8.Ověřování identity v prostředí internetu

* OpenID, poskytovatelé ověření, access\_token

# Jméno

* Unikátní údaj, nestačí samostatně, snadno se prozradí, proto je k němu potřeba heslo

# Heslo

* Tajný údaj, určuje míru bezpečnosti, nestačí samostatně (musel by být unikátní)
* Lze prolomit pomocí sociálního inženýrství nebo pomocí bruteforce
* Passphrase = místo hesla se použijedlouhá zapamatovatelná fráze

# Dvoufázové ověřování

* Dodatečný kód zaslaný na email, SMS nebo použití Authenticatoru

# Biometrické ověřování

* Otisk prstu, sken rohovky
* Není vhodné jako jediný údaj, z unikátního otisku prstu se může stát neunikátní hash

# OAuth 2.0

* Open Authorization
* Standard navržený k tomu, aby umožnil webům nebo aplikacím přístup ke zdrojům jiným aplikací jménem uživatele
* Dovoluje uživatelům sdílet specifická data s aplikací ale zároveň držet ostatním informace v tajnosti
* Aplikace může například použít OAuth2 k získání premise od uživatele k uložení dat na jejich Google Disk
* Token: náhodný kód identifikující uživatelova oprávnění
* Scope – určení, o kterou konkrétní část dat usilujeme
* Role:
  + User – uživatel snažící se přistoupit k nějakému zdroji dat (k Resource)
  + Resource – chráněná data uložená na serveru
  + Resource Owner – uživatel vlastnící nějaká data, může k nim dát přístup jinému uživateli
  + Client – aplikace skrz kterou User k datům přistupuje
  + Resource Server – server, na kterém jsou data uložena
  + Authorization Server – po úspěšné autentifikaci předává klientovi access token

# OpenID

* Standard popisující decentralizovaný způsob autentizace uživatelů
* Poskytovatel služeb nemusí mít na své straně vlastní systém pro autentizace
* Přihlášení přes Steam, Facebook, Spotify…
* Token vydaný OAuth2 je náhodný řetězec, neobsahuje žádné informace; autorizován je klient
* OpenID do tokenu zakóduje navíc informace o uživateli
* Dnes obvykle JWT – JSON (JavaScript Object Notation) Web Token, uvnitř něj je mimo jiné zakódováno:
  + sub – subject – ID uživatele
  + iss – issuer – kdo vydal token
  + aud – audience – klient, kterému byl token poskytnut

# Authorization Grant

* Je použit klientem pro získání tokenu
* Několik druhů:

## Authorization Code

* Krátkodobý autorizační kód, za který uživatel dostane token
* Nejběžnější, pro server-side aplikace

## Implicit

* Zjednodušený mechanismus, optimalizováno pro prohlížečové klienty (JavaScript)

## Resource Owner Password Credentials

* Token vyměněn přímo za ID a heslo
* Používat pouze pokud je vysoká důvěra mezi klientem a poskytovatelem dat
* Většinou u mobilních a desktop aplikací

## Client Credentials

* Pro API bez kontextu uživatele

# Refresh Token

* Pro obnovení platnosti tokenu
* Access token mívá platnost kolem 30 minut, Refresh token několik dnů

# Použití access tokenu

axios.get(

"example.com/data",

{"Authorization": "Bearer" + accessToken}

);

