ASP.NET Core Identity
Jak to dělat správně

Michal Altair Valášek
development & security consultant

michal.valasek@altairis.cz
www.altair.blog

MVP Microsoft® Most Valuable Professional

altairis

altair.blog

altairis

Access Control

- Přihlašování, odhlašování uživatelů
- Reset hesla
- Vícefaktorová autentizace
- Ukládání hesel
- …
- **Nejčastější bezpečnostní problém webových aplikací** podle OWASP TOP 10 2021
- https://owasp.org/Top10/

altairis

ASP.NET Core Identity

- Součást ASP.NET Core, starající se o správu uživatelů a operace s nimi
  - Pokračovatel Membership a Role providerů z ASP.NET 2.0
  - Mnohem lépe a moderněji napsaný, architektura vychází ze SOLID principů
- Co řeší
  - Přihlašování, odhlašování
  - Úschovu informací o uživatelích, jejich hesel atd.
  - Přihlašování pomocí externích identity providerů (FB, MSA, Google)
- Co neřeší
  - Uchovávání informací o identitě přihlášeného uživatele
  - Používá se (by default) Cookie Authentication Middleware
  - Založeno na claims architektuře

Vytváření uživatelů

jak to dělat správně

altairis

Vytváření uživatelů

- Uživatele vytváří správce z backendu
  - Je třeba řešit vytvoření prvotního hesla a jeho předání uživateli
  - Ideálně tak, aby ho vytvářející uživatel neznal
  - Nebo použít aktivační link poslaný e-mailem
- Uživatelé se registrují sami
  - Je třeba ověřit e-mailovou adresu
- (První) uživatel je vytvořen při nasazení aplikace
  - Nepoužívat statické heslo

altairis

Ověřování kontaktů

- Důsledně si ověřujte kontaktní údaje
  - E-mailové adresy
  - Telefonní čísla
- Ověřujete:
  - Syntaktickou správnost
  - Funkčnost
  - Souhlas uživatele schránky s registrací
- Tři možné přístupy
  - Bez ověření nelze vůbec založit uživatele
  - Bez ověření je uživatel neaktivní, nemůže se vůbec přihlásit
  - Bez ověření je uživatel omezen, nemůže dělat některé akce – velmi rizikové!

altairis

Ověření e-mailové adresy

- `GenerateEmailConfirmationTokenAsync`
  - Vytvoří token pro ověření e-mailu
  - Zaslat jako součást URL ve zprávě
- `ConfirmEmailAsync`
  - Nastaví s pomocí tokenu adresu jako potvrzenou

- Obdobně lze ověřit telefonní číslo pomocí SMS

altairis

Kontrola, zda uživatel existuje

- Informace, že má někdo s danou e-mailovou adresou účet ve vašem systému může být bezpečnostně kritická
  - Záleží na typu aplikace
  - Např. pokud provozujete seznamku pro záletníky
- Proto byste ji možná neměli vyzrazovat při registraci ani resetu hesla
  - Registraci nového účtu s existujícím emailem považujte za žádost o reset hesla

Politika hesel

jak to dělat správně

altairis

It‘s annoying, so it must be secure…

- Nevyžadujte pravidelnou změnu hesel
  - Heslo se mění, pokud bylo ohroženo, ne preventivně
- Nevyžadujte zbytečně složitá hesla
  - Speciální znaky a podobně
- Nebraňte používání správců hesel
  - Např. blokováním práce se schránkou
- Netrestejte uživatele za zapomenuté heslo
  - Proces jeho resetu musí být bezbolestný

altairis

Rozumná politika pro hesla

- Vyžadujte, aby bylo heslo dlouhé
  - Minimum je 8 znaků, lépe 12
  - Neomezujte maximální délku hesla
  - Podporujte používání „passphrase“ místo „password“
- Podporujte používání správců hesel
  - Nebraňte jejich používání
  - Neblokujte používání schránky
- Odchyťte triviální hesla

altairis

Pwned Passwords Validator

- Validátor, který kontroluje heslo proti databázi uniklých hesel na HIBP
  - https://www.haveibeenpwned.com/
  - API je bezpečné, nesděluje použité heslo externí službě
- Slouží i k odchycení triviálních hesel
  - Alternativa k porovnávání s „X nejčastějších hesel“

- https://github.com/ridercz/Altairis.Services.PwnedPasswordsValidator
- NuGet: `Altairis.Services.PwnedPasswordsValidator`

Reset hesla

jak to dělat správně

altairis

Reset zapomenutého hesla

- Naprosto nezbytná funkcionalita
  - Musí to být pro uživatele snadné!
  - Bohužel také jeden z nejčastějších bezpečnostních problémů
- V zásadě existují dva přístupy
  - Knowledge-based authentication
    - Otázka a odpověď; v praxi velmi nebezpečné
  - Komunikace záložním kanálem
    - E-mail, SMS, osobní návštěva pobočky atd.

altairis

Reset hesla v ASP.NET Identity

- `GeneratePasswordResetTokenAsync`
  - Vytvoří token pro reset hesla
  - Ten zašlete e-mailem, typicky jako součást URL
- `ResetPasswordAsync`
  - Na základě tokenu nastaví nové heslo

altairis

Vlastní token providers

- Generování tokenů pro reset hesla, ověření změny e-mailu, telefonního čísla a podobně
- Výchozí používá ASP.NET Data Protection a Base64 kódování
- Lze napsat vlastní
  - Zcela nezávislé generování (např. náhodné + ukládání)
  - Jiný algoritmus kódování (např. Base32)
  - https://www.altair.blog/2021/03/token-providers

Úschova hesel

jak to dělat správně

altairis

Jak uchovávat hesla

- Pokud možno to nedělejte
  - Nechte úschovu hesel na nějakém jiném IdP
- Když už to musíte dělat, tak se v tom nehrabte
  - Použijte výchozí password hasher v ASP.NET Identity
- Když už se v tom musíte hrabat, tak se v tom hrabte co nejmíň
  - Použijte výchozí password hasher v ASP.NET Identity, kdykoliv můžete

altairis

Jak uchovávat hesla

- Co nepoužívat
  - Plaintext, šifrovaná hesla
  - Libovolný hashovací algoritmus
  - Hashování se solí/HMAC
- Co používat
  - Specializované algoritmy pro tento účel
  - PBKDF2 (je v .NET)
  - Bcrypt, Argon (nejsou tam a nejsou výrazně lepší)
  - S dostatečným počtem opakování

altairis

Kdy se do toho musíte hrabat

- Při importu uživatelů (a hesel) ze starého systému
- Použijte „onion hashing“
  - Starým algoritmem zahashované heslo zahashujte ještě jednou tím novým
  - Uchovávejte všechny potřebné údaje, včetně údaje o verzování
  - Při první příležitosti heslo přehashujte novým algoritmem

Vícefaktorová autentizace

jak to dělat správně

altairis

Autentizační faktory

- **Znalost**
  - *Něco co znáš*
  - Heslo, PIN, passphrase…
- **Vlastnictví**
  - *Něco co máš*
  - Autentizační token, čipová karta, důvěryhodné zařízení…
- **Charakteristika**
  - *Něco co jsi*
  - Otisk prstu, sken sítnice, podpis, obecně biometrika.

altairis

Dvoufaktorová (vícefaktorová) autentizace

- Každý z autentizačních faktorů je samostatně slabý
- Jejich kombinace je ale výrazně silnější
  - Vyžaduje totiž kombinaci různých druhů útoku
  - Útok musí být cílený, což většinou není
- Na webu se obvykle používají dva přístupy
  - Schválení přihlášení pomocí mobilní aplikace
    - Moc neškáluje, použitelné jenom u high stakes aplikací nebo pokud už uživatel vaši aplikaci stejně má
  - Použití jednorázových hesel
    - Univerzální, pokud použijete obecné standardy

altairis

Použití jednorázových hesel

- One-time password (OTP)
  - Generuje se kryptograficky na základě seedu (dohodnutého předem) a counteru
  - EOTP (event-based OTP) používá počet hesel vygenerovaných na žádost
  - TOTP (time-based OTP) používá aktuální čas
- Generují se pomocí univerzálních aplikací
  - Microsoft Authenticator
  - Google Authenticator
  - ...a další
  - Seed se předává jako URI ve 2D kódu
- ASP.NET Core Identity pro OTP má nativní podporu

michal.valasek@altairis.cz

facebook.com/**rider.cz**
github.com/**ridercz**
twitter.com/**ridercz**
linkedin.com/in/**ridercz**

www.**altairis.cz**
www.**rider.cz**
www.**altair.blog**