ZADANIE 3 - DOKUMENTÁCIA

ÚVOD

V tomto zadaní sme implementovali základné prvky autentifikácie pre webovú aplikáciu s dôrazom na bezpečné prihlasovanie a správu hesiel. Cieľom bolo oboznámiť sa s princípmi autentifikácie, medzi ktoré patrí registrácia, prihlasovanie, bezpečné ukladanie hesiel a ochrana pred bežnými útokmi. Pracovali sme s kostrou projektu napísanou v jazyku Python pomocou knižníc Flask, Flask-Login, SQLAlchemy a Flask-WTF. Okrem implementácie jednotlivých funkcií sme sa zamerali na zabezpečenie aplikácie proti brute-force útokom a zabránili sme používaniu bežných slovníkových hesiel.

1. KRITÉRIA POŽADOVANÉ NA HESLÁ

Pri návrhu bezpečnostných kritérií pre heslá sme sa zamerali na minimálnu zložitosť, ktorá zabezpečí odolnosť proti útokom. Zvolili sme nasledovné kritériá:
- Heslo musí obsahovať aspoň 8 znakov.
- Heslo musí obsahovať aspoň jedno číslo.
- Heslo musí obsahovať aspoň jedno veľké a jedno malé písmeno.
- Heslo musí obsahovať aspoň jeden špeciálny znak (napríklad !@#$%^&*()-_=+[]{}|;:,.<>?/~`).
- Heslo nesmie obsahovať neplatné znaky.

Týmto spôsobom zaisťujeme, že heslá budú dostatočne zložité na to, aby odolali základným útokom. Zložité heslá s vyšším počtom znakov a špecifickými znakmi výrazne zvyšujú čas potrebný na úspešné uhádnutie hesla a znižujú pravdepodobnosť úspešnosti útoku.

2. SYSTÉM BEZPEČNÉHO UKLADANIA HESIEL

Bezpečné ukladanie hesiel je kľúčovou súčasťou každej autentifikačnej aplikácie. Na hashovanie hesiel sme využili funkciu Argon2, ktorú sme aplikovali cez knižnicu PyNaCl. Argon2 je považovaný za jeden z najbezpečnejších hashovacích algoritmov pre heslá, pretože je odolný voči rôznym typom útokov vrátane útokov s použitím grafických kariet.
Hashovanie hesiel pomocou Argon2 zabezpečuje, že aj v prípade získania databázy útočníkom nie je možné jednoducho získať pôvodné heslá. Pri prihlásení sa heslo overuje pomocou uloženej hash hodnoty a ak sa zadané heslo zhoduje s hash hodnotou, prihlásenie je úspešné. Týmto spôsobom chránime heslá používateľov a dodržujeme bezpečnostné štandardy.

3. OCHRANA PROTI BRUTE-FORCE ÚTOKOM

Pre ochranu proti brute-force útokom sme implementovali sledovanie počtu pokusov o prihlásenie podľa IP adresy. Počet pokusov na IP adresu je obmedzený na päť pokusov. Ak je tento limit prekročený, prihlasovanie je zablokované na 15 minút. Týmto spôsobom zabezpečujeme, že útočník nebude môcť rýchlo testovať rôzne kombinácie hesiel. Navyše, časové blokovanie po prekročení maximálneho počtu pokusov ďalej zvyšuje bezpečnosť systému.

4. OCHRANA PROTI SLOVNÍKOVÝM HESLÁM

Pre ochranu proti slovníkovým heslám sme použili súbor bežných hesiel `commonPasswords.txt`, ktorý obsahuje najčastejšie používané heslá. Pri registrácii sa heslo používateľa porovnáva so zoznamom a ak sa zhoduje s niektorým bežným heslom, registrácia je zamietnutá. Týmto spôsobom zabránime používateľom registrovať sa s príliš jednoduchými heslami, ktoré sú náchylné na útoky. Zoznam bežných hesiel je vytvorený na základe štúdií o najčastejšie používaných heslách a poskytuje dodatočnú vrstvu ochrany aplikácie.

ZDROJE

1. Flask Documentation: https://flask.palletsprojects.com/en/3.0.x/
2. Flask-Login Documentation: https://flask-login.readthedocs.io/en/latest/
3. SQLAlchemy Documentation: https://flask-sqlalchemy.readthedocs.io/en/3.1.x/
4. Flask-WTF Documentation: https://flask-wtf.readthedocs.io/en/1.2.x/
5. Argon2 Documentation (PyNaCl): https://pynacl.readthedocs.io/en/latest/