DOI 10.37539/2949-1991.2025.24.1.028
УДК 004.056

Дмитриев Дмитрий Валерьевич,
к.т.н., доцент кафедры “Информатика и Системы Управления”
Нижегородский государственный технический университет
им. Р.Е. Алексеева, Нижний Новгород, Россия

Исаев Максим Александрович, магистрант,
Нижегородский государственный технический университет
им. Р.Е. Алексеева, Нижний Новгород, Россия

Вайнбаум Денис Алексеевич, магистрант,
Нижегородский государственный технический университет
им. Р.Е. Алексеева, Нижний Новгород, Россия

Мельников Роман Васильевич, магистрант,
Нижегородский государственный технический университет
им. Р.Е. Алексеева, Нижний Новгород, Россия

ОЦЕНКА МЕХАНИЗМОВ АУТЕНТИФИКАЦИИ В PYTHON BACKEND-ПРИЛОЖЕНИЙ

Аннотация: В работе рассматриваются различные методы аутентификации в backend-приложениях, их преимущества и недостатки. Представлены возможные уязвимости аутентификации в backend-приложениях и обзор существующих методов реализации. В заключении определяются наиболее подходящие способы реализации для определенных методов аутентификации, в зависимости от требований приложения.

Ключевые слова: безопасность, конфиденциальность, веб-уязвимости, атаки на приложения, протоколы безопасности, идентификация пользователя.

Введение

Аутентификация является одним из основных способов обеспечения безопасности веб-приложений. Она позволяет идентифицировать пользователя в системе, обеспечивает безопасность контроля доступа к данным, предоставляет информацию для дальнейшего взаимодействия пользователя с системой.

Рассматривая рейтинг OWASP, составленный на основе исследования уязвимостей в 2019 году, следует отметить, что уязвимости веб приложений, связанные с недостатками аутентификации имеют отметку «высокий риск» и лидируют в рейтинге уязвимостей [1].

Данная тенденция сохраняется и подтверждается исследованиями лаборатории Касперского в 2021-2023 годах. Много веб-приложений содержит определённую уязвимость и критичное воздействие этой ошибки на приложение [2]. Недостатки аутентификации попали в топ-6 угроз.

Существующие проблемы аутентификации

Уязвимости аутентификации делятся на следующие ключевые группы: недостаточная аутентификация, уязвимости в конфигурациях аутентификации и проблемы с управлениями сессиями. Каждая группа каждая представляет конкретные опасности и риски

Недостаточная аутентификация возникает, когда система позволяет получить доступ к важной информации или функциям без достаточной аутентификации [3], например в интерфейсах администрирования, доступных без надлежащей проверки подлинности, т.е. без кода доступа для администратора или аутентификации.

Уязвимости в конфигурациях возникает при неправильной конфигурации систем аутентификации и авторизации, например в разрозненных средствах контроля безопасности или уязвимых настройках сетевых ресурсов. Сюда же можно отнести утечки, возникающие при восстановлении пароля, перегрузке системы аутентификации, атаки при передаче и обработке конфиденциальных данных [4], атаки на внутренние хранилища системы.

Проблемы с управлениями сессиями включают слабую реализацию механизмов, отвечающих за идентификацию пользователей и управление их сессиями [5], например отсутствие ограничений на количество попыток входа или неправильное управление сессиями, неправильное хранение токена, редкое обновление [6].

Модели аутентификаций

Наиболее популярным методом аутентификации является парольная аутентификация, Single Sign-On (SSO), двухфакторная аутентификация и OAuth.

Парольная аутентификация – это самый распространенный метод, основанный на сравнении введенного пользователем пароля с хранимым в базе данных хешем. Преимущества – простота и универсальность; недостатки – высокая вероятность взлома и необходимость надежного хранения паролей.

Single Sign-On (SSO) предоставляет возможность единовременной аутентификации для доступа ко всем связанным сервисам. Преимущества – удобство использования и повышение безопасности за счет исключения слабых паролей; недостатки – сложность настройки и возможные проблемы с безопасностью в случае утечек идентификаторов.

Двухфакторная аутентификация (2FA) включает два этапа проверки: введение пароля и подтверждение через дополнительный фактор (SMS-код, приложение). Преимущества – повышенная безопасность благодаря уникальности временных кодов; недостатки – дополнительные требования к оборудованию и возможная уязвимость при использовании слабых паролей.

OAuth – это стандарт авторизации, позволяющий пользователям предоставлять доступ к своим данным сторонним сервисам без передачи учетных данных. Преимущества – отсутствие необходимости делиться своими данными напрямую, независимость стандарта; недостатки – необходимость доверия третьей стороне и потенциальный риск излишнего доступа к данным.

Все перечисленные методы имеют свои сильные и слабые стороны, которые важно учитывать при выборе подходящего решения для обеспечения безопасности.

Существующие решения аутентификации

Для реализации методов аутентификации используются решения, включающие обширный функционал и обеспечивающие удобное использование и быструю реализацию.

Библиотека Authomatic предназначена для упрощения процесса аутентификации и авторизации пользователей через различные социальные сервисы и провайдеры OAuth. Она поддерживает протоколы OAuth 1.0a и OAuth 2.0 [8]. Поддержка данных протоколов позволяет этой библиотеке работать с широким спектром провайдеров, включая Facebook, Google, LinkedIn, Twitter, и многие другие.

Keycloak является мощной и гибкой системой управления идентификацией (Identity Management) и аутентификации, разработанной компанией Red Hat. Keycloak поддерживает различные протоколы аутентификации и авторизации, включая OpenID Connect, SAML, и OAuth 2.0. Keycloak поддерживает двухфакторную аутентификацию (2FA), включая использование одноразовых паролей (OTP) с помощью инструментов Google Authenticator.

Passlib является мощной и универсальной библиотекой для хэширования паролей в Python, предназначенной для обеспечения безопасного хранения и управления паролями.

Passlib поддерживает более 30 различных алгоритмов хэширования паролей, включая популярные алгоритмы seperti bcrypt, PBKDF2, Argon2, и многие другие. Библиотека работает на различных платформах и поддерживает Python 2 и Python 3 [9].

Authlib является мощной и гибкой библиотекой для Python, предназначенной для упрощения процесса аутентификации и авторизации в веб-приложениях. Authlib разработана для поддержки различных протоколов аутентификации и авторизации, включая OAuth 1.0, OAuth 2.0, OpenID Connect, и другие. Поддержка данных протоколов делает ее универсальным инструментом для интеграции с различными провайдерами идентификации.

Основными критериями выбора той или иной библиотеки являются простота интеграции, поддержка сообщества, качество документации, сложность тестирования, гибкость и область применения.

Заключение

В данной работе рассмотрены существующие решения, применимые для реализации методов аутентификации, каждое из которых имеет определенный функционал и ограничения в использовании, из-за чего необходимо учитывать конкретную область проекта [10].

Рассматривая многопользовательские приложения, где backend написан на Python, на примере сайтов с онлайн-продажами, следует применять KeyCloak или Authtomatic, которые предоставляют наиболее гибкие способы реализации рассмотренных методов аутентификации и обеспечивают наиболее высокий уровень безопасности. При этом KeyCloak используется для двухфакторной и SSO аутентификации, но за счёт высокой гибкости имеет сложность реализации. AuthLib, более проста в реализации, но более применима для метода OAuth. Соответственно, необходимо грамотно использовать эти решения, под соответствующую ситуацию, с возможностью их интеграции и совместного использования, для дальнейшей эксплуатации системы аутентификации в приложении.

Список литературы:

1. Рейтинг OWASP [Электронный ресурс]. – Режим доступа: https://www.ptsecurity.com/ru-ru/research/analytics/web-vulnerabilities-2020/ (дата обращения 27.11.2024).

2. Рейтинг лаборатории Касперского [Электронный ресурс]. – Режим доступа: https://dzen.ru/a/ZfbA_9UUziZwlhba?ysclid=m4ygu2pnu3433070244 (дата обращения 1.12.2024).

3. A. Sharif Best current practices for OAuth/OIDC Native Apps / A. Sharif, R. Carbone, G. Sciarretta, S. Ranise // Journal of Information Security and Applications. – 2022.

4. Хлебницын С. А. Анализ современных методов аутентификации и их уязвимости / С. А. Хлебницын // Личность, право, государство: проблемы развития и взаимодействия: сборник статей научно-представительских мероприятий. – Москва: ИП Колупаева Е.В., 2023. – С. 437-439.

5. Кураков В. И. Анализ уязвимостей биометрических методов аутентификации / В. И. Кураков, А. С. Худадян, У. М. Баева // Вестник науки. – 2022. – Т. 2, № 5 (50). – С. 87-98.

6. Uvarov A. D. Analysis and modification of information security methods for ensuring the mobile UMTS data transmission networks / A. D. Uvarov, N. V. Kormiltsev, G. S. Kornilov // Information Technology. Problems and Solutions. – 2019. – No. 2 (7). – P. 113-116.

7. Мешкова, А. А. Методы аутентификации в современных web-приложениях / А. А. Мешкова, Д. В. Кругликов // Студент года 2023: Сборник статей Международного учебно-исследовательского конкурса, Петрозаводск, 15 мая 2023 года. – Петрозаводск: Международный центр научного партнерства «Новая Наука» (ИП Ивановская И.И.), 2023. – С. 372-379.

8. Uralov Ja. Effectiveness of the Metasploitable environment in the learning process for teaching web application security: an innovative approach in education / Ja. Uralov // Universum: технические науки. – 2024. – No. 9-4 (126). – P. 68-70.

9. Документация KeyCloak [Электронный ресурс]. – Режим доступа: https://pypi.org/project/python-keycloak/ (дата обращения 27.11.2024.

10. Corman Aguado A. Solutions for non-web OAuth 2.0 authorisation at CERN / A. Aguado Corman, Ja. Henschel, H. Short, S. Lopienski // EPJ Web of Conferences. – 2024. – Vol. 295. – P. 04038.