Connaissez-vous vraiment JWT ?

Date : 21 avril 2022
Speaker : Karim Pinchon (Ornikar)
Format : conférence (45mn)

Une grande partie des dévs présents dans la salle connait JWT. Bien souvent connaitre JWT en surface suffit. Cette conférence vise à approfondir nos connaissances.

Concepts de base

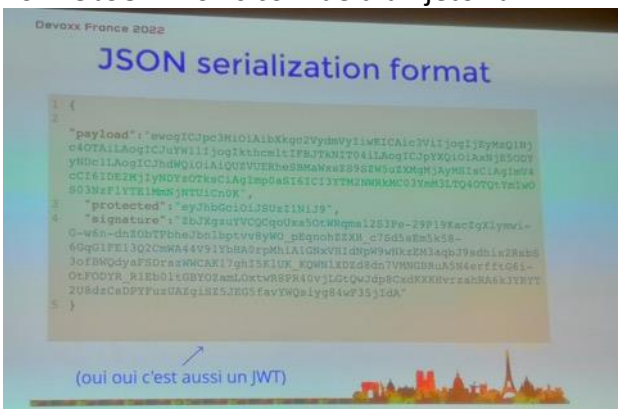
Jeton / token : **chaine de caractères**, pouvant être utilisée pour l'authentification et l'autorisation.
2 types de token : **référence** et **valeur** (porte de l'infos)
JWT fait partie des tokens à valeur
JSON Web Token : token compact pour transmettre des données
A sa création, JWT s'est appuyé sur les Simple Web Tokens.

Cryptographie
- Basée sur des mathématiques
- Apporte de la confiance
- Communication
- Canal non sûr

Opérations cryptographiques : signature numérique, chiffrement, hachage, MAC/HMAC pour Message Authentication Code et Hash-Based MAC

A ne pas confondre avec l'encodage qui est un simple changement de caractères.
Les sites jwt.io et token.dev permettent de jouer avec un token JWT
Un token JWT est encodé en base 64 et contient 3 parties séparées par des points : 2 objets JSON et un HMAC
La signature permet d'assurer l'intégrité du payload.

Forme JSON moins connue d'un jeton JWT :



Ce format permet d'ajouter plusieurs signatures.

L'en-tête contient tous les moyens cryptographiques ayant été utilisés pour générer le token : algo, type de clé …

Le payload peut être séparé en 3 parties :

1. Registererd claims : iss, aud, sub, iat, nbf, exp, jti
2. Public claims : augmente le jeton (notamment dans le cas d'OIDC) avec l'iss, le sub, aud, exp, name, given_name, family_name, email … Ces claims sont définis par l'IANA
3. Private claims : tout ce qu'on veut

JOSE : Javascript Object Signing and Encryption
Famille de RFC définissant comment faire circuler du JavaScript de manière sécurisée.
Contient :

- JSON Web Signature
- JSON Web Encryption (JWE)
- JSON Web Algorithms (algos pour chiffrer et signer les jetons)
- JSON Web Key (comme faire transiter les clés pour chiffrer, vérifier la signature)

JSON Web Encryption (JWE)
Un JWE est plus volumineux et contient 5 parties : header, clé chiffré, vecteur d'initialisation (optionnel), payload chiffré, tag d'authentification.
Pour générer un JWE on commence par générer un jeu de clés.

Gestion des clés avec JSON Web Key (JWK)
Représentation d'une clé symétrique en 3 parties : alg, kty, k
Le kid correspond à l'identifiant de la clé. On peut insérer ce kid dans un token JWT.
JWK peut référencer l'URL vers la clé ou l'ensemble des clés publiques.

Usages des token JWT

Premier usage de JWT **: jeton d'API**
L'API qui reçoit le jeton valide l'authenticité du jeton.
JWT est utilisé dans **OAuth 2**. L'**Access Token** donne à son porteur l'autorisation d'accéder à des ressources. Il peut être sous la forme d'un JWT.
OpenID Connect (OIDC) défini l'**Id Token** comme un token JWT.

Autre usage : **session stateless**
Les informations sont placées dans un jeton par le serveur puis renvoyées au serveur.

Usage « custom » non détaillé.

Les attaques

Contenant bien souvent des données d'authentification, les token JWT peuvent être des vecteurs d'attaque.

Technique dite du jeton non sécurisé : l'attaquant passe à « none » le nom de l'algorithme : permet d'avoir des jetons non signés. La signature du token est ignorée.
Cette faille a été exploitée par des librairies laissant passer cette attaque.

RSA Public Keys as shared Key : jeton généré avec une clé RSA privée. L'attaque consiste à manipuler le token en changement la clé asymétrique par une clé symétrique, en l'occurrence la clé publique.

Brute force : aucun mécanisme protège JWT de cette attaque. La taille des clés joue un rôle considérable pour s'en prémunir.

Modification des données chiffrées : un attaquant doué peut modifier des données du token sans connaitre la clé de chiffrement. Il pourrait par exemple modifier le flag admin de false à true.

Le paradoxe de JWT

JWT porte ses données. Pour valider le jeton, on exploite des données du header. Mais on ne peut pas faire confiance au jeton sans avoir vérifier son intégrité.

Quelques conseils de sécurité

- Le secret : protégé, robuste (suffisamment long), politique de rotation des clés à mettre en place.
- Ne pas tout accepter : ne pas exploiter le header du jeton : vérifier côté serveur les algorithmes envoyés dans le jeton et refuser le jeton. Même chose pour les URL et n'accepter que les domaines de confiance
- Valider les claims : expiration du jeton, émetteur connu
- Préférer l'asymétrique, surtout si on a un partenaire externe au SI
- Utiliser une librairie existante et éprouvée (et pas du code maison)
- Ne vous battez pas pour révoquer un JWT
- Point sur la confidentialité : un JWT contient du texte en clair. Attention aux claims qu'on positionne
- Eviter de logger les jetons
- Mettre les informations nécessaires et suffisantes : y aller de manière incrémentale, faire attention à la taille du token (pour ne pas casser la requête HTTP)
- Utiliser HTTPS pour faire transiter les JWT

Alternatives à JWT

- **Biscuit** pensé par Clever Cloud (implémentation dispo en Java)
- **Paseto** : Platform-Agnostic Security Tokens
- **Macaroons** de Google
- **Branca**