Университет ИТМО, факультет программной инженерии и компьютерной техники
Двухнедельная отчётная работа по «Информатике»: аннотация к статье

Дата прошедшей лекции	Номер прошедшей лекции	Название статьи/главы книги/видеолекции	Дата публикации (не старше 2022 года)	Размер статьи (от 400 слов)	Дата сдачи
10.09.2025	1	Нестандартные системы счисления	18.09.2024	~1060	08.10.2025
24.09.2025	2	Обзор методов помехоустойчивого кодирования	11.09.2023	~870	08.10.2025
08.10.2025	3	Применение регулярных выражений для обработки данных	21.11.2022	~796	22.10.2025
22.10.2025	4	Анализ типовых уязвимостей при использовании JSON Web Token (JWB) в системах аутентификации	20.10.2025	~1757	19.11.2025
	5				
	6				
	7				

Выполнил(а) Жгилев И. И., № группы *P3130*, оценка _____
Фамилия И.О. студента — не заполнять

Прямая полная ссылка на источник или сокращённая ссылка (bit.ly, tr.im и т.п.)
https://cyberleninka.ru/article/n/analiz-tipovyh-uyazvimostey-pri-ispolzovanii-json-web-oken-jw-v-sistemah-autentifikatsii

Теги, ключевые слова или словосочетания (минимум три слова)
JWT, аутентификация, токен, уязвимость, подпись токена, срок действия, защита токенов

Перечень фактов, упомянутых в статье (минимум четыре пункта)
1. JSON Web Token – компактный вариант передачи данных в виде строки, состоящей из заголовка, полезной нагрузки и подписи.
2. Сервер получает JWT от клиента, расшифровывает его и проверяет подлинность по подписи.
3. Безопасность зависит от реализации алгоритма подписи.
4. Существует несколько типовых уязвимостей: смена алгоритма(alg «none»), подмена алгоритма HS256 RS256 или наоборот, утечка секретного ключа, неправильная проверка срока действия токена, уязвимости в библиотеке или зависимостях, повторное использование токена.
5. Безопасное использование JWT заключается в комплексном подходе к хранению, проверке и контролю жизненного цикла токена.
6. Существуют инструменты с графическим интерфейсом и консольные утилиты, помогающие в анализе и тестировании безопасности JWT.

Позитивные следствия и/или достоинства описанной в статье технологии (минимум три пункта)
1. Серверу не нужно хранить состояние сессии пользователя в памяти или базе данных. Для аутентификации достаточно проверить валидность подписи и данных самого токена. Это значительно упрощает архитектуру приложения и облегчает его масштабирование.
2. Полезная нагрузка JWT позволяет хранить различные утверждения о пользователе (идентификатор, роли, срок действия), которые могут быть использованы сервером для авторизации без необходимости дополнительных запросов к базе данных.
3. JWT представляет собой лаконичную строку, которая содержит всю необходимую информацию. Это делает токен удобным для передачи в HTTP-запросах

Негативные следствия и/или недостатки описанной в статье технологии (минимум три пункта)
1. Уязвимость системы к критическим ошибкам настройки и кодирования. Малейшая оплошность (например, доверие алгоритму из заголовка токена) может привести к полной компрометации системы аутентификации.
2. Полезная нагрузка JWT по умолчанию не зашифрована, а только закодирована в Base64URL. Это означает, что конфиденциальные данные не должны помещаться в токен, так как их может легко прочитать любой, кто получит к нему доступ.
3. Поскольку JWT часто хранится на клиенте (в браузере), он становится мишенью для атак. Без использования защищённых cookie токен может быть украден.

Ваши замечания, пожелания преподавателю *или* анекдот о программистах[1]
Открыл 4 лабу, увидел в варианте перевод из RON в YAML, подумал, что теперь надо учить RUST, проснулся, переосмыслил жизнь, вышел потрогал траву и пошел писать парсер из INI в HCL

[1] Наличие этой графы не влияет на оценку