Refresh Tokens

O que é um token?

Tokens são pedaços de dados que carregam apenas informações suficientes para identificar a identidade ou autorização de um usuário para executar uma ação. Ou seja, os tokens são artefatos que permitem que os sistemas aplicativos executem o processo de **autorização** e **autenticação**.

Autenticação: é o processo de confirmação da identidade de um usuário ou dispositivo. Durante o processo de autenticação, uma entidade geralmente conta com alguma prova para se autenticar, ou seja, um fator de autenticação como por exemplo o fluxo de Login.

Autorização: refere-se ao processo de verificação de quais recursos o usuário ou dispositivo pode acessar ou quais ações podem realizar, ou seja, seus direitos de acesso

Token de acesso

Quando um usuário efetua login, o servidor de autorização emite um **token de acesso**, que é um artefato que os aplicativos clientes podem usar para fazer chamadas seguras para o servidor.

Quando um aplicativo cliente precisa acessar recursos protegidos em um servidor em nome de um usuário, o token de acesso permite que o cliente sinalize ao servidor que recebeu autorização para executar determinadas tarefas ou acessar determinados recursos.

Por que gerar um novo token?

Por questões de segurança, normalmente as aplicações não querem que o usuário fique para sempre logado. Para isso, as estratégias de autenticação normalmente são acompanhadas de algum **prazo de expiração**.

Motivo: o token de acesso é um token de portador, ou seja, aqueles que possuem o token podem usá-lo. O token de acesso atua como um artefato de credencial para acessar recursos protegidos em vez de um artefato de identificação. Usuários mal-intencionados com o token de acesso poderiam usar para acessar recursos protegidos apresentando esses tokens diretamente ao servidor.

Como mitigar

Por isso, é fundamental ter estratégias de segurança que minimizem o risco de comprometer os tokens de acesso. Um método de mitigação é criar **tokens de acesso que tenham uma vida útil curta.** ⏳

Quando trabalhamos com tokens JWT, conseguimos configurar facilmente a expiração desse token que normalmente é um tempo curto (alguns minutos).

Estratégias

Desloga o usuário: quando um **token de acesso** expira o aplicativo cliente pode solicitar que o usuário efetue o login novamente para obter um novo token de acesso.

Expira Refresh Token: quando um token de acesso expira a aplicação revalida o token de acesso utilizando um refresh token. No entanto, **se o refresh token expira**, o usuário obtém um erro 401 e será deslogado para fazer login novamente.

Refresh Token Rotation: o servidor de autorização emite um **token de atualização rotativo** que permite gerar um novo token de acesso e mantém o usuário sempre logado (a não ser que deletado o refresh token do lado do servidor).

Token Rotativo

O **Refresh Token Rotation** é descartado sempre que é utilizado para gerar um novo token de acesso. E um novo token de acesso e refresh token será gerado.

A ameaça de acesso ilegítimo é reduzida, pois os tokens de atualização são continuamente trocados e invalidados. E se o refresh token for comprometido, como ele é um registro no banco de dados, é fácil invalidá-lo, o que normalmente não acontece com o token JWT.

Estratégias

Expira Refresh Token

Traz uma proteção maior mas uma UX menor

Exemplo

Refresh Token Rotation

Traz melhor UX em prol de uma proteção um pouco menor.

Exemplo

Se você desejar aprofundar ainda mais na implantação do lado do servidor, eu recomendo a trilha de Node.js

Refresh Tokens