Autenticação

Roberto Rocha robrocha@deloitte.pt

Autenticação – Introdução

- Forma comum de incorporar segurança para as aplicações
- Como saber quem ou que sistema está a aceder à aplicação / aos dados
- Não confundir com Autorização (Authorization)
  - Autorização apenas trata a verificação do que pode ou não aceder e não acerca de quem é
- Não é obrigatório para nenhuma aplicação ou serviço, no entanto, tipicamente os requisitos obrigam à sua implementação
- Normalmente, quando queremos que utilizadores utilizem métodos de CRUD (manipulem dados), devemos implementar Autenticação

Autenticação – Implementação típica

- Implementação típica HTTP / HTTPS funciona da seguinte forma
  - Utilizador tenta aceder a um conteúdo que não pode aceder sem autenticação
  - A aplicação faz um pedido de autenticação (pode reencaminhar para um ecrã de login)
  - O utilizador preenche os dados de login e submete e os dados são enviados através de POST
  - O servidor irá receber os dados e tentará autenticar com os dados recebidos
  - Caso consiga autenticar, irá armazenar alguns dados
    - Estes dados podem ser cookies / session, tokens, signatures, etc..
  - O utilizador irá agora conseguir aceder ao conteúdo, caso tenha permissão.

Autenticação – Métodos

- Autenticação baseada em Cookies
  - Método default de autenticação de utilizadores
  - Cliente envia credenciais ao servidor
  - Servidor verifica as credenciais
  - Cria sessão, armazena SessionId no servidor e envia ao cliente através de set-cookie
  - Essa cookie é depois utilizada nos pedidos do Cliente
  - Quando a sessão é terminada, a cookie é eliminada do cliente e a sessão é eliminada do servidor

Autenticação – Métodos

- Autenticação baseada em Tokens
  - Muito popular em APIs RESTful
  - Muitas implementações distintas, mas a utilização mais comum é JSON Web Token (JWT)
  - Ao receber as credenciais do cliente, o servidor gera um JWT assinado que contém a informação do utilizador
  - O Token nunca é armazenado no servidor, mas sim no cliente e é enviado em cada pedido.
    - É armazenado no cliente em session, cookies ou local storage
  - Ao receber o Token, o servidor irá descodifica-lo e verificar os dados contidos.

Autenticação – Métodos

- Acessos Third-Party (OAuth, API-Token, etc)
  - Métodos utilizados tipicamente quando existe a necessidade de expor uma API para aplicações de terceiros
  - API-Token
    - Igual a JWT Token, o token é enviado no header Authorization e é tratado por um Handler da API
  - Oauth – Open Authentication
    - Protocolo que permite a uma aplicação autenticar-se num servidor como um utilizador

Autenticação – Métodos

- OpenID
  - Protocolo baseado em HTTP
  - Utiliza um provedor de identidade para validar um utilizador
  - Permite Single SignOn

  - É o que nos permite fazer login com Google, Facebook, Twitter, etc em diversas plataformas
  - Por vezes implementado em conjunção com Oauth 2.0 por questões de segurança

Autenticação – Métodos

- SAML - Security assertion markup language
  - Também utiliza um provedor de identidades
  - Baseado em XML e mais flexível
  - Permite implementar Single SignOn
  - Login unificado, utilizando um URL do provedor de Identidade
  - O provedor responde (XML) com informação para encaminhar o utilizador à pagina pretendida

  - Utilizado por Oracle (OIM), Office365, etc.

Autenticação – Token

- Basicamente, permite ao utilizador:
  - Autenticar-se apenas uma vez, através de username e password
  - Gera um Token
  - Concede permissão para aceder aos recursos utilizando apenas esse Token
- Mais seguro que as password comuns
  - Mais complexa e não é gerada por um utilizador (o que adiciona vulnerabilidades)
  - Encriptado
- Com este tipo de autenticação, o servidor consegue verificar se foi feita alguma alteração ao Token e bloqueia o acesso através desse Token.
- Apenas é necessário transmitir o Token e não estar sempre a autenticar através de username / password

Autenticação – Vantagens de Token

- Não obriga manter os dados em Base de Dados
- Cross-Domain /CORS
  - Utilizando Cookies, CORS normalmente bloqueia a transmissão de dados entre diferentes domínios. Utilizando Autenticação com Token, AJAX pode fazer chamadas para outros servidores pois a informação do utilizador é transmitida no header HTTP
- “Statelessness”
  - O servidor não tem de guardar dados de sessão, o Token contém toda a informação suficiente para o servidor validar a informação do utilizador.
- CDN – Content Delivery Network
  - Todo o conteúdo estático pode ser fornecido por um CDN, e apenas a API é comunicada pelo servidor
- Desacoplamento
  - Não obriga a associação a um tipo de autenticação em particular, o Token pode ser gerado em qualquer lugar, podendo assim a API ser invocada a partir de qualquer lugar com uma única forma de autenticar as invocações
- Mobile Ready
- CSRF – Cross-Site Request Forgery
  - Não é possível desde que o Token não seja guardado numa Cookie
- Performance
  - Mais rápido de validar

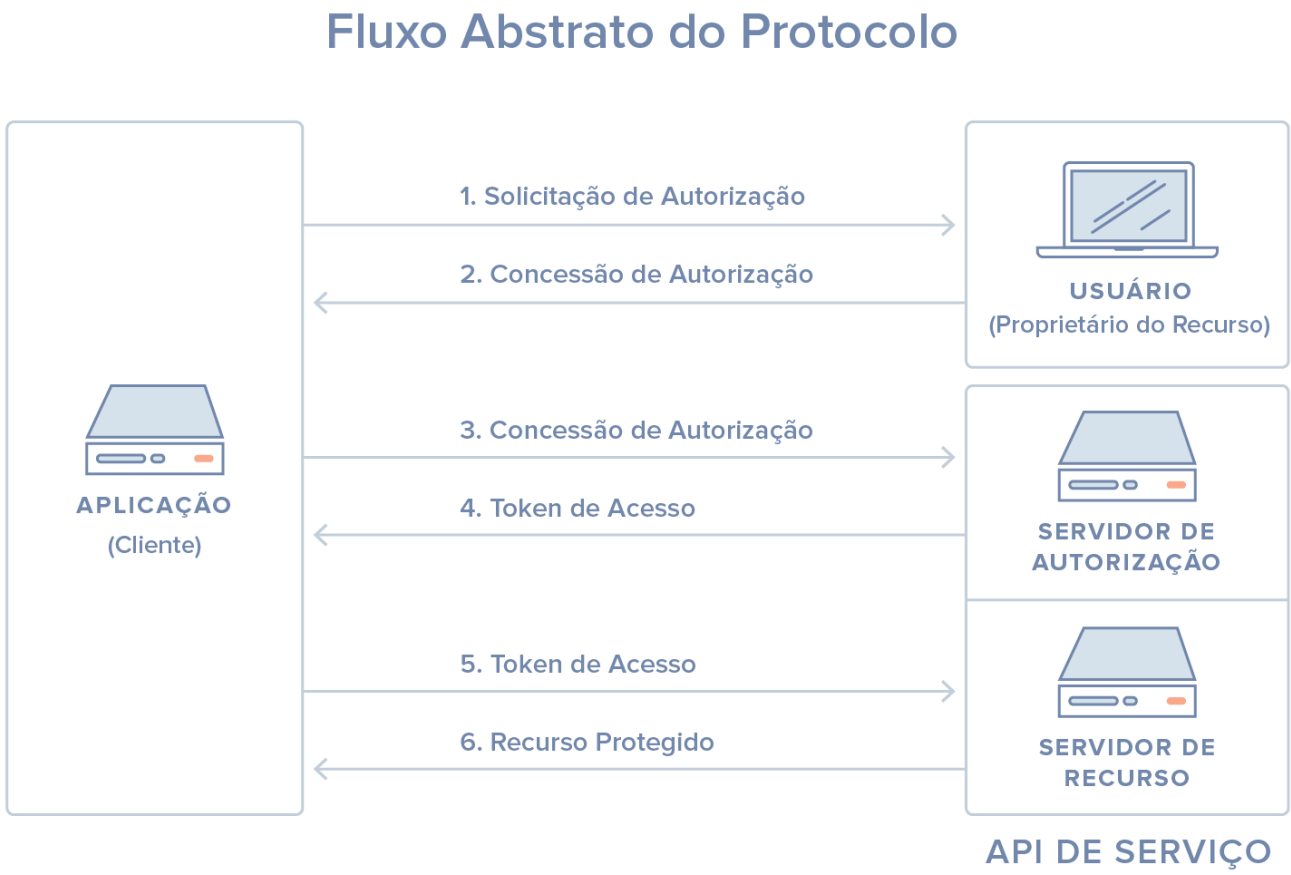
Autenticação – Desvantagens de Token

- Vulnerável se o Token não é guardado de forma segura ou é partilhado
  - O Token irá permitir efetuar todos os pedidos como se se tratasse do utilizador que fez gerar o Token
- O servidor não controla as sessões
  - Não pode forçar logout por exemplo
- Não temos dados de utilizador, limita algumas funcionalidades
  - P.E.: Notificações
- Adiciona overhead aos pedidos
  - O Token é incluído no header dos pedidos

Autenticação – oAuth

- Protocolo aberto que permite Autorização segura, com métodos simples e standard de Web, mobile e aplicações Desktop.
- Permite o login através de contas Facebook, Google, Twitter, etc.
- Fornece aos clientes “acesso seguro delegado” aos serviços do servidor
- Utiliza Tokens de acesso emitidos para clientes de terceiros, mediante autorização do servidor, com a aprovação do proprietário do recurso
- Distindo de OpenID, mas OpenID Connect é uma camada de apresentação construída em cima de OAuth 2.0
- Existe:
  - Resource Owner – Entidade que concede acesso a um recurso
  - Resource Server – Servidor que contém os recursos protegidos (acedido através de Token)
  - Client – Aplicação requesitando recursos protegidos, através da identificação do dono
  - Authorization Server – Servidor que valida autenticação e emite Token de acesso ao Cliente

Autenticação – oAuth



Autenticação – Vantagens de oAuth

- Facilidade de utilização
  - Para o utilizador comum é mais simples, ao aceder a um recurso não autorizado numa página, fazer login através de outra plataforma (P.E.: Login with Facebook)
- Poupa tempo
  - Poupa processos de registo, simplifica logins, etc
- Privacy
  - Apesar de poder partilhar dados de acesso entre diferentes aplicações, não permite aceder a qualquer outra informação dessa outra aplicação, a não ser que tenha permissão para tal
- Seguro
- Controlo de acessos
- Etc.

Autenticação – Desvantagens de oAuth

- Partilha de dados bases entre plataformas (ao fazer login com Facebook por exemplo)
- Não é assim tão uniforme
  - Não é usado globalmente
- Uma má implementação poderá deixar uma solução vulnerávekl a ataques informáticos
- Etc.

Autenticação – Exercícios