УДК 004.056.2
DOI 10.52575/2687-0932-2023-50-1-144-151

Способы улучшения защищённости сервисов, использующих JWT-токены

Девицына С.Н., Пилькевич П.В., Удод Е.В.
Севастопольский государственный университет,
Россия, 299053, г. Севастополь, ул. Университетская, д. 33
E-mail: sndevitsyna@sevsu.ru pavel.piksel2012@mail.ru elena.udod03@gmail.com

Аннотация. Целью работы является определение наиболее эффективного и универсального механизма авторизации, соответствующего требованиям безопасности и защищённости информации пользователя в современном цифровом обществе. В результате были исследованы основные недостатки существующих механизмов авторизации, их уязвимые места. С помощью моделирования и дальнейшего исследования пользовательских приложений были установлены наиболее подходящие технологии обеспечения эффективной авторизации на основе Json Web Token (JWT). Предполагается не отказываться от технологии использования JWT, а, наоборот, начать её рассматривать как одну из самых подходящих и безопасных технологий на сегодняшний день. Проанализированы актуальные методы реализации данной технологии и сделаны выводы о необходимости дополнительного их улучшения из-за существующих недостатков в виде перехвата данных. В дальнейшем планируется в компьютерной киберлаборатории исследовать уровень безопасности систем авторизации на основе предложенной технологии JWT в реальных условиях.

Ключевые слова: информационные технологии, информационная безопасность, авторизация, улучшение технологии авторизации, JWT

Для цитирования: Девицына С.Н., Пилькевич П.В., Удод Е.В. 2023. Способы улучшения защищённости сервисов, использующих JWT-токены. Экономика. Информатика, 50(1): 144–151. DOI 10.52575/2687-0932-2023-50-1-144-151

Ways to Improve the Security of Services Using JWT Tokens

Svetlana N. Devitsyna, Pavel V. Pilkevich, Elena V. Udod
Sevastopol State University,
33 University St, Sevastopol, 299053, Russia
E-mail: sndevitsyna@sevsu.ru pavel.piksel2012@mail.ru elena.udod03@gmail.com

Abstract. The aim of the work is to determine the most effective and universal authorization mechanism that meets the requirements for the safety and security of user information in today's digital society. As a result, the main shortcomings of the existing authorization mechanisms, their vulnerabilities were investigated. Through modeling and further research of user applications, the most suitable technologies for providing effective authorization based on Json Web Token (JWT) were established. It is supposed not to abandon the technology of using JWT, but, on the contrary, to begin to consider it as one of the most suitable and secure technologies today. The current methods for implementing this technology are analyzed and conclusions are drawn about the need for additional improvement due to existing shortcomings in the form of data interception. In the future, it is planned to study the level of security of authorization systems based on the proposed JWT technology in real conditions in a computer cyber laboratory.

Keywords: information technology, information security, authorization, improvement of authorization technology, JWT tokens

For citation: Devitsyna S.N., Pilkevich P.V., Udod E.V. 2023. Ways to Improve the Security of Services Using JWT Tokens. Economics. Information technologies, 50(1): 144–151. (in Russian). DOI 10.52575/2687-0932-2023-50-1-144-151

Введение

Современный цифровой мир предлагает людям широкие возможности благодаря развитию глобальной сети Интернет и появлению новых сервисов и приложений. Все большее количество людей переходит в интернет-пространство для работы и общения. Большой объем информации, быстродействующие жизненные процессы и растущий спрос на общие знания заставляют людей постоянно искать надежные и исчерпывающие источники информации, которые Интернет может им предоставить. Использование сервисов и приложений требует регистрации и авторизации, в этих условиях возникает всё больше проблем с защитой конфиденциальной информации и прежде всего сведений о личности пользователя, его учетной записи.

В цифровом пространстве преимущество получают порталы и сервисы, которые подстраиваются и поддерживают интересы пользователя. Это необходимо в первую очередь для удобства и упрощения однотипных задач в сети. Например, в ежедневную рутину могут входить ввод пароля для доступа к какой-либо платформе, поиск различных товаров и просмотр интересующего контента. Все это просто реализуется и подстраивается под конкретного человека на основе анализа системой предпочтений и персональных данных пользователя, то есть его «следа» в сети — «цифрового следа». Адаптация предлагаемого пользователю контента основывается на том, что интересовало пользователя ранее. Несмотря на безусловное удобство, такой механизм может привести и к негативным последствиям с точки зрения защиты личных данных человека. Таким образом, объектом данного исследования являются механизмы авторизации пользователей в веб-приложениях, а предметом исследования — методы улучшения технологии авторизации для большей защиты информации о пользователе в цифровом пространстве.

Основная часть

С увеличением спроса со стороны пользователей на разнообразные приложения, и повсеместном использовании сервисов, работающих с персональными данными, возникает необходимость обеспечения защищённости процесса авторизации со стороны создателей данных сервисов, и возрастает интерес к безопасным технологиям авторизации.

При этом необходимо решать сразу несколько задач:

– определять алгоритм авторизации пользователей на сервисе;

– выбирать способы взаимодействия пользователя с сервисом;

– решать проблемы безопасного хранения данных авторизации пользователя и оптимизации ресурсов, выделенных на их хранение;

– анализировать возможные исходы при утрате человеком основного ключа, используемого при авторизации и идентификации на используемом им сервисе.

Для обеспечения защиты от несанкционированного доступа к ресурсу и, соответственно, персональным данным пользователя, может быть использована самая распространенная – парольная защита. Пароль используется для ограничения доступа к информации тех пользователей, кто не имеет на это право. Однако важным замечанием является то, что использование стандартных паролей может приводить сразу к нескольким критическим для информационной безопасности последствиям. Главным из таких последствий является получение безграничного доступа злоумышленника к аккаунту взломанного пользователя

в случае раскрытия пароля. Чтобы избежать подобного исхода, необходимо использование дополнительной процедуры проверки прав на выполнение определенных действий в системе. В качестве такой процедуры принято использовать дополнительные методы авторизации, такие как, например, отправка уведомлений на дополнительные надёжные сервисы пользователя. То есть будет несколько раз проверено, что пользователь действительно является тем, за кого себя выдает. Но такой поэтапный вид авторизации значительно усложняет процесс взаимодействия с сервисами и не способен защитить от ситуаций, когда пользователь использует один и тот же пароль для всех своих учётных данных, что зачастую практикуется многими. При таком раскладе можно будет лишиться сразу большого количества защищаемой информации.

В качестве более эффективного и безопасного метода передачи информации от источника к получателю можно использовать технологию JSON Web Token (JWT) [Колесников, 2021]. Такой токен создаётся по стандарту RFC 7519 и вмещает в себя всю необходимую информацию для идентификации любого пользователя. Допустим, пользователь испытывает необходимость регистрации на сайте. В этом случае существует три участника процесса. Первый участник — это непосредственно пользователь, выполняющий действие, второй участник — сервер приложения, где происходит регистрация, а третьим является сервер аутентификации. В дальнейшем он будет выполнять роль обеспечения пользователя тем самым токеном, с помощью которого будет происходить его взаимодействие с приложением [Голубь, 2020; Бетелин, 2021].

Специальный алгоритм взаимодействия с токеном позволяет повысить безопасность обмена информацией с сервисом и предотвратить некоторые из попыток злоумышленников получить доступ к данным пользователя [Волосенков, Лупарев, 2015].

В процессе работы предполагается взаимодействие с тремя частями токена: заголовок, полезная нагрузка и подпись. Заголовок — это краткое описание, содержащее в себе необходимую техническую информацию для обработки токена. По структуре написания заголовок должен включать в себя обязательный ключ, характеризующий алгоритм шифрования, который используется при работе токена. Также заголовок может содержать необязательный ключ. Он может отражать тип токена или тип его содержимого. Вторая часть токена — это полезная нагрузка, непосредственно данные, передаваемые токеном. Нагрузка содержит общедоступную информацию, передаваемую между пользователем и сервисом. Например, это может быть имя пользователя и уровень его доступа [Беликов и др., 2021]. Также в структуру полезной нагрузки могут входить некоторые служебные ключи. Они являются необязательным элементом нагрузки, но имеют возможность передавать дополнительную информацию о токене.

Рассмотрим, какими бывают служебные ключи нагрузки:

1) iss: чувствительная к регистру строка или URI, которая является уникальным идентификатором стороны, генерирующей токен (issuer);

2) sub: чувствительная к регистру строка или URI, которая является уникальным идентификатором стороны, о которой содержится информация в данном токене. Значения с этим ключом должны быть уникальны в контексте стороны, генерирующей JWT;

3) aud: массив чувствительных к регистру строк или URI, являющийся списком получателей данного токена;

4) exp: время в формате Unix Time, определяющее момент, когда токен станет невалидным;

5) nbf: в противоположность ключу exp, это время в формате Unix Time, определяющее момент, когда токен станет валидным;

6) jti: строка, определяющая уникальный идентификатор данного токена;

7) iat: время в формате Unix Time, определяющее момент, когда токен был создан.

Третий элемент — это подпись, которая проверяет корректность токена. Подпись является вычисляемой и основывается на двух предыдущих составляющих токена. Алгоритм

соединяет закодированные строки через точку. Затем полученная строка хешируется алгоритмом, заданным в заголовке на основе секретного ключа. Объединив все составляющие, получим токен, который можно передавать.

В процессе взаимодействия пользователя и системы предполагается использование двух токенов: access и refresh. Идентифицирующий токен (access) необходим для обеспечения доступа к защищаемым данным и имеет определённое время действия. Для продления его действия нужно использовать refresh. Этот токен выдается на длительный период.

Секретный ключ знают только сервер приложения и сервер аутентификации. Сформированные access и refresh токены отправляются пользователю после его запроса и применяются при авторизации. При попытке входа сервер проверяет на соответствие идентифицирующие токены и выдает результат.

Почему необходимо использовать два токена? Такой подход обезопасит ситуацию, когда злоумышленник получает доступ к идентифицирующему токену. Исследуя его характеристики, стало известно, что через определенное время токен потеряет жизнеспособность и использовать его для авторизации не удастся. В этом случае поможет токен refresh и произойдет обновление данных для авторизации [Кольчугин и др., 2021].

Кроме того, необходимо задуматься о выборе метода получения и хранения пользователем данных токенов. Существует два способа их хранения. Токены могут находиться в локальном хранилище. Но важно помнить, что при атаке на сервис, злоумышленник может с легкостью получить защищаемые данные. Чтобы обезопасить пользователя от подобных неприятностей, был предложен еще один метод хранения токенов. Этот метод основывается на использовании cookie. В них будут помещены токены, а безопасность будет обеспечена в следствие того, что cookie отправляются пользователю после его запроса, а не хранятся в базе постоянно. Таким образом, злоумышленнику будет сложнее получить к ним доступ.

Для повышения безопасности использования JWT-токенов, рассмотрим структуру заголовка HTTP-пакета (рис. 1) [Золотов, 2015]:

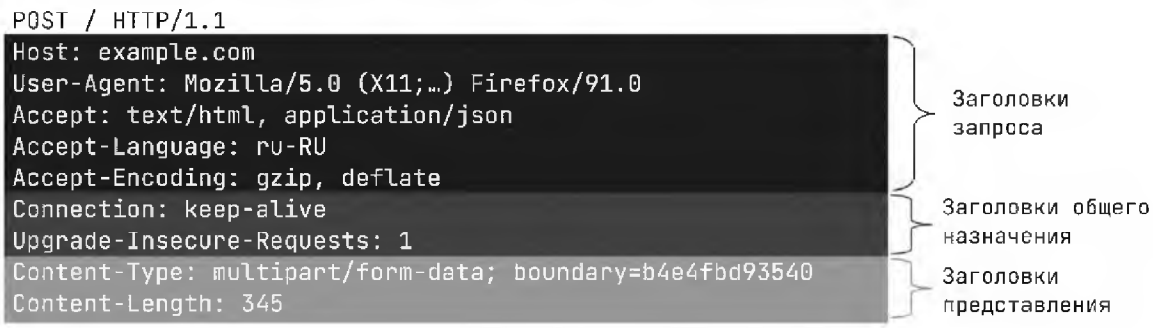

Рис. 1. Структура HTTP-заголовка
Fig. 1. Structure of HTTP headers

Следует отметить, что каждый пакет, приходящий на сервер, будет содержать в себе некоторую техническую информацию об источнике, который посылает сообщения на сервер. Не стоит забывать и об информации, которая содержится на более низких уровнях модели OSI, такая, например, как IP-адрес источника [Изюмов, 2005]. Собрав воедино всю эту информацию, так как она приходит с каждым пакетом в соединении, приложение сможет быть уверено, что каждая отправка сообщений будет сопровождаться данной вспомогательной информацией. Таким образом, можно сформировать из неё некоторый цифровой отпечаток пользователя, который будет пытаться пройти процесс авторизации [Кришнамурти и др., 2002]. Простейшим методом для выполнения описанной операции будет объединение всех строк в заранее определённой последовательности и проведение операции хеширования. Так можно убедиться, что источник access-токена не изменился даже

на уровне приложения (веб-браузер) [Ткаченко и др., 2021]. Это позволяет достичь дополнительного уровня защищенности и безопасности в случаях, когда злоумышленнику всё же удалось заполучить доступ к access-токену пользователя. В таком случае при попытке отправить токен с другого устройства будет выдана ошибка, сигнализирующая о том, что отпечатки не совпадают и требуется дополнительно ввести данные от запрашиваемой в токене учётной записи [Мельникова, 2016; Московченко и др., 2018; Даянов, 2020].

На данный момент подобные решения уже используются и в качестве отпечатка хранится информация о User-Agent пользователя (рис. 1). Но некоторые злоумышленники всё ещё могут получить доступ к User-Agent и подделать таким образом данные для входа в систему. Следует заметить, что вероятность этого крайне мала, и такие действия выполняются, как правило, с целью не просто обычного интернет-мошенничества, а когда совершается целевая атака на определённого пользователя. Тем не менее, такая угроза всё ещё остаётся, хотя и сведена к минимуму. В рамках данной работы предлагается применять максимально широкий спектр полей, сканируемых для формирования цифрового отпечатка пользователя. Очевиден вывод, что при добавлении данного решения приложение будет терять небольшую долю скорости вычислений, хотя данная потеря будет совсем незначительной, но зато мы полностью исключим вероятность подмены токена со стороны злоумышленника [Подшибякина, 2020; Попов и др., 2022; Бессонова и др., 2012].

Обязательно при этом предусмотреть способ и место хранения на серверном устройстве данных о таких цифровых отпечатках пользователей [Мурадян, 2020.]. Также следует учесть оптимизацию памяти на хранящем токены устройстве. Для этого требуется, во-первых, составить верную структуру хранящихся данных для токенов. Очевидно, что в состав структуры входит сам токен (refresh), его цифровой отпечаток, а также ссылка на пользователя, к которому указанный токен относится (рис. 2).

```
type RefreshToken struct {
        ID        string    `json:"id" bson:"_id,omitempty"`
        UUID      string    `json:"uuid" bson:"uuid"`
        UserID    string    `json:"userID" bson:"userID"`
        ExpiresAt time.Time `json:"expiresAt" bson:"expiresAt"`
}
```

Рис. 2. Структура таблицы для хранения сессий
Fig. 2. Structure of table that contains user's sessions

Также следует заметить, что хранить данные о цифровом отпечатке в непреобразованном виде не является хорошим решением, так как в случае вероятных атак на базу данных, некоторые важные данные можно будет извлечь из предполагаемых таблиц. Вместо этого необходимо захешировать входящие данные сразу и хранить их именно в преобразованном виде, тем самым снизив уровень критичности утечки данных из нашей базы. Также следует определиться с выбором СУБД, которая будет использоваться для хранения и операций над данными токенов. Учитывая небольшие объёмы данных, которые будут храниться, а также весьма малую несложность запросов, которые будут посылаться СУБД, можно использовать решения, которые не используют обращение к себе по SQL (NoSQL базы данных) [Тарасов, 2015]. Базы данных NoSQL специально созданы для определенных моделей данных и обладают гибкими схемами, что позволяет разрабатывать современные приложения. Базы данных NoSQL получили широкое распространение в связи с простотой разработки, функциональностью и производительностью при любых масштабах [Крамаренко и др., 2016]. Базы данных NoSQL используют разнообразные модели данных для доступа к данным и управления ими. Базы данных таких типов оптимизированы для приложений, которые ра-

ботают с большим объемом данных, нуждаются в низкой задержке и гибких моделях данных. Все это достигается путем смягчения жестких требований к непротиворечивости данных, характерных для других типов БД. Например, одним из эффективных и удобных решений может стать СУБД MongoDB [Муравьев и др., 2015].

Заключение

В результате исследования проанализированы основные проблемы, связанные с вопросами авторизации пользователя на сервисах, описаны недостатки других методов авторизации по сравнению с JWT-токенами, показаны особенности работы с данными токенами. Было определено, что для повышения безопасности взаимодействия с JWT-токеном можно использовать техническую информацию о пользовательском устройстве, с которого отправляют токен на этапе анализа access-токена. Это предотвратит случаи их перехвата злоумышленником.

В дальнейшем планируется провести натурные испытания предложенной методики в компьютерной киберлаборатории и исследовать уровень безопасности систем авторизации на основе предложенной технологии JWT в реальных условиях.

Список литературы

Беликов Г.В., Крылов И.Д., Селищев В.А. 2021. SQL-инъекция как способ обхода авторизации. Известия Тульского государственного университета. Технические науки. 12: 217–221.

Бетелин А.Б. и др. 2021. О некоторых особенностях JWT аутентификации в веб-приложениях. Труды научно-исследовательского института системных исследований Российской академии наук. 11(1): 4–10.

Бессонова Е.Е. и др. 2012. Способ идентификации пользователя в сети Интернет. Научно-технический вестник информационных технологий, механики и оптики. 3 (79): 133–137.

Волосенков В.О., Лупорев С.Н. 2015. Способы обеспечения информационной безопасности веб-приложений. Проблемы безопасности российского общества. 1: 80–85.

Голубь И.С. 2020. Приложение Net Core Web API с использованием JWT-токенов для авторизации. Постулат. [Электронный ресурс]. Режим доступа: https://elibrary.ru/item.asp?id=42763413 (Дата обращения: 10.01.2023).

Даянов А.М. 2020. Особенности использования технологии «фингерпринтинг». Мавлютовские чтения: 14–15.

Золотов А.А. 2015. Протокол HTTP, его расширения и надстройки. Молодежный научно-технический вестник. 2: 9–10.

Изюмов А.Е. 2005. Исследование безопасности протокола http. Научно-технический вестник информационных технологий, механики и оптики. 19: 161–166.

Колесников А.О. 2021. Идентификация пользователей клиент-серверных приложений с помощью JWT-токена. Сборник статей XXXVI международной научно-практической конференции: 42–43. [Электронный ресурс]. Режим доступа: https://elibrary.ru/item.asp?id=45675397 (Дата обращения: 10.01.2023).

Кольчурин М.Л. и др. 2021. Рекомендации по реализации безопасной авторизации в условиях микросервисной архитектуры. Физика для Пермского края: 205–210.

Крамаренко Т.А., Деменков И.А., Михеев А.М. 2016. Выбор клиент-серверной СУБД для реализации информационной системы. Современные информационные технологии. 24: 11–15.

Кришнамурти Б., Рексфорд Д. 2002. Web-протоколы. Теория и практика. БИНОМ: 58-61.

Мельникова Т.В. 2016. Некоторые особенности работы бизнес-аналитика в IT-сфере. Моделирование, оптимизация и информационные технологии. 1: 5.

Московченко В.М. и др. 2018. Анализ технологий защиты от идентификации веб-браузеров. NBI-technologies. 12(1): 34–39.

Муравьев С., Дворянкин С., Насенков И. 2015. СУБД: проблема выбора. Открытые системы. СУБД. 1: 22–24.

Мурадян А.Х. 2020. Классификация способов детектирования и блокирования автоматизированного сбора данных с веб-ресурсов. Альманах научных работ молодых учёных Университета ИТМО: 134–138.

Подшибякина В.В. 2020. Цифровой отпечаток браузера. Информационные технологии в процессе подготовки современного специалиста: 122–126.

Попов А.Ю. и др. 2022. Парсинг электронных ресурсов. Библиотека Selenium или fake useragent.

Тарасов С.В. 2015. СУБД для программиста. Базы данных изнутри.

Ткаченко А.Л., Сафронов Е.С., Кузнецова В.И. 2021. Анализ эффективности защиты персональных данных и проблема cookie файлов. Дневник науки. 6.

References

Belikov G.V., Krylov I.D., Selishchev V.A. 2021. SQL injection as a way to bypass authorization. News of the Tula State University. Technical science. 12: 217–221.

Betelin A.B. et al. 2021. On some features of JWT authentication in web applications. Proceedings of the Research Institute for System Research of the Russian Academy of Sciences. 11(1): 4–10.

Bessonova E.E. et al. 2012. Method of user identification in the Internet // Scientific and technical bulletin of information technologies, mechanics and optics. 3 (79): 133–137.

Volosenkov V.O., Luporev S.N. 2015. Ways to ensure information security of web applications. Problems of security of the Russian society. 1: 80–85.

Golub I.S. 2020. Application Net Core Web API using JWT tokens for authorization. Postulate. [Antlion]. Available at: https://elibrary.ru/item.asp?id=42763413 (accessed 10.01.2023). (in Russian)

Dayanov A.M. 2020. Features of using the fingerprinting technology. Mavlyutov readings: 14–15.

Zolotov A.A. 2015. HTTP protocol, its extensions and add-ons. Youth Scientific and Technical Bulletin. 2: 9–10.

Izyumov A.E. 2005. Research on the security of the http protocol. Scientific and technical bulletin of information technologies, mechanics and optics. 19: 161–166.

Kolesnikov A.O. 2021. Identification of users of client-server applications using JWT-token: 42–43. [Antlion]. Available at: https://elibrary.ru/item.asp?id=45675397 (accessed 10.01.2023). (in Russian)

Kolchurin M.L. et al. 2021. Recommendations for the implementation of safe authorization in the conditions of microservice architecture. Physics for the Perm region: 205–210.

Kramarenko T.A., Demenkov I.A., Mikheev A.M. 2016. Choosing a client-server DBMS for the implementation of an information system. Modern information technologies. 24: 11–15.

Krishnamurti B., Rexford D. 2002. Web protocols. Theory and practice. BINOM: 58-61.

Melnikova T.V. 2016. Some features of the work of a business analyst in the IT sphere. Modeling, optimization and information technologies. 1: 5.

Moskovchenko V.M. et al. 2018. Analysis of technologies for protection against identification of web browsers //NBI-technologies. 12(1): 34–39.

Muravyov S., Dvoryankin S., Nasenkov I. 2015. DBMS: the problem of choice //Open Systems. DBMS. 1: 22–24.

Muradyan A.Kh. 2020. Classification of methods for detecting and blocking automated data collection from web resources // Almanac of Scientific Works of Young Scientists of ITMO University: 134–138.

Podshibyakina V.V. 2020. Digital browser imprint. Information technologies in the process of training a modern specialist: 122–126.

Popov A.Yu. et al. 2022. Parsing of electronic resources. Selenium library or fake useragent.

Tarasov S.V. 2015. DBMS for the programmer. Databases from the inside.

Tkachenko A.L., Safronov E.S., Kuznetsova V.I. 2021. Analysis of the effectiveness of personal data protection and the problem of cookies. Diary of Science. 6.

Конфликт интересов: о потенциальном конфликте интересов не сообщалось.
Conflict of interest: no potential conflict of interest related to this article was reported.

ИНФОРМАЦИЯ ОБ АВТОРАХ

Девицына Светлана Николаевна, кандидат технических наук, доцент, доцент кафедры «Информационная безопасность» института информационных технологий Севастопольского Государственного университета, г. Севастополь, Россия

Пилькевич Павел Вадимович, студент кафедры «Информационная безопасность» института информационных технологий Севастопольского Государственного университета, г. Севастополь, Россия

Удод Елена Витальевна, студент кафедры «Информационная безопасность» института информационных технологий Севастопольского Государственного университета, г. Севастополь, Россия

INFORMATION ABOUT THE AUTHORS

Svetlana N. Devitsyna, Candidate of Technical Sciences, Associate Professor, Associate Professor of the Department "Information Security", Institute of Information Technologies, Sevastopol State University, Sevastopol, Russia

Pavel V. Pilkevich, student of the Department "Information Security", Institute of Information Technologies, Sevastopol State University, Sevastopol, Russia

Elena V. Udod, student of the Department "Information Security", Institute of Information Technologies, Sevastopol State University, Sevastopol, Russia