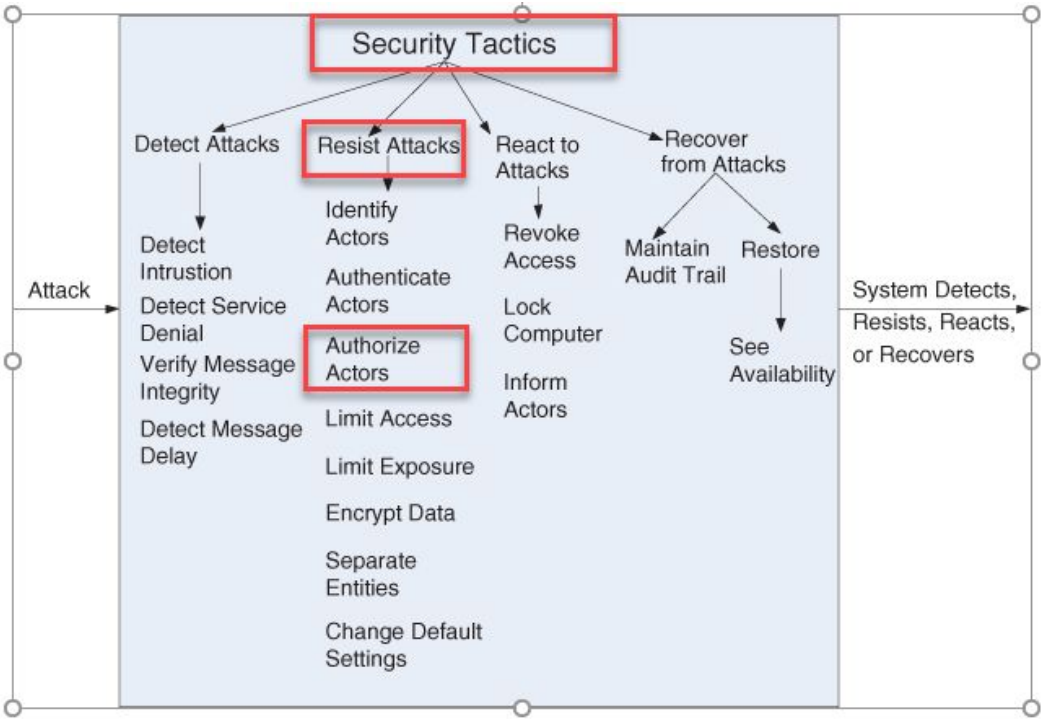
Arquitectura de Software

Tecnología - 2022

Seguridad


Security Tactics
Detect Attacks
Resist Attacks
React to Attacks
Recover from Attacks
Attack
Detect Intrustion
Detect Service Denial
Verify Message Integrity
Detect Message Delay
Identify Actors
Authenticate Actors
Authorize Actors
Limit Access
Limit Exposure
Encrypt Data
Separate Entities
Change Default Settings
Revoke Access
Lock Computer
Inform Actors
Maintain Audit Trail
Restore
See Availability
System Detects, Resists, Reacts, or Recovers

JWT

¿Qué es el token web JSON?

JSON Web Token (JWT) es un estándar abierto (RFC 7519) que define una forma compacta y autónoma de transmitir información de forma segura entre las partes como un objeto JSON. Esta información se puede verificar y confiar porque está firmada digitalmente.

JWT

Aunque los JWT se pueden cifrar para proporcionar también confidencialidad entre las partes, nos centraremos en los tokens firmados.

Los tokens firmados pueden verificar la integridad de los reclamos contenidos en ellos, mientras que los tokens encriptados ocultan esos reclamos de otras partes. Cuando los tokens se firman utilizando pares de claves pública/privada, la firma también certifica que solo la parte que posee la clave privada es la que la firmó.

¿Para que deberíamos usar JSON Web Tokens?

Autorización: este es el escenario más común para usar JWT. Una vez que el usuario haya iniciado sesión, cada solicitud posterior incluirá el JWT, lo que permitirá al usuario acceder a rutas, servicios y recursos permitidos con ese token. El inicio de sesión único es una función que se usa ampliamente en JWT en la actualidad, debido a su pequeña sobrecarga y su capacidad para usarse fácilmente en diferentes dominios.

Intercambio de información: los tokens web JSON son una buena manera de transmitir información de forma segura entre las partes. Debido a que los JWT se pueden firmar, por ejemplo, utilizando pares de claves pública/privada, puede estar seguro de que los remitentes son quienes dicen ser. Además, como la firma se calcula utilizando el encabezado y la carga útil, también puede verificar que el contenido no haya sido alterado.

Partes de JWT

“Header”

Header

El encabezado *generalmente* consta de dos partes: el tipo de token, que es JWT, y el algoritmo de firma que se utiliza, como HMAC SHA256 o RSA.

Por ejemplo: `{ "alg": "HS256", "typ": "JWT" }`

Luego, este JSON está codificado en **Base64Url** para formar la primera parte del JWT.

En la criptografía, un **HMAC** (a veces expandido como **código de autentificación de mensajes en clave-hash** o **código de autenticación de mensaje basado en hash**) es una construcción específica para calcular un código de autentificación de mensaje (MAC) que implica una función hash criptográfica en combinación con una llave criptográfica secreta.

Partes de JWT

“Payload”

Payload

La segunda parte del token es el Payload(carga útil), que contiene los “claims o privilegios”. Los “claims” son declaraciones sobre una entidad (normalmente, el usuario) y datos adicionales. Hay tres tipos de “claims o privilegios”: *registrados* , *públicos* y *privados* .

- **Reclamos registrados** : se trata de un conjunto de reclamos predefinidos que no son obligatorios pero se recomiendan para proporcionar un conjunto de reclamos útiles e interoperables. Algunos de ellos son: **iss** (emisor), **exp** (tiempo de caducidad), **sub** (sujeto), **aud** (audiencia), entre otros .
- **Claims públicos** : estos pueden ser definidos a voluntad por aquellos que usan JWT. Pero para evitar colisiones, deben definirse en el Registro de tokens web JSON de IANA o definirse como un URI que contenga un espacio de nombres resistente a colisiones.
- **Claims privados** : Son los reclamos personalizados creados para compartir información entre partes que acuerdan usarlos y no son claims *registrados* ni *públicos* .

Partes de JWT

“Signature”

Firma: se usa para verificar que el token es válido, e aquí el quid de la cuestión!

Se construye de tal forma que podemos verificar que el remitente es quien dice ser y que el mensaje no fue alterado en el camino.

Se construye con el HASH(HMACSHA256) de:

+ Codificación en base 64 del header
+ Codificación en base 64 del payload
+ y un “Secret” establecido por la aplicación.

Cuando recibimos un token JWT

Debemos verificar siempre que sea válido!, si no lo es debemos rechazar la petición.

Si fuera válido se puede utilizar la información que porta.

Consideraciones

JWT va en formato “abierto” base 64, es un formato “codificado” y no “cifrado” por lo que se puede se invita a utilizar otro soporte seguro como el https.