МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра теоретических основ компьютерной безопасности и криптографии

Методы и средства для обеспечения информационной безопасности web-сервисов с использованием токенов

АВТОРЕФЕРАТ
дипломной работы

студентки 6 курса 631 группы
специальности 10.05.01 Компьютерная безопасность
факультета компьютерных наук и информационных технологий
Гавриловой Виктории Викторовны

Научный руководитель
доцент, к.п.н. _____ А. С. Гераськин
22.01.2024 г.

Заведующий кафедрой
д. ф.-м. н., доцент _____ М. Б. Абросимов
22.01.2024 г.

Саратов 2024

ВВЕДЕНИЕ

В наше время в виде веб-сайтов реализовано огромное количество различных платформ, социальных сетей, обычных сайтов. Многие из них хранят персональную для пользователя информацию, разглашение которой является критической для пользователя: его логин, пароль, а также все его личные данные, которые он добавляет на веб-сайт или с которыми взаимодействует с помощью веб-сайта.

У веб-сайтов могут быт уязвимости, связанные, как и с внешней атакой на веб-сайт, попытками обхода защиты веб-сайта, так и внедрением вредоносной информации в веб-сайт.

Атаки могут быть направлены как на крупные компании, банковскую сферу, так и на частных лиц. На частных лиц было направлено примерно 15% атак. Преимущественно злоумышленники прибегали к методам социальной инженерии (88% атак), а основным мотивом было получение данных. Среди украденной у пользователей информации за три квартала 2021 года большую часть составили учетные данные (48% атак), персональные (20%) и данные платежных карт (10%).

В настоящее время огромный массив информации, которым мы пользуемся каждый день, располагается на веб-сайтах. Поэтому защита данных в нем является актуальной повесткой на сегодняшний день.

Особенно остро вопрос защиты данных веб-сервисов стоит на этапе авторизации. Актуальность JSON Web Token в современном информационном обмене обусловлена несколькими факторами. Во-первых, он предоставляет эффективный механизм аутентификации и авторизации, обеспечивая цифровую подпись для подтверждения подлинности данных. Во-вторых, JWT легко интегрируется в микросервисные архитектуры и API, что упрощает обмен данными между распределенными компонентами системы. В-третьих, его легковесная структура и поддержка шифрования делают его эффективным выбором для обеспечения безопасности в сфере веб-разработки. Таким образом,

использование JWT актуально в контексте современных требований к безопасности, масштабируемости и гибкости в области веб-приложений и взаимодействия между сервисами.

Цель работы: разработка веб-сервиса с реализованной защитой авторизации, ролями пользователей, автоматическим обновлением сессий и отслеживанием и пресечением подозрительной активности с помощью JWT.

Задачи работы:

1. Изучить теоретический материал
2. Рассмотреть основные виды атак и способы защиты веб-сервисов от этих атак
3. Разработать веб-сервис с защитой авторизации, автоматическим обновлением сессий и реализованным отслеживанием и пресечением подозрительной активности с помощью JWT.

Дипломная работа состоит из введения, 2 разделов, заключения, списка использованных источников и 4 приложений. Общий объем работы – 70 страниц, из них 48 страниц – основное содержание, включая 48 рисунков и 0 таблиц, список использованных источников из 23 наименований.

КРАТКОЕ СОДЕРЖАНИЕ

1 Использование JSON Web Token для защиты веб-сервисов

В первом разделе рассматриваются основные уязвимости веб-сервисов, определение JSON Web Token, основные компоненты JSON Web Token, описан процесс создания и верификации токена, рассмотрены виды атак на JSON Web Token и способы защиты от атак, также было проведено сравнение JSON Web Token с альтернативными методами аутентификации и авторизации

1.1 Наиболее часто распространенные уязвимости веб-сервисов

Были рассмотрены следующие атаки:

1. Фишинг
2. SQL-инъекции
3. Brute Force Attack (атака грубой силы)
4. Межсайтовый скриптинг (XSS)
5. DoS-атаки и DDoS-атаки
6. Атака «Человек посередине (Man-in-the-middle, MITM)

Были даны их определения, возможные пути реализации.

1.2 JSON Web Token

Токены предоставляют безопасный и стандартизированный способ аутентификации и авторизации. Они позволяют пользователям получать доступ к ресурсам и сервисам, предоставляя доверенные утверждения, которые могут быть проверены и использованы сервером.

JSON Web Token (JWT) – это открытый стандарт (RFC 7519), который определяет компактный и автономный способ безопасной передачи информации между сторонами в формате JSON. Эта информация может быть проверена и доверена, поскольку она подписана. JWT могут быть подписаны с использованием секрета (алгоритм HMAC) или ключевой пары открытого/закрытого ключа с применением RSA или ECDSA.

Вот несколько сценариев, в которых JSON Web Tokens оказываются полезными:

1) Аутентификация.

Аутентификация происходит, когда клиент успешно подтверждает свою личность через конечную точку входа. В случае успеха сервер создает JSON Web Token и отправляет его в ответ клиенту.

Клиент будет использовать этот JWT при каждом запросе к защищенному ресурсу.

2) Авторизация.

Сервер, построенный на основе JWT для авторизации, создаст JWT при входе клиента. Этот JWT подписан, поэтому никакая другая сторона не может его изменить.

Каждый раз, когда клиент имеет доступ к защищенным ресурсам, сервер будет проверять, что подпись JWT соответствует его полезной нагрузке и заголовку, чтобы убедиться в его допустимости.

Если JWT успешно проверен, он может предоставить или отказать в доступе к ресурсу.

3) Обмен данных

JWT также отличный способ обеспечить безопасность передачи информации между сторонами — например, между двумя серверами — потому что можно проверить допустимость токена (подпись, структура или стандарты, указанные в JWT).

1.3 Основные компоненты JWT

В компактной форме JSON Web Tokens состоят из трех частей, разделенных точками (.):

1) Заголовок (Header)
2) Нагрузка (Payload)
3) Подпись (Signature)

Давайте подробно рассмотрим каждую часть.

1) Заголовок (Header)

Заголовок обычно состоит из двух частей: типа токена (JWT) и используемого алгоритма подписи, такого как HMAC SHA256 или RSA.

Затем этот JSON кодируется в Base64Url, чтобы формировать первую часть JWT.

2) Нагрузка (Payload)

Вторая часть токена - нагрузка, содержащая утверждения. Утверждения - это заявления об объекте (обычно пользователе) и дополнительные данные. Существует три типа утверждений: зарегистрированные, публичные и частные.

Затем нагрузка кодируется в Base64Url, чтобы сформировать вторую часть JSON Web Token.

Важно отметить, что для подписанных токенов эта информация, хотя и защищена от вмешательства, доступна для прочтения всем. Не помещайте секретную информацию в нагрузку или заголовок JWT, если они не зашифрованы.

3) Подпись (Signature)

Для создания части подписи необходимо взять закодированный заголовок, закодированную нагрузку, секрет, указанный в заголовке алгоритм, и подписать их.

1.4 Процесс создания и верификации JWT

1. Приложение или клиент запрашивает авторизацию у сервера авторизации. Это выполняется через один из различных потоков авторизации. Например, типичное веб-приложение, совместимое с OpenID Connect, пройдет через конечную точку `/oauth/authorize`, используя поток авторизации кода.

2. Когда авторизация предоставлена, сервер авторизации возвращает приложению токен доступа.

3. Приложение использует токен доступа для доступа к защищенному ресурсу (например, API).

1.5 Виды атак на JWT

Виды атак на JWT:

1) Использование JWT без верификации
2) Прием JWT без указанного алгоритма подписи
3) Подбор секретных ключей

4) Атака с путаницей алгоритмов
5) Подделка заголовка JKU с помощью SQL-инъекций
6) Подделка заголовка kid с помощью SQL-инъекций
7) Brute Force атака на секретный ключ для алгоритмов с симметричным шифрованием
8) Изменение алгоритма шифрования с ассиметричного на симметричное

1.6 Защита от атак на JWT

Для защиты от описанных атак на JWT рекомендуется следовать правилам:

1. Всегда использовать метод верификации, а не только декодирование
2. Обязательно указывать алгоритм подписи
3. Не доверять токенам без подписи
4. Секретные ключи:
   - Использовать долгие и случайные секретные ключи для симметричных алгоритмов.
   - Регулярно обновлять секретные ключи, особенно если существует риск их утечки.
   - Избегать использования секретов по умолчанию или общих секретов.
5. Проверка параметра jku
6. Ассиметричные алгоритмы:
   - При использовании асимметричных алгоритмов (например, RSA) удостовериться, что токены подписаны закрытым ключом и проверяются открытым ключом.
   - Не позволять изменять алгоритм шифрования токена после его подписи.
7. Ограничения срока действия (exp)
8. Использовать белый список вместо черного списка
9. Обновление библиотек
10. Обучение и обзор кода

1.7 Сравнение JWT с альтернативными методами аутентификации и авторизации

Ниже будет проведено сравнение JSON Web Token с сессионной аутентификацией и сравнение с OAuth и OpenID Connect

1.7.1. Сравнение JWT с сессионной аутентификацией

При сравнении рассматривались основные преимущества и недостатки сессионной аутентификации, основные преимущества и недостатки JSON Web Token и были сделаны общие выводы относительно преимуществ и недостатков JSON Web Token и сессионной аутентификации относительно друг друга

1.7.2. Сравнение с OAuth и OpenID Connect

При сравнении рассматривались основные преимущества и недостатки JSON Web Token относительно OAuth, основные преимущества и недостатки JSON Web Token относительно OpenID Connect и были сделаны общие выводы относительно преимуществ и недостатков JSON Web Token.

2 Практическая часть

Во втором разделе представлена реализация программы для создания и защиты веб-сервиса. Описано создание серверной части веб-сервиса, реализация дополнительной защиты для веб-сервиса, осуществлена проверка веб-сервиса на устойчивость к атакам и описано создание клиенткой части веб-сервиса.

2.1 Создание серверной части веб-сервиса

В данной работе серверная часть веб-сервиса была реализована на Node.js с подключением базы данных PostgreSQL.

В данном веб-сервисе будет реализована система ролей и прав. В данном случае будут роли администратора и обычного пользователя. У администратора будет возможность изменять данные любых пользователей, а для всех остальных будет доступно изменение только собственных данных.

Была реализована система, в которой пользователи могли авторизоваться, при авторизации определялись их роли в системе и, в соответствии с ролями, пользователям было либо разрешено удаление всех пользователей системы, либо

было разрешено только удаление самого себя. Все данные, за исключением токена доступа, о котором будет рассказано дальше, хранятся в базе данных.

JSON Web Token применялся при авторизации. В процессе входа в систему, при корректно введенных логине и пароля, пользователю выдавался токен доступа, который хранится на сервере, и токен обновления, который уже заносится в базу данных.

Токен обновления предназначен для реализации сессий в системе и для обмена данными. Токен хранит в себе информацию о правах пользователя и на основании этих прав подтверждает или запрещает выполнение тех или иных действий пользователя.

Токен доступа хранится на сервере и используется для верификации запроса на обновление токена обновления, без корректного токена доступа нельзя обновить токен обновления и продолжить взаимодействие с веб-сервисом, пользователь автоматически выйдет из системы.

2.2 Реализация дополнительной защиты веб-сервиса

Была реализована защита:

1) От SQL-инъекций

Использовались параметризированные запросы, благодаря которым введенные пользователем значения будут обработаны как данные, а не как часть SQL-кода, что уменьшает риск SQL-инъекций. Библиотека автоматически обрабатывает и экранирует введенные значения, предотвращая возможность внедрения зловредного SQL-кода.

2) Обнаружение подозрительной активности

Под подозрительной активностью в данной системе предполагается: превышение установленного числа неудачных попыток входа, превышение установленного числа ошибок, возникших при использовании токена обновления, что позволяет нам предположить, что используется недействительный токен обновления, использование токена обновления не с того IP-адреса, с которого токен обновления изначально был создан.

Функция проверки на подозрительную активность запускается каждый раз при отправке запроса пользователем, а также дополнительно при каждом обновлении токена обновления и при функции авторизации для фиксирования превышения установленного количества неудачных попыток.

Обнаруженная подозрительная активность также фиксируется в базе данных.

3) Блокировка IP-адресов, с которых совершалась подозрительная активность

Мы фиксируем подозрительную активность и блокируем IP-адреса сначала на 30 секунд, а затем на 1 час, с которых эта подозрительная активность осуществляется. Также была добавлена функция проверки на наличие актуальной блокировки IP-адреса.

Заблокированные IP-адреса и время начала и окончания их блокировки также хранятся в базе данных.

4) Оповещение администратора об обнаруженной подозрительной активности

Если была обнаружена подозрительная активность, то о ней также сообщается на почту администратору с указанием IP-адреса, с которого была совершена подозрительная активность.

2.3 Проверка веб-сервера на устойчивость к атакам

Здесь были рассмотрены следующие атаки и подтверждена устойчивость к ним:

1) Превышение установленного числа попыток неудачной авторизации.
2) Попытка входа, если IP-адреса, с которого осуществляется вход, в данный момент заблокирован
3) Попытка использования уже истекшего токена обновления
4) Попытка выполнения запроса с корректным токеном обновления, но с IP-адреса, отличающего от того, с которым был создан токен обновления
5) Выполнение запросов без верификации токена обновления

2.4 Создание клиентской части веб-сервиса

Логика клиентской части веб-сервиса была реализована на React, визуальное оформление страницы осуществлялось с помощью HTML и CSS.

Было реализовано подключение, получение и передача запросов к серверной части и созданы страницы авторизации и главная страница, на которой можно осуществлять удаление пользователей. Внешний вид страниц представлен на рисунках 1-2.

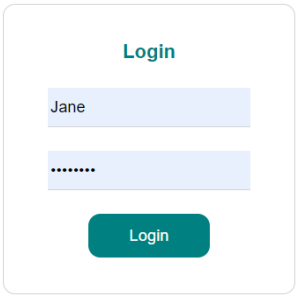

Рисунок 43– Результат реализации экрана авторизации

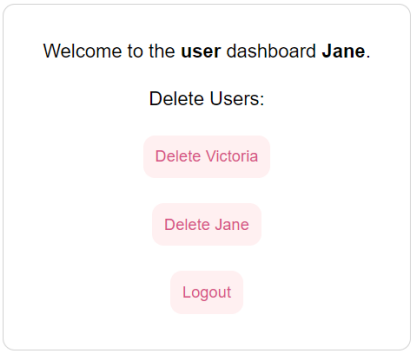

Рисунок 44 – Результат реализации экрана удаления пользователей

ЗАКЛЮЧЕНИЕ

В результате теоретического исследования было выявлено, что актуальность JWT обусловлена несколькими факторами. Во-первых, он предоставляет эффективный механизм аутентификации и авторизации, обеспечивая цифровую подпись для подтверждения подлинности данных. Во-вторых, JWT легко интегрируется в микросервисные архитектуры и API, что упрощает обмен данными между распределенными компонентами системы. В-третьих, его легковесная структура и поддержка шифрования делают его эффективным выбором для обеспечения безопасности в сфере веб-разработки.

В результате теоретического исследования были рассмотрены основные виды атак и способы защиты веб-сервисов от этих атак

Был реализован веб-сервис с защитой авторизации, автоматическим обновлением сессий и реализованным отслеживанием и пресечение подозрительной активности с помощью JWT. Использование JWT и особенно автоматическое обновление сессий с обновлением токена доступа является одним из необходимых требований для создания качественного и безопасного сайта, поэтому при необходимости рекомендую подстроиться под существующие стандарты.

Поставленные задачи были выполнены полностью.