УДК 004.056.53 Вестник УрФО № 2(52) / 2024, с. 77–88

Басалай К. А., Зулькарнеев И. Р. DOI: 10.14529/secur240208

ПОДХОДЫ К КЛАССИФИКАЦИИ ВЕКТОРОВ АТАК И УЯЗВИМОСТЕЙ JSON WEB TOKENS[1]

В статье рассмотрены вопросы обеспечения безопасности JSON Web Tokens при их использовании в микросервисной архитектуре, мобильных и веб-приложениях. Обозначены преимущества использования JWT по сравнению с другими методами аутентификации и авторизации. Авторами поднята проблема реализации системного и комплексного подхода по обеспечению безопасности при использовании JWT и отсутствия какой-либо классификации уязвимостей и векторов атак на JWT в отечественной и зарубежной научно-технической литературе.

В ходе работы был исследован полный жизненный цикл JWT и их структура, используемые стандартные поля и алгоритмы формирования подписи, была составлена классификация уязвимостей на основе этапов жизненного цикла токена, и для каждого класса уязвимостей определены вектора атак. Были проанализированы существующие 119 уязвимостей базы CVE, связанных с JWT, определены новые категории и построены корреляции по данным с 2015г. На основе этого анализа авторами была предложена классификация векторов атак на основе жизненного цикла JWT и объектов атаки, а также выявлены новые уязвимости, которые не рассмотрены в научных работах на данный момент.

Данная классификация может быть использована при проектировании информационных систем на базе микросервисной архитектуры для обеспечения безопасности процессов аутентификации и авторизации на стороне клиента и сервера, определения слабых мест работающих сервисов, а также подбора инструментария для тестирования безопасности JWT.

Ключевые слова: *JWT, CVE, классификация уязвимостей, вектор атаки, аутентификация, авторизация, микросервисы.*

[1] Работа выполнена в рамках гранта РФФИ 20-47-720005

APPROACHES TO CLASSIFYING ATTACK VECTORS AND VULNERA-BILITIES OF JSON WEB TOKENS

Security issues of JSON web tokens (JWT) usage in a microservice architec-ture, in mobile and web applications are presented in the paper. It also outlines the benefits of JWT using over other authentication and authorization methods. The au-thors raise the problem of implementing a systematic and comprehensive approach to security assurance while using JWT. They also highlight the lack of national and international scientific and technical resources classifying JWT vulnerabilities and attack vectors.

The entire lifecycle and structural components of JSON Web Tokens, as well as the standard claims and algorithms used for signature generation were investigat-ed. In addition, the paper presents a classification of vulnerabilities based on JWT lifecycle stages, and defines attack vectors for each vulnerability class. The authors analyzed all existing 119 CVE vulnerabilities associated with JWTs, identified new categories and built correlations using data since 2015. Through the analysis the au-thors proposed the attack vectors classification based on the JWT lifecycle and attack targets, and identified new vulnerabilities that have not been considered in previous researches.

There are several ways to use the developed classification: to ensure the security of authentication and authorization processes on the client and server sides in the information systems design based on microservice architecture; to find vulnerabilities in the services; to select tools for testing the security of JWT.

Keywords: *JWT, CVE, vulnerability classification, attack vector, authentica-tion, authorization, microservices.*

Введение

Мобильные или веб-приложения являются неотъемлемой частью функционирования любой организации, стартапа или государственного учреждения. Крупные организации с большим количеством внутренних и внешних пользователей стараются создавать свои собственные информационные экосистемы, используя в том числе веб-сервисы и внутренние вычислительные ресурсы. Согласно исследованию [1] большинство организаций используют для этих целей микросервисную архитектуру, отмечая ее гибкость, масштабируемость, быстроту внесения изменений и удобство для разработчиков. Необходимость доступа к постоянно растущему числу сервисов большого количества пользователей поставило задачу использования быстрой, удобной и ресурсоемкой системы аутентификации, в том числе сквозной для всей экосистемы. Одним из самых распространенных решений данной проблемы является использование JSON Web Tokens (далее – JWT) [2]. Спектр применения этих токенов настолько высок, что они используются даже для управления доступом и обеспечения безопасности устройств IoT [3], что может быть применено при построении и управлении доступом, например, в медицинских информационных системах, клинико-диагностических лабораториях и телемедицине.

Популярность использования JWT обусловлена следующими преимуществами перед остальными методами аутентификации и авторизации:

Удобство – использование токенов упрощает доступ пользователей к большому количеству сервисов за счет сквозной аутентификации [4].

Автоматическая проверка пользователя – при каждом запросе от поль-зователя на сервер, его JWT помещается в заголовок http-пакета, что позволяет автоматически пройти аутентификацию [5].

Повышение скорости обработки запросов – JWT содержит в себе все необходимые пользовательские данные и передает их в открытом виде, снижая нагрузку на сервер [2, 5].

Гибкость – разработчики могут создавать собственные поля внутри токена, самостоятельно определять их содержимое и вариант применения [2].

Наряду с очевидными преимуществами JWT остро встает вопрос об обеспечении безопасности при их использовании, так как особенности структуры и методов реализации JSON Web Tokens служат причиной большого количества потенциальных уязвимостей, что приводит к появлению новых векторов атак на систему аутентификации и ее компрометации.

Целью данной статьи является проведение комплексного анализа существующих проблем JWT и составление классификации уязвимостей и векторов атак на основе их жизненного цикла. Классификация позволит сформировать целостное представление о возможных атаках на протяжении всех этапов использования JWT, что позволит обеспечить безопасность процессов аутентификации и авторизации; рассмотреть слабые места работающих сервисов, а также подобрать инструментарий для тестирования безопасности JWT.

Обзор предыдущих работ

Коллектив авторов статьи [5] описывает механизмы аутентификации и авторизации на основе JWT, рассматривает состав токена, выделяет базовые уязвимости JWT и методы их устранения, а также отмечает преимущества и недостатки использования JWT. Статья носит преимущественно обзорный характер и предложенные авторами способы обеспечения безопасности JWT охватывают лишь малую часть известных уязвимостей.

Авторы научной работы [4] сравнили механизмы аутентификации и авторизации на основе JWT и на основе сессий. Для обеспечения безопасности системы аутентификации авторы предлагают реализовать защиту от XSS-атак на стороне клиента, а также применение сложных механизмов валидации и использование сложных ключей шифрования на стороне сервера. Указанные ме-ры носят точечный характер и охватывают только некоторые атаки на JWT.

В публикации А. Булгаковой [6] рассмотрены проблемы хранения и ис-пользования JWT. Авторы пришли к выводу о том, что хранение токенов в cookies и Local Storage небезопасно, поскольку в этом случае клиент может быть подвергнут атакам типа XSS и CSRF. Был предложен подход с использованием refresh-токенов для обеспечения безопасности пользователей. Но в то же время в статье игнорируются возможные уязвимости на стороне сервера при обработке JWT, а также атаки на сам токен.

Похожая проблема была исследована коллективом ученых [7] на специально разработанном стенде. По итогам эксперимента авторы предлагают ис-пользовать HTTP Only Cookie для предотвращения CSRF-атак на пользователя. Подобная мера позволяет нейтрализовать данную угрозу, но для обеспечения комплексной безопасности системы аутентификации требуется предотвратить атаки не только на клиента, но и на сервер и сам токен.

В работе С.Н. Девициной [8] был проведен анализ проблем, связанных с вопросами аутентификации пользователей на сервисах, продемонстрированы особенности аутентификации и авторизации на основе JWT. Авторами предлагается использование технической информации о клиенте для повышения безопасности взаимодействия с его токеном, что должно предотвратить случаи перехвата JWT злоумышленником. Но для комплексного подхода к данной проблеме требуется убедиться в том, что производится корректная проверка подлинности токена на сервере, и что JWT не содержит иных уязвимостей, позволяющих обойти предложенный авторами подход.

О.Р. Никитин и А.Г. Уймин [9] в своей работе провели тестирование JWT на безопасность, реализовали атаки с использованием алгоритма «none» и удалением сигнатуры. Авторы подчеркивают, что базовой защиты JWT недостаточно для его использования в приложениях и в качестве методов защиты предлагают маркирование трафика на стороне клиента, а также применение VPN для организации защищенного канала передачи токена до сервера.

Вышеупомянутые научные работы рассматривают лишь некоторые аспекты, связанные с безопасностью JWT. Применение

каждой из предложенных мер по отдельности не может обеспечить безопасность системы аутентификации и авторизации. Для комплексного решения данной проблемы и обеспечения безопасности всего жизненного цикла JWT необходимо изучить, каким образом клиент и сервер взаимодействуют с токеном и на каких этапах этого взаимодействия могут возникнуть уязвимости.

Формулировка проблемы

Несмотря на ряд преимуществ, при использовании JWT возникают про-блемы, связанные с безопасностью. Чаще всего в научной и технической литературе описываются уязвимости, связанные с атаками на клиента [6, 7] и на алгоритмы подписи токена [4, 5, 9]. Это ограничивает область выбора защитных мер при обеспечении безопасности JWT из-за малого числа исследуемых уязвимостей и отсутствия полной картины потенциальных атак. В качестве возможных решений данной проблемы мы видим систематизацию существующих уязвимостей и построение классификации векторов атак на JSON Web Tokens, позволяющей установить взаимосвязь между этапами жизненного цикла токена и уязвимостями. Использование классификации позволит, во-первых, на этапе проектирования информационной системы понять слабые места процесса аутентификации и авторизации и ликвидировать их, во-вторых, выявлять уязвимости уже работающих сервисов и своевременно принимать меры по их нейтрализации, в-третьих, с помощью предложенной классификации возможно подобрать инструментарий для комплексного тестирования безопасности JWT на всех этапах его использования.

Жизненный цикл JSON Web Tokens

На рисунке 1 изображена схема взаимодействия пользователя с приложением с использованием JWT. За выпуск токенов отвечает сервис аутентификации (далее – SAuth), в качестве которого может выступать как специально созданный сервис, так и отдельный модуль в рамках существующих сервисов. Процедура проверки подлинности токенов осуществляется без участия SAuth, валидация происходит при их обработке на стороне целевых сервисов.

Для понимания того, каким образом появляются уязвимости при использовании JWT, рассмотрим процессы аутентификации и авторизации на основе токенов. Для этого опишем жизненный цикл JWT, который состоит из нескольких этапов.

Этап 0. Инициация доступа: пользователь должен подтвердить, что он тот, за кого себя выдает, посылая запрос на целевой сервис. Обработка запроса осуществляется посредством SAuth. Данный этап инициирует жизненный цикл JWT.

Этап 1. Получение токена: в случае, если предыдущий этап прошел успешно, SAuth генерирует для пользователя токен. На этом этапе в JWT формируются поля полезной на-

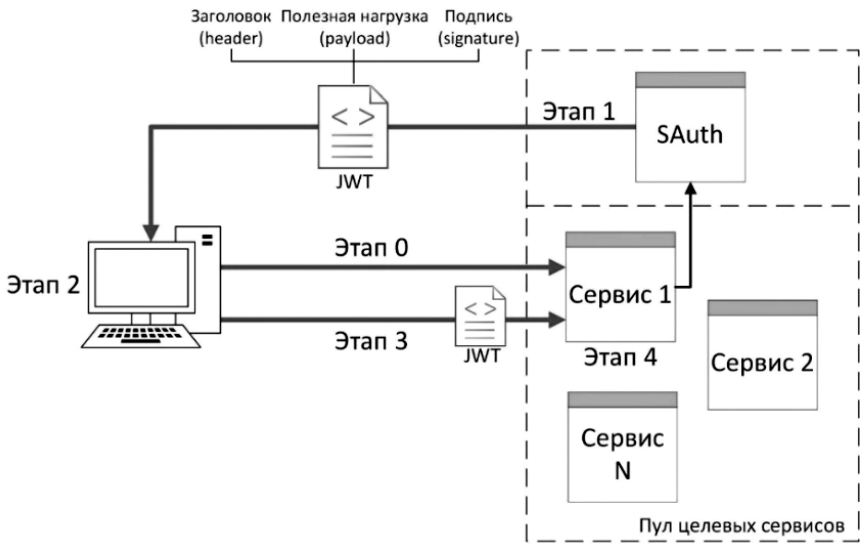

Рис. 1. Описание жизненного цикла JSON Web Token

грузки, в которых, как правило, содержится служебная информация, а также роль и права пользователя в данном приложении. Количество полей токена и их содержание может быть произвольным и определяется администратором системы. Сгенерированный токен может использоваться с целью автоматической сквозной аутентификации и авторизации для всех сервисов, а не только для целевого. После того, как JWT был сформирован, он отправляется клиенту. В случае, если пользователь изначально ввел некорректные данные, возвращается соответствующая ошибка.

Этап 2. Сохранение токена: после получения JWT сохраняется браузе-ром клиента в локальных хранилищах (например, cookies или Local Storage) и используется при дальнейшей аутентификации.

Этап 3. Запрос на аутентификацию: при каждом клиентском обращении к приложению JWT из хранилища автоматически помещается в заголовок http-запроса. При этом должна быть обеспечена безопасная среда передачи токена, поскольку содержащаяся в JWT информация передается в открытом виде.

Этап 4. Обработка токена: при получении JWT сервисом запускается процесс его валидации (за счет проверки подписи токена), параметры которого определяются администратором системы. В случае, если JWT не проходит проверку сервиса, пользователю придется заново пройти этап инициации доступа. Если же проверка пройдена успешно, то сервис, обрабатывая поля токена, производит авторизацию клиента и предоставляет доступ к запрашиваемому ресурсу в соответствии с ролью пользователя в приложении.

Структура JWT

Для понимания природы возможных уязвимостей на всех этапах жизненного цикла токена отдельного рассмотрения требует структура JWT.

JSON Web Token представляет из себя средство авторизации, аутентификации и безопасной передачи информации в формате JSON между двумя сторонами. Структура, описание, варианты использования и формирования JWT определены в стандарте RFC 7519 JSON Web Token (JWT) [10]. Согласно стандарту, токен состоит из 3-х частей: заголовка, полезной нагрузки и подписи. Каждая из частей кодируется алгоритмом Base64 и разделяется друг от друга точкой.

Заголовок (*header*) является первой частью JWT. В нем, как правило, со-держится информация о типе токена (поле «*typ*») и используемом алгоритме подписи (поле «*alg*») [5]. Но поскольку стандарт RFC 7519 носит лишь реко-мендательный характер, то в заголовок можно включать и иную информацию по желанию администраторов системы. Как правило, значение поля «typ» равно «JWT». Возможные для использования в JWT криптографические алгоритмы подписи описаны в стандарте RFC 7518 JSON Web Algorithms (JWA). Стандартом рекомендованы несколько из них: ES256 и RS256, но существует и множество других доступных для использования алгоритмов [11].

Полезная нагрузка (*payload*) является следующей частью JWT, в которой содержится служебная информация, а также необходимые данные для аутентификации и авторизации пользователя. Обычно в данной части токена находятся такие сведения, как уникальный идентификатор пользователя, его имя и роль в приложении (пользователь, администратор и т.п.). Также существуют определенные в стандарте [10] поля, которые предназначены для предоставления служебной информации, например, времени жизни токена, информации о выпускающем токен сервисе и о получателе данного JWT.

Подпись (*signature*) – третья часть JWT, которая обеспечивает безопас-ность процедур аутентификации и авторизации в приложении. Подпись формируется следующим образом: сначала нужно закодировать с помощью алгоритма Base64 поля заголовка и полезной нагрузки, разделив их между собой точкой. Затем полученная строка хэшируется с помощью алгоритма, заданного в заголовке JWT и с использованием секретного ключа. Полученная строка также кодируется с помощью Base64 и помещается после части полезной нагрузки в подпись.

В результате анализа видно, что структура JWT задана не строго, и администратор системы может по своему усмотрению создавать новые поля или удалять старые в любой части токена.

Для определения и классификации векторов атак на JWT необходимо исследовать актуальные и потенциальные уязвимости, опираясь на структуру токена и этапы его жизненного цикла.

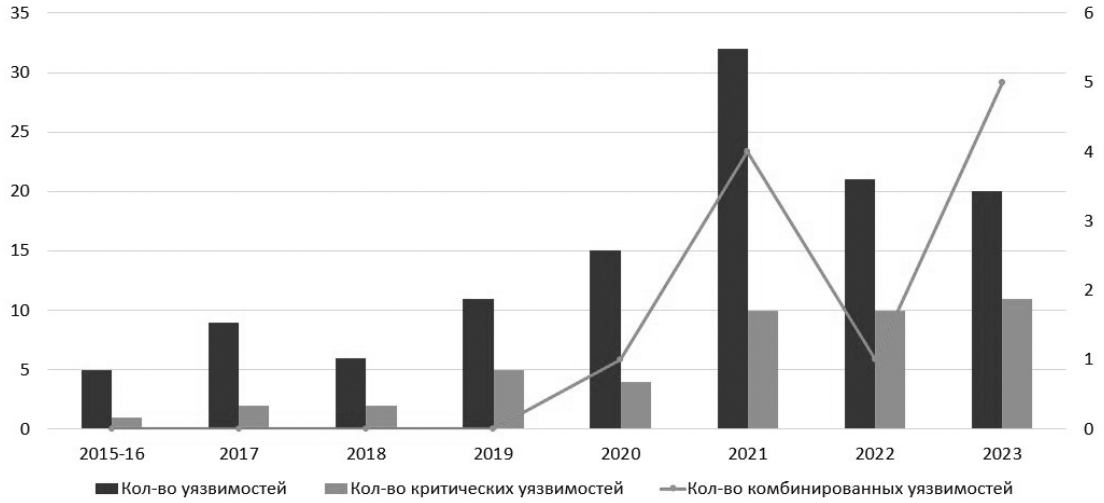

Рис. 2. Диаграмма уязвимостей JWT

Анализ существующих уязвимостей JWT

В ходе исследования нами было проанализировано 119 существующих на момент написания статьи уязвимостей базы данных CVE, которые напрямую связаны с процессами аутентификации на основе JSON Web Tokens. На рисунке 2 представлена динамика появления новых уязвимостей (синий столбец), связанных с JWT, с 2015 по 2023 гг.

Начиная с 2020 г. растет количество критических уязвимостей токенов (серый столбец), т.е. имеющих рейтинг 9.0 и выше по CVSS 3.0, растет и их отношение к общему ежегодному числу новых уязвимостей. При этом с 2021 г. наблюдается снижение количества новых уязвимостей с рейтингом «высокий» (от 7.0. до 8.9) и «средний» (от 4.0. до 6.9) по классификации CVSS 3.0. Оба этих фактора формируют положительную динамику роста среднего значения рейтинга всех новых уязвимостей в год: от 7.7 в 2020 г. до 8.5 в 2023 г.

Все существующие в базе CVE уязвимости JWT можно условно поделить на несколько категорий в зависимости от способов эксплуатации и их локализации:

- уязвимости проверки полей токена;
- уязвимости, связанные с формированием и проверкой подписи JWT;
- уязвимости конфигурации сервера.

Стоит отметить, что с 2021 г. появляются новые категории уязвимостей токенов, которых раньше не было или которые были представлены в единичном экземпляре в 2020 г.:

- уязвимости серверной части, такие как некорректная работа микросервисов и уязвимости в API;
- уязвимости каналов связи, связанные с перехватом токенов или небезопасным процессом передачи JWT;
- уязвимости, которые появляются при использовании OpenSource решений.

Также за этот период в разы увеличилось количество уязвимостей, связанных с формированием и проверкой подписи JWT.

Анализ показал, что 42% уязвимостей, связанных с подписью JWT и с новыми категориями, имеют рейтинг «критический» и составляют 93% от об-щего числа критических уязвимостей за период с 2021 по настоящее время.

Начиная с 2020 г., появляются уязвимости, которые используют слабости одновременно в нескольких областях жизненного цикла JWT и поэтому могут быть отнесены сразу к нескольким категориям. Мы будем называть такие уязвимости комбинированными (оранжевый график). Половина этих уязвимостей являются критическими, а вторая половина имеют рейтинг «высокий».

Подводя итог, мы видим, что с 2021 г. наблюдается резкое увеличение числа уязвимостей JWT. По мнению авторов, это обусловлено массовым использованием токенов в системах аутентификации и авторизации, а также в OpenSource решениях, набравших популярность в постковидный период. При этом за счет использования новых технологий в серверной части (например, микросервисной архитектуры) появляются новые виды

уязвимостей, в том числе комбинированные с уже существующими типами уязвимостей, что существенно повышает итоговый уровень их критичности.

Анализ жизненного цикла JWT

Рассмотрим жизненный цикл JWT с точки зрения возможных уязвимостей. Так как на нулевом этапе создание токена только инициируется, то анализировать жизненный цикл JWT нужно с первого этапа. На данном шаге происходят процессы генерации токена и отправка его клиенту. Выделяются следующие виды возможных уязвимостей:

Хранение в полях токена конфиденциальной информации. В виду того, что JWT хранит информацию в открытом виде, передавать электронные почты, номера телефонов и иную чувствительную информацию пользователей внутри токена недопустимо, поскольку возможен риск компрометации этих данных. Данная уязвимость может эксплуатироваться проведением атак типа «человек посередине» (далее – MITM) при условии использования незащищенного канала связи между клиентом и сервисом, а также проведением XSS-атак (CVE-2022-22311).

Использование криптографически нестойкого секретного ключа. Подпись токена – это гарантия безопасности передаваемой информации и секретный ключ играет здесь главную роль. Если удалось его получить, то будет скомпрометирована вся система аутентификации, а у злоумышленника появится возможность делать запросы к приложению от имени любого пользователя (CVE-2023-26089, CVE-2022-44796, CVE-2022-42980). Данная уязвимость может эксплуатироваться атакой типа bruteforce.

Использование незащищенного канала связи. Если между SAuth и клиентом не обеспечивается защищенный канал связи, становится возможным про-вести атаку MITM. За счет нее нарушитель, находясь в одной сети с пользователем, может перехватывать исходящие и входящие http-пакеты. Если трафик не будет зашифрован, то злоумышленник получает возможность узнать и перехватить токен пользователя в тот момент, когда SAuth будет передавать JWT клиенту (CVE-2022-22311).

На втором этапе жизненного цикла JWT происходит процесс сохранения токена на стороне клиента. На данном шаге существует уязвимость, связанная с *небезопасным хранением токена.*

Сохранение пользовательского JWT возможно в нескольких местах браузера, например, в cookies или LocalStorage. Если токен был сохранен в cookies, то клиент уязвим к атаке типа CSRF [6, 7]. Данная уязвимость влечет за собой возможность выполнения произвольных действий от имени жертвы, а также риск захвата аккаунта (CVE-2020-1762).

Если JWT, полученный от SAuth, сохранятся в LocalStorage браузера, то возникает угроза XSS-атак на пользователя. Их суть состоит в том, чтобы от-править пользователю вредоносную ссылку, при переходе по которой злоумышленник получит информацию из локального хранилища браузера, в котором находится токен жертвы. Итогом эксплуатации XSS-уязвимости является возможность захвата аккаунта пользователя и отправка от его имени произвольных запросов на сервис (CVE-2023-34088, CVE-2021-3509).

На третьем этапе жизненного цикла JWT происходит обращение клиента к серверу с предъявлением токена. Уязвимым в данном случае является канал передачи JWT, и если он не является безопасным, то становится возможным проведение атаки типа MITM.

На последнем этапе цикла выполняется процедура обработки JWT. Дан-ный этап содержит в себе наибольшее количество уязвимостей и возможностей атак на токены:

Некорректная обработка полей JWT. Поскольку стандарт RFC 7519 JSON Web Token (JWT) предусматривает возможность применения произвольных полей в составе JWT, то их нужно правильно и безопасно обрабатывать. Если администраторы системы допустили ошибку при создании механизма обработки полей, то становится возможным проведение целого ряда атак на поля JWT:

1. Поле «username»: обычно оно используются с целью авторизации пользователя на сервисе и определения его роли. Но бывают случаи, когда значение данного поля напрямую подставляется в базу данных с именами пользователей. В таком случае становится возможным провести атаку типа SQL-injection.

2. Поле «role»: как правило, данное поле, если оно присутствует в токене, обозначает, какими привилегиями обладает пользователь. Если значение данного поля не сопоставляется со значениями из других полей токена, то становится возможным повысить привилегии, изменив значения поля, к примеру, на «admin». Также нужно учитывать тот факт, что значения некоторых символов, на-

пример, пробела, могут не учитываться при обработке токена. Таким образом, если значение «admin» вызовет ошибку, то значение «admin» может пройти проверку (CVE-2023-23612).

3. Поле «aud»: является одним из стандартных полей JWT, оно означает, для какого пользователя данный токен был выпущен. Обычно значением данного поля выступает уникальный идентификатор (ID) пользователя. При недостаточной проверке становится возможным повысить привилегии путем смены своего ID на ID администратора (CVE-2018-6873).

4. Поле «iss»: также является стандартным полем и обозначает ресурс, выпустивший данный токен. При проверке данного поля становится возмож-ным изменить его значение так, что JWT будет корректно проходить проверку на тех сервисах, где этого не должно происходить (CVE-2017-8034).

5. Поле «jku» (JWK Set URL): данное поле описано стандартом RFC 7515 JSON Web Signature (JWS) и применятся в случае, когда для формирования подписи токена используются асимметричные алгоритмы шифрования. В данном поле размещается URL-адрес, на котором находится публичный ключ в виде объекта JSON. Если данное поле уязвимо, то становится возможным провести следующую атаку: сначала нужно создать свою пару приватного и публичного ключей, затем добавить на контролируемый веб-сервер файл с расширением .json, в котором будет находиться информация об открытом ключе. После этого необходимо сформировать свой JWT токен и подписать его сгенерированным приватным ключом, а также изменить значение поля «jku» на URL-адрес файла с информацией о публичном ключе. В случае успеха данной атаки, вся система аутентификации и авторизации становится скомпрометированной, поскольку злоумышленник будет способен выдать себя за любого пользователя.

6. Поле «jwk» (JSON Web Key): как и поле «jku», оно описано стандартом RFC 7515 и применяется только при использовании асимметричного алгоритма формирования подписи JWT. Поле предназначено для размещения информации о публичном ключе шифрования. Суть атаки заключается в формировании собственной пары приватный-публичный ключ, размещении информации о публичном ключе в поле «jwk» и подписи токена сформированным приватным ключом. Итог проведения успешной атаки – компрометация системы авторизации и аутентификации (CVE-2019-1010263, CVE-2018-0114).

7. Поле «kid» (Key ID): данное поле является стандартным [12] и предназначено для идентификации ключа, которым производится подпись JWT. Если значение данного поля не подвергать проверке, то оно потенциально уязвимо к таким атакам, как Path Traversal, SQL-injection, Remote Code Execution (RCE).

8. Поле «exp» (Expiration Time) определено стандартом RFC 7519 и предназначено для установления времени, после которого JWT считается недействительным. В случае уязвимости данного поля, токены с истекшим сроком годности считаются действительными и успешно обрабатываются сервисом, что позволяет злоумышленнику действовать от имени другого пользователя неограниченное количество времени (CVE-2020-26892).

Некорректная проверка подписи JWT. Алгоритмы формирования подписи различны и, несмотря на наличие рекомендаций со стороны стандарта RFC 7518, выбор алгоритма зависит только от администратора системы. На этапе проверки подписи возможны следующие виды атак:

1. Использование непредусмотренных алгоритмов: алгоритм формирования подписи задается сервисом аутентификации на этапе генерации токена и находится в поле «alg». Но если на стороне сервисов данное поле не проверяется или проверяется некорректно, становится возможно реализовать несколько атак. Например, в качестве значения поля «alg» можно указать «none» и в случае успешной реализации данной атаки целевой сервис не будет проверять подпись токена, т.е. станет возможным менять содержимое токена произвольным образом (CVE-2022-23540, CVE-2021-22160). Если для формирования подписи используется асимметричный алгоритм шифрования, то можно попробовать провести атаку типа algorithm confusion («путаница в алгоритмах»). Ее суть заключается в том, чтобы подписать токен открытым ключом шифрования и поменять в поле «alg» алгоритм на симметричный. Таким образом, при проверке JWT сервис не выдаст ошибку. Это произойдет потому, что в JWT мы указали симметричный алгоритм шифрования и сервис произвел расшифровку JWT тем ключом, которым мы его зашифровали (CVE-2022-23541, CVE-2021-46743). Итогом реализации

любой из атак является компрометация системы аутентификации и авторизации.

2. Недостаточная проверка сигнатуры токена или ее отсутствие: система проверки токена может быть спроектирована таким образом, что подпись JWT не проверяется или сервис считает действительными токены с нулевой длиной подписи. Это дает возможность любому пользователю выдать себя за другого и действовать от его имени (CVE-2023-2827, CVE-2022-39366, CVE-2022-39227).

Уязвимость механизма отзыва токена: если пользовательский JWT был скомпрометирован, то должен быть механизм прекращения действия данного токена, в противном случае злоумышленник сможет на протяжении длительного времени действовать от имени жертвы (CVE-2022-22332, CVE-2021-35342).

Незащищенность от внутреннего нарушителя: при обеспечении ком-плексной безопасности системы аутентификации и моделировании возможных угроз следует рассматривать как потенциальных злоумышленников не только внешних пользователей, но и тех, кто напрямую имеет доступ к серверному оборудованию и программному обеспечению. Характерными для внутреннего нарушителя атаками можно считать компрометацию секретного ключа для подписи JWT на серверной части (CVE-2021-23207, CVE-2021-3167), а также возможность проведения атаки типа MITM во время отправки токена от SAuth к клиенту.

Хранение секретных ключей в исходном коде приложения: если приложение имеет закрытый исходный код, то риск компрометации секретного ключа для подписи JWT существует только со стороны разработчиков данного сервиса или администраторов системы, если же программное обеспечение является открытым, то любой человек может узнать секретный ключ (CVE-2023-39846, CVE-2023-33372, CVE-2023-33371).

Некорректная обработка входящих запросов: вследствие уязвимости алгоритма обработки пользовательские запросы с некорректной подписью токена вызывают отказ в обслуживании. Зная об этом, злоумышленники могут на некоторое время вывести сервис из строя (CVE-2023-34429, CVE-2021-43824). При неправильной конфигурации сервиса, становится возможным обойти процедуру аутентификации, используя специальные поля в http-заголовках. Данная атака позволяет получить злоумышленнику несанкционированный доступ к сервису (CVE-2023-30845, CVE-2023-27487). Также при получении токена с неправильной подписью, сервис может в http-заголовках ответа прислать значение правильного секретного ключа (CVE-2023-40171, CVE-2022-39304).

Классификация атак на JWT

На основании проведенного анализа этапов жизненного цикла JWT и уязвимостей базы CVE, мы можем выделить классы атак на токены, которые представлены на рисунке 3.

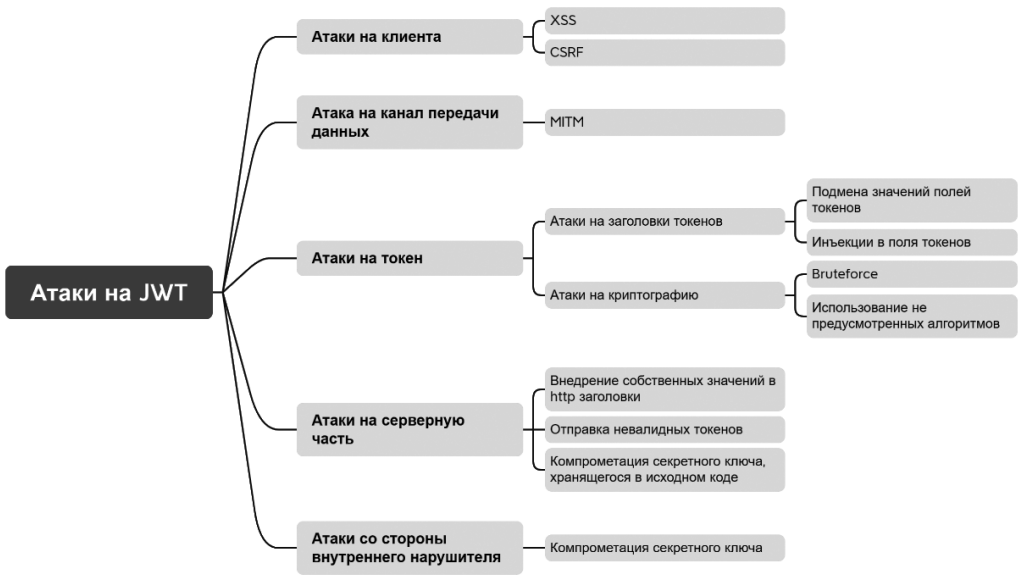

Рис. 3. Классификация атак на JWT

В ходе анализа мы выяснили, что нескольким уязвимостям на различных этапах жизненного цикла JWT может соответствовать одна атака и, как следствие, один объект атаки. В связи с чем было принято решение составить классификацию на основании возможных объектов атак, а также с учетом возможности наличия внутреннего нарушителя.

Атаки на клиента. Данная категория подразумевает под собой возможность эксплуатации уязвимостей посредством проведения атак типа XSS и CSRF, связанных прежде всего с небезопасным хранением JWT.

Атаки на канал передачи данных возможны в результате передачи JWT по незащищенному каналу связи. В связи с тем, что данные передаются в открытом виде, то для получения доступа к содержимому токена будет проведе-ния атаки типа MITM.

Атаки на токен реализуются за счет уязвимостей в структуре JWT. Из-за отсутствия в JWT ограничения на число создаваемых полей могут возникнуть проблемы их корректной и безопасной обработки, что позволяет скомпрометировать систему аутентификации за счет внедрения инъекций в поля заголовков или подмены их значений. Использование некриптосткойкого секретного ключа при подписывании токена и отсутствие контроля алгоритмов позволяют злоумышленникам провести атаки типа bruteforce или algorithm confusion.

Атаки на серверную часть становятся возможны за счет наличия уязви-мостей в конфигурации сервера. Так, например, уязвимость алгоритма обработки токенов с некорректной подписью может привести к ошибке типа отказ в обслуживании, некорректная обработка http-пакетов может позволить обойти механизм аутентификации, а хранение секретного ключа в исходном коде приложения дает возможность действовать от имени любого пользователя данного сервиса.

Атака со стороны внутреннего нарушителя подразумевает вредоносное воздействие на систему со стороны лица, у которого есть непосредственный доступ к серверному оборудованию и/или программному обеспечению. Для внутреннего злоумышленника характерны те же атаки, что и для внешнего, за исключением возможности компрометации секретного ключа при физическом доступе к серверу.

Данная классификация позволяет определить возможные объекты атаки и в сочетании с описанием уязвимостей жизненного цикла JWT позволит упростить построение векторов атак на токены. Как следствие, это обеспечит системный подход к нейтрализации угроз при проектировании информационной системы, использующей аутентификацию и авторизацию на основе JWT, а также поможет при проведении аудитов информационной безопасности. Однозначное разграничение атак и сопоставление их с выявленными уязвимостями позволит более точно подобрать инструментарий для тестирования безопасности JWT.

Заключение

В ходе исследования был проведен анализ существующих 119 уязвимостей базы CVE, связанных с использованием JWT в системе аутентификации. В результате был выявлен резкий рост числа появления новых уязвимостей, особенно критического уровня, начиная с 2021г. Отмечено появление новых категорий уязвимостей, в том числе комбинированных, которые могут быть обусловлены распространением JWT в качестве способа аутентификации и использованием его при работе в микросервисной архитектуре и OpenSource решениях.

Авторами был определен жизненный цикл JSON Web Tokens, состоящий из 4 основных этапов, и описана его структура с учетом анализа стандартов по JWT. Это позволило выявить 11 типов уязвимостей и сгруппировать их в соответствии с принадлежностью к тому или иному этапу жизненного цикла. Некоторые из выявленных уязвимостей не были представлены в научной литературе на данный момент. Для каждого из рассмотренных типов уязвимостей были определены возможные классы атак и способы их реализации. В результате была предложена классификация векторов атак на токены, основанная на их объектах атаки.

Для реализации комплексного подхода к обеспечению безопасности систем аутентификации и авторизации на основе JWT необходимо расширить и дополнить предложенную классификацию с целью обоснованного выбора методов и средств выявления уязвимостей токенов, оценки последствий их эксплуатации, построения векторов атак и определения защитных мер.

Литература

1. Mike Loukides, Steve Swoyer. Microservices Adoption in 2020. URL: https://www.oreilly.com/radar/microservices-adoption-in-2020 (дата обращения: 10.09.2023).

2. Феоктистов И.В. Сравнительное исследование методов аутентифика-ции в информационных системах // Инновации и инвестиции. 2023. № 7. С. 193–198.

3. Kalaska R., Czarnul P. Benchmarking Scalability and Security Configura-tion Impact for A Distributed Sensors-Server IOT Use Case, Proceedings of the 37th International Business Information Management Association (IBIMA), 30–31 May 2021, Cordoba, Spain, ISBN: 978-0-9998551-6-4, ISSN: 2767-9640.

4. Бетелин А.Б., Егорычев И.Б., Прилипко А.А. О некоторых особенно-стях JWT аутентификации в веб-приложениях // Труды научно-исследовательского института системных исследований российской академии наук. 2021. С. 4–10.

5. Mahindrakar P., Pujeri U. Insights of JSON Web Token // International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-6, March 2020. P. 1707–1710. DOI:10.35940/ijrte.F7689.038620.

6. Bulgakova O., Mashkov V., Zosimov V. Risk of Information Loss Using JWT Token // CIT-Risk'2021: 2nd International Workshop on Computational & Information Technologies for Risk-Informed Systems, September 16–17, 2021, Kherson, Ukraine.

7. Darmawan I., Gunawan R., Pramesti D. JSON Web Token Penetration Testing on Cookie Storage with CSRF Techniques // International Conference Advancement in Data Science, E-learning and Information Systems ICADEIS 2021. DOI: 10.1109/ICADEIS52521.2021.9701965.

8. Девицына С.Н., Пилькевич П.В., Удод Е.В. Способы улучшения защищенности сервисов, использующих JWT-токены // Экономика. Информатика. 2023. № 1. С. 144–151.

9. Никитин О.Р., Уймин А.Г. Инфраструктура JSON Web Token. Реализация основных типов атак // ПЕРСПЕКТИВЫ НАУКИ. 2023. № 2. С. 28-34.

10. RFC 7519 JSON Web Token (JWT). URL: https://datatracker.ietf.org/doc/html/rfc7519 (дата обращения: 15.09.2023)

11. RFC 7518 JSON Web Algorithms (JWA). URL: https://datatracker.ietf.org/doc/html/rfc7518 (дата обращения: 15.09.2023)

12. RFC 7515 JSON Web Signature (JWS). URL: https://datatracker.ietf.org/doc/html/rfc7515 (дата обращения: 17.09.2023)

References

1. Mike Loukides, Steve Swoyer. Microservices Adoption in 2020. Available at: https://www.oreilly.com/radar/microservices-adoption-in-2020 (accessed 10 Sep-tember 2023).

2. Feoktistov I.V. Sravnitel'noe issledovanie metodov autentifika-cii v informacionnyh sistemah [Innovacii i investicii]. 2023, no. 7, pp. 193–198.

3. Kalaska R., Czarnul P. Benchmarking Scalability and Security Configura-tion Impact for A Distributed Sensors-Server IOT Use Case, Proceedings of the 37th International Business Information Management Association (IBIMA), 30–31 May 2021, Cordoba, Spain, ISBN: 978-0-9998551-6-4, ISSN: 2767-9640.

4. Betelin A.B., Egorychev I.B., Prilipko A.A. O nekotoryh osobennostjah JWT autentifikacii v veb-prilozhenijah [Trudy nauchno-issledovatel'skogo instituta sistemnyh issledovanij rossijskoj akademii nauk]. 2021, pp. 4–10.

5. Mahindrakar P., Pujeri U. Insights of JSON Web Token [International Journal of Recent Technology and Engineering (IJRTE)]. 2020, vol. 8, issue 6, pp. 1707–1710. DOI:10.35940/ijrte.F7689.038620.

6. Bulgakova O., Mashkov V., Zosimov V. Risk of Information Loss Using JWT Token [CIT-Risk'2021: 2nd International Workshop on Computational & In-formation Technologies for Risk-Informed Systems]. September 16–17 2021, Kherson, Ukraine.

7. Darmawan I., Gunawan R., Pramesti D. JSON Web Token Penetration Testing on Cookie Storage with CSRF Techniques [International Conference Advancement in Data Science, E-learning and Information Systems ICADEIS]. 2021. DOI: 10.1109/ICADEIS52521.2021.9701965.

8. Devicyna S.N., Pil'kevich P.V., Udod E.V. Sposoby uluchshenija zashhishhjonnosti servisov, ispol'zujushhih JWT-tokeny [Jekonomika. Informatika]. 2023, no. 1. pp. 144-151.

9. Nikitin O.R., Ujmin A.G. Infrastruktura JSON Web Token. Realizacija osnovnyh tipov atak [PERSPEKTIVY NAUKI]. 2023, no. 2, pp. 28-34.

10. RFC 7519 JSON Web Token (JWT). URL: https://datatracker.ietf.org/doc/html/rfc7519 (accessed 15 September 2023)

11. RFC 7518 JSON Web Algorithms (JWA). URL: https://datatracker.ietf.org/doc/html/rfc7518 (accessed 15 September 2023)

12. RFC 7515 JSON Web Signature (JWS). URL: https://datatracker.ietf.org/doc/html/rfc7515 (accessed 15 September 2023)

ЗУЛЬКАРНЕЕВ Искандер Рашитович, доцент кафедры информационной безопасности федерального государственного автономного образовательного учреждения высшего образования «Тюменский государственный университет». 625003, г. Тюмень, ул. Володарского, 6. E-mail: i.r.zulkarneev@utmn.ru

БАСАЛАЙ Константин Алексеевич, студент федерального государст-венного автономного образовательного учреждения высшего образования «Тюменский государственный университет». 625003, г. Тюмень, ул. Володар-ского, 6. E-mail: stud0000272406@study.utmn.ru

ZULKARNEEV Iskander Rashitovich, Associate Professor of the Depart-ment of Information Security at the Federal State Autonomous Educational Institu-tion of Higher Education «Tyumen State University». 625003, Tyumen, st. Volodarskogo, 6. E-mail: i.r.zulkarneev@utmn.ru

BASALAY Konstantin Alekseevich, student of the Federal State Autono-mous Educational Institution of Higher Education «Tyumen State University». 625003, Tyumen, st. Volodarskogo, 6. E-mail: stud0000272406@study.utmn.ru