JWT (JSON Web Token)

1. Aspecte generale

Așa cum se precizează în articolele <https://jwt.io/introduction>, <https://en.wikipedia.org/wiki/JSON_Web_Token> și în cartea <https://assets.ctfassets.net/2ntc334xpx65/o5J4X472PQUI4ai6cAcqg/13a2611de03b2c8edbd09c3ca14ae86b/jwt-handbook-v0_14_1.pdf>, JSON Web Token este un standard utilizat în transmiterea securizată a informațiilor alcătuit din trei părți separate prin punct: antet, payload și semnătură sau așa-numitele: header, payload, signature. Datorită faptului că JWT are posibilitatea de a fi semnat și criptat, acesta devine o metodă puternică de securizare a datelor servind astfel proceselor de autentificare și de autorizare. Pentru partea de autorizare, JWT este util în acordarea de permisiuni utilizatorului în accesarea diferitelor resurse. De exemplu, prin JWT se poate specifica rolul de utilizator sau de admin, roluri prin care se gestionează accesul la resurse protejate.

1. Mod de funcționare

Potrivit sursei <https://www.youtube.com/watch?v=UBUNrFtufWo>, înaintea utilizării autentificării cu jeton sau token, se utiliza autentificarea cu sesiune. Autentificarea cu sesiune presupune transmiterea datelor de logare de la client la server, date pe baza cărora se creează o sesiune în baza de date sau în memoria serverului, iar id-ul acestei sesiuni va fi la rândul lui pus de către browser în secțiunea cookies. La fiecare cerere, browserul va trimite cookie-urile spre server pentru a se putea verifica sesiunea, iar mai apoi serverul va trimite un răspuns înapoi către client. Acest tip de autentificare are o vulnerabilitate crescută la atacuri, un exemplu de atac fiind CSRF (Cross-site request forgery). Problema principală care conduce la asemenea atacuri este stocarea sesiunii în baza de date sau în memoria browserului. De aceea, o metodă mai sigură este autentificarea cu jeton/token. Aceasta presupune inițial același lucru ca și autentificarea cu sesiune, adică trimiterea datelor utilizatorului de la client la server. Serverul creează un JSON Web Token (JWT) pe care îl transmite spre browser, iar browserul îl stochează într-un depozit local denumit local storage. Împreună cu cererile care vor fi făcute către server, browserul transmite și JSON Web Token-ul semnat pentru a fi validat de către server prin intermediul semnăturii. În comparație cu autentificarea prin sesiune prin care sesiunea e gestionată la nivelul serverului, acest tip de autentificare permite gestionarea token-ului la nivel de client pentru evitarea atacurilor.

1. Structură

JWT este alcătuit din 3 părți componente codificate separat:

* Header – alcătuit din tipul jetonului și din algoritmul utilizat
* Payload – cuprinde datele utilizatorului
* Signature – semnătura este creată prin codificarea headerului și a payloadului prin algoritmul dat în header fiind utilizat și un mesaj secret în componența acesteia.

1. Tipuri de atacuri ale securității site-urilor web

* Eliminarea semnăturii din conținutul JSON Web Token-ului
* Falsificarea cererii ( Cross-Site Request Forgery – CSRF)
* Falsificarea scripturilor ( Cross-Site Scripting – XSS)

Bibliografie

1. The JWT Handbook Sebastián E. Peyrott, Auth0 Inc. Version 0.14.1, 2016-2018 ->>>> BIBTEX SAU LATEX
2. <https://jwt.io/introduction>
3. <https://en.wikipedia.org/wiki/JSON_Web_Token>
4. <https://www.bezkoder.com/jwt-json-web-token/>
5. <https://datatracker.ietf.org/doc/html/rfc7519>
6. <https://www.bezkoder.com/angular-12-spring-boot-jwt-auth/>
7. <https://www.bezkoder.com/angular-12-jwt-auth/>
8. <https://www.bezkoder.com/spring-boot-refresh-token-jwt/>
9. <https://www.bezkoder.com/spring-boot-security-postgresql-jwt-authentication/>
10. https://www.bezkoder.com/spring-boot-jwt-mysql-spring-security-architecture/