса *B*.

OAuth 2.0 является не доработкой Oauth 1.0, а полным его переосмыслением и содержит в себе следующие кардинальные отличия:

1. Гораздо проще в использовании и внедрении.

2. Безопасность основана на SSL.

3. Предоставляет удобные сценарии доступа не только для веб, а также для десктоп и мобильных приложений.

4. Токен имеет более короткий срок жизни; вводится понятие refresh token.

Токен часто помещается в заголовок Authorization: bearer <token>, но также встречаются реализации, когда он помещается в тело запроса.

Таким образом в статье были рассмотрены самые распространенные методы аутентификации на сегодняшний день. Выбор метода в первую очередь зависит от разрабатываемого проекта или инфраструктуры и требований к их безопасности. Общей рекомендацией в соответствии с текущими тенденциями является использование протокола OAuth 2.0, если разрабатываемый продукт подразумевает обращение сторонних приложений к данным пользователя, в целях более простого и надежного управления и разграничения доступа. В ином случае стоит использовать BASIC или Digest аутентификацию в связи с более простой реализацией.

Список литературы

1. Архитектура REST // Хабрахабр. URL: https://habrahabr.ru/post/38730/ (дата обращения: 05.11.2017).

2. HTTP Authentication: Basic and Digest Access Authentication // The Internet Engineering Task Force. URL: https://tools.ietf.org/html/rfc2617 (дата обращения: 05.11.2017).

3. The OAuth 2.0 Authorization Framework // The Internet Engineering Task Force. URL: https://tools.ietf.org/html/rfc6749 (дата обращения: 05.11.2017).