
Newton
Quem se prepara, não para.

Arquitetura de Aplicações Web

5º período

Professora: Michelle Hanne

Sumário

- Autenticação e Autorização
- Autenticação HTTP
- JSON Web Token
- OAuth

Autenticação e Autorização

Newton
Quem se prepara, não para.

- O conceito de autenticação é definido por Metz (1999) como sendo o processo que envolve a verificação da identidade de um usuário ou serviço em relação a um determinado recurso.

De acordo com Andrade (2010), no processo de autenticação é característico o fato de que o usuário (ou serviço) que realiza o acesso a um determinado recurso é detentor de uma informação que é única a ele e serve como uma credencial de acesso para distingui-lo dos outros usuários.

Autenticação HTP

Newton
Quem se prepara, não para.

O protocolo HTTP (Hypertext Transfer Protocol) é **empregado na comunicação e transferência de dados na web** e conta com recursos de autenticação de usuários de modo a permitir acesso a recursos protegidos.

O recurso de autenticação que conta com o protocolo HTTP faz uso da inclusão de dados no **header (cabeçalho) tanto na requisição realizada pelo usuário**, por meio de um cliente, **quanto na resposta a essa requisição enviada pelo servidor**.

Existem dois tipos de autenticação no **HTTP, a saber: basic e digest.**

Autenticação HTP

Newton
Quem se prepara, não para.

A Base64 consiste em um método de codificação de dados utilizado para transferência de conteúdo na internet, sendo utilizada quando existe a necessidade de codificar dados binários que precisam ser armazenados e transferidos por mídia projetada para trabalhar com dados de texto. Além disso, essa codificação garante que os dados permaneçam intactos sem que ocorram alterações durante o transporte. A codificação Base64 é normalmente utilizada em várias aplicações, incluindo o e-mail via MME e armazenamento de dados complexos em XML e JSON.

O HTTP também define o método **digest** que, conforme Andrade (2010), consiste em um método de autenticação que realiza a criptografia dos dados antes do envio deles pelo cliente através da rede, por meio de uma função hash irreversível, tais como Message-Digest Algorithm 5 (MD5) ou Secure Hash Algorithm (SHA).

JWT

Newton
Quem se prepara, não para.

Elemento JWT	Exemplo

Headers	{
	“alg”: “HS256”,
	“typ”: “JWT”
	}

Payload	{
	“sub”: “3581348752”,
	“name”: “John John”,
	“admin”: true
	}

Signature	TJVA9duiedbBab30RMHrHDcEfxjoYZgeFONFh7HgQ

JWT

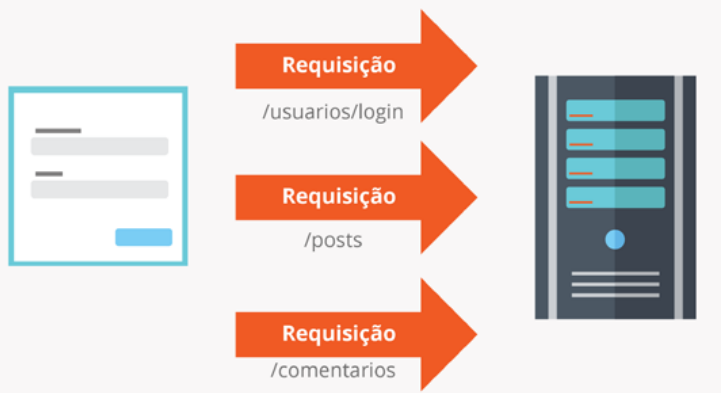
Newton
Quem se prepara, não para.

De acordo com Montanheiro et al. (2017), o JSON Web Token (JWT) consiste em um padrão aberto que tem como objetivo definir um modo compacto e independente que pode ser enviado dentro de um cabeçalho HTTP, contendo informações sobre o usuário para a transmissão segura de informações entre cliente e servidor por meio de um objeto JSON

JWT

Fonte: DevMedia, 2019.

JWT



De posse do token autenticado, de acordo com a DevMedia (2019), o cliente possui acesso aos endpoints da aplicação que antes eram restritos. A fim de realizar esse acesso, é necessário informar esse token ao header autorizado da requisição.

Atualmente estão disponíveis inúmeras bibliotecas que implementam o padrão JWT nas mais diferentesblinguagens de programação, dentre elas a Python, Ruby, JavaScript, Java, Perl, .NET, PHP e Node.js.

Fonte: DevMedia, 2019.

Vulnerabilidades

Newton
Quem se prepara, não para.

Dentre as dúvidas mais comuns acerca desta técnica está a segurança do token, afinal ele é o ponto mais exposto na técnica e sequestro de tokens é a forma mais comum de tentar burlar este mecanismo.

Usando SSL (exemplo Lets Encrypt que é gratuito) esse risco de captura diminui consideravelmente uma vez que a conexão está criptografada.

O token gerado pelo **JWT é codificado em base64**, uma representação textual de um conjunto de bytes. Ao usar um decodificador, verá as três partes que o compõem sendo que a única ilegível é a terceira onde **temos a assinatura digital do servidor**, atestando que aquele token foi gerado **corretamente pelo servidor,** o que impede que tokens fake se passem por tokens reais.

Fonte: https://owasp.org/www-community/attacks/Session_hijacking_attack

Vulnerabilidades

Newton
Quem se prepara, não para.

No entanto, para essa assinatura é necessário um segredo/secret. Esse segredo é usado tanto para assinar quanto para verificar a assinatura (e autenticidade) de um token.

Se você tem apenas uma webapi que usa o JWT, isso é bem tranquilo. Agora, se você possui diferentes microservices e todos eles precisam de autenticação/autorização via JWT, então você tem um problema pois:

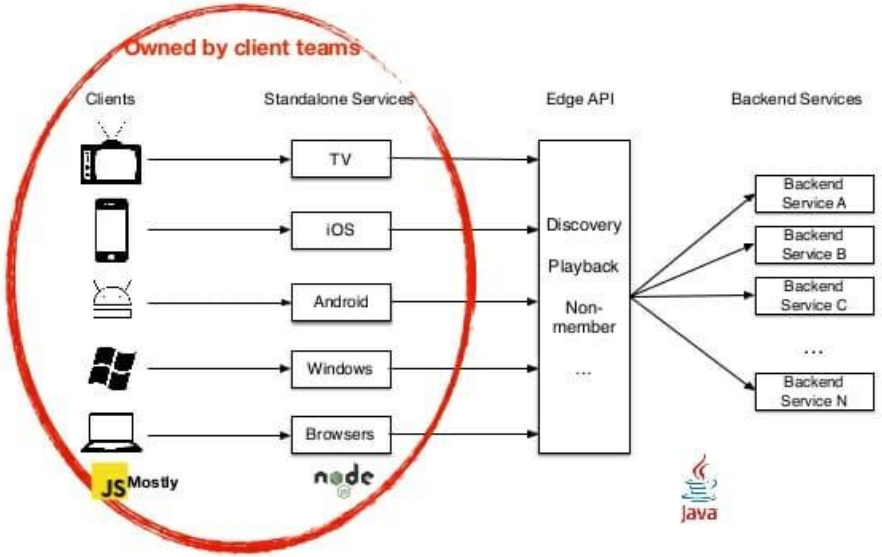
- **ou você faz com que o cliente se autentique em cada um dos microservices que vai usar;**
- **ou você compartilha o secret entre todos eles.;**

Fonte: https://owasp.org/www-community/attacks/Session_hijacking_attack

API Gateway

Newton
Quem se prepara, não para.

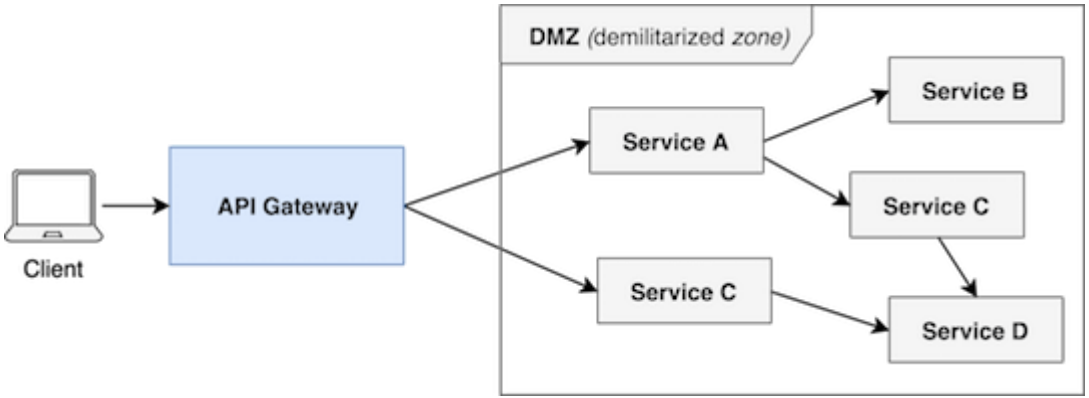
Uma solução possível para este problema é usar um API Gateway na frente de todos microserviços. Daí todos eles confiariam no gateway, sem ficar pedindo ou verificando tokens.



É exatamente o que fez a Netflix, que construiu um API Gateway em Node.js para ficar na frente das suas centenas de APIs Java, como mostra o diagrama

Fonte: https://www.luiztools.com.br/post/api-gateway-em-arquitetura-de-microservices-com-node-js/

API Gateway



Nesta arquitetura, o API Gateway recebe e roteia as requisições para microservices dentro de uma DMZ (Zona Desmilitarizada ou Zona de Perímetro), um conceito de segurança da informação para proteger a rede da empresa dos serviços e vice-versa.

Fonte: https://www.luiztools.com.br/post/api-gateway-em-arquitetura-de-microservices-com-node-js/

OAuth

Newton
Quem se prepara, não para.

O ***Open Authorization***, ou apenas OAuth, consiste em um padrão aberto que tem como objetivo permitir aos usuários de um site acesso, por meio de uma aplicação externa, **aos seus recursos privados sem a necessidade de ter que compartilhar senhas ou logins**. OAuth criado em 2003 e OAuth 2.0 a partir de 2009.

Segundo **Xavier e Carvalho (2011), o OAuth** tem como foco a **autorização e não a autenticação**. Nesse sentido, ele prevê níveis de autorização que o **usuário pode aceitar ou não**. Ainda segundo os autores, o OAuth foi construído baseado nos padrões dos proprietários **AuthSub do Google**, **BBAuth do Yahoo e o API Auth do Flickr**, podendo também ser caracterizado como um protocolo.

OAuth

Newton
Quem se prepara, não para.

- As especificações do OAuth são constituídas por duas partes**: a primeira delas define um processo no navegador que tem como objetivo redirecionar o usuário final para a autorização dos seus recursos pelo cliente**.
- Já a segunda parte das especificações do OAuth tem como **objetivo definir um método para a realização de requisições HTTP autenticadas por meio de dois conjuntos de credenciais**, um deles destinado à identificação do cliente e o outro destinado à identificação do proprietário do recurso a ser requisitado.

OAuth

Newton
Quem se prepara, não para.

De acordo com Xavier e Carvalho (2011), o protocolo segue o seguinte fluxo:

- **1. Token de requisição:** quando o usuário entra na aplicação consumidora, ela requisita ao servidor um token de requisição. Quando a aplicação consumidora recebe o token, ela redireciona o usuário para a tela de autenticação do servidor. O token de requisição é enviado, bem como um link para redirecionamento assim que o usuário se autenticar.
- **2. Autenticação e autorização**: ao ser redirecionado para a tela de autenticação do servidor, o usuário é requisitado a se autenticar. Uma vez autenticado, o usuário recebe um questionamento acerca da autorização para a aplicação consumidora.
- **3. Redirecionamento à aplicação consumidora:** caso o usuário autorize o acesso, o servidor marca o token de requisição, enviado no passo 1, como autorizado. O usuário então é redirecionado para o URL previamente informado pela aplicação consumidora.

OAuth

Newton
Quem se prepara, não para.

De acordo com Xavier e Carvalho (2011), o protocolo segue o seguinte fluxo:

- **4. Token de acesso:** quando o fluxo retorna à aplicação consumidora, ela se encarrega de fazer um intercâmbio do token de requisição por um token de acesso. O token de requisição tem como único objetivo a aprovação pelo usuário. Entretanto, o token de acesso é destinado às requisições dos recursos privados.
- **5. Acesso aos recursos privados:** com o token de acesso, a aplicação consumidora pode consultar todos os recursos privados, aos quais o usuário concedeu permissão.

JWT e OAuth

Newton
Quem se prepara, não para.

É possível realizar a implementação do **JSON Web Token** em uma determinada aplicação por meio do serviço **Auth0**, o qual, conforme Montanheiro *et al.* (2017), implementa o JWT em seu código e libera uma API na qual o cliente se cadastra por meio do serviço, sendo enviadas para a aplicação apenas informações do usuário e senha, e outros dados ficando guardados de forma segura pelo Auth0

Referências

- ANDRADE, Marcos Tadeu de. **Mecanismos de autenticação e autorização em redes sociais virtuais: o caso futweet**. Dissertação de Mestrado (Mestrado em Ciência da Computação) - Universidade Federal e Pernambuco, Recife, 2010. Disponível em: *https://repositorio.ufpe.br/handle/123456789/2318* Acesso em: 25 jul. 2020.
- DEVMEDIA. **Como o JWT funciona**. Disponível em: *https://www.devmedia.com.br/como-o-jwt-funciona/40265*. Acesso em: 24 jul. 2020.
- METZ, Cristopher. **AAA PROTOCOLS:** authentication, authorization, and accounting for the internet. IEEE Internet Computing, 1999.
- MONTANHEIRO, Lucas Souza *et al.* Utilização de JSON Web Token na autenticação de usuários em APIs
- REST. *In:* XIII ENCONTRO ANUAL DE COMPUTAÇÃO, 2017. Goiânia. **Anais do XIII Encontro Anual de**
- **Computação EnAComp**. Goiânia: UFG, 2017. Disponível em: *https://www.enacomp.com.br/biblioteca-de-*
- *-publicacoes/publication-details.php?id=209* Acesso em: 24 jul. 2020.
- XAVIER, Otávio C.; CARVALHO, Cedric L. de. **Desenvolvimento de Aplicações Sociais A Partir de APIs em Redes Sociais Online.** [S. l.], 2011. Disponível em: *https://bit.ly/2XIH910*. Acesso em: 8 ago. 2020.

- **Título do artigo:** Desenvolvimento de aplicações sociais a partir de APIs em redes sociais online - **Link:** *https://bit.ly/3q05zu1*
- **Título do artigo:** Utilização de JSON Web Token na autenticação de usuários em APIs REST - **Link:** *https://www.enacomp.com.br/biblioteca-de-publicacoes/publication-details.php?id=209*