Zadanie 3 – AUTENTIFIKÁCIA

Vypracovali: Samuel Michalčík, Marek Štrba

1. Inštalácia

Na implementáciu webového servera sme použili knižnicu Flask, SQLAlchemy pre podporu databáz, FlaskWTF a Flask-Login.

pip install flask flask-sqlalchemy flask-wtf flask-login

2. Implementácia

V úlohe 1 sa zameriavame na kontrolu zložitosti hesla pri registrácii užívateľa. Za bezpečné heslo považujeme heslo, ktoré nie je prelomiteľné v zmysluplnom čase (za predpokladu, že nepríde k úniku/vyzradeniu/ukradnutiu hesla alebo neočakávanému vývoju vo výpočtových možnostiach). Po prieskume sme zistili, že je potrebné heslo dĺžky aspoň 12 znakov s obsahom malých a veľkých písmen, číslic a špeciálnych znakov.

Number of Characters	Numbers Only	Lowercase Letters	Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters, Symbols
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	Instantly	Instantly	Instantly
6	Instantly	Instantly	Instantly	Instantly	Instantly
7	Instantly	Instantly	1 sec	2 secs	4 secs
8	Instantly	Instantly	28 secs	2 mins	5 mins
9	Instantly	3 secs	24 mins	2 hours	6 hours
10	Instantly	1 min	21 hours	5 days	2 weeks
11	Instantly	32 mins	1 month	10 months	3 years
12	1 sec	14 hours	6 years	53 years	226 years
13	5 secs	2 weeks	332 years	3k years	15k years
14	52 secs	1 year	17k years	202k years	1m years
15	9 mins	27 years	898k years	12m years	77m years
16	1 hour	713 years	46m years	779m years	5bn years
17	14 hours	18k years	2bn years	48bn years	380bn years
18	6 days	481k years	126bn years	2tn years	26tn years

Zdroj: https://images.tech.co/wp-content/uploads/2023/05/19091649/Hive-passwords.jpg

Pri registrácii overujeme pomocou funkcie *password_complexity_check()* tieto požiadavky (aspoň 1 malý znak, aspoň 1 veľký znak, aspoň 1 číslica, aspoň 1 špeciálny znak, aspoň celková dĺžka hesla 12 znakov). Okrem toho sme pri registrácii po overení zložitosti hesla implementovali aj ochranu voči slovníkovým heslám. Na to nám slúži

funkcia *check_common_password()*, ktorá overí, či sa používateľom zadané heslo nenachádza v súbore *passwords.txt*, ktorý sme stiahli ako súhrn najpoužívanejších, jednoduchých a uniknutých hesiel. Jedná sa o skrátený súbor, kde je 10 tisíc hesiel, na internete vieme nájsť súbory s miliónovými počtami, ale pre jednoduchosť a rýchlosť zadanie sme použili skrátenú verziu. V prípade, že heslo spĺňa všetky požiadavky a používateľské meno nie je obsadené, je umožnené sa úspešne registrovať. Pomocou nami implementovanej funkcie *hash_password()* zahashujeme heslo. Na to využívame knižnicu hashlib a os. Pomocou os vygenerujeme tzv. *salt*, ktorý predstavuje náhodných 16 bytov, ktoré sa pridajú k používateľovmu heslu. Následne sa heslo spojené s týmto „saltom“ zahashuje, na čo využívame pbkdf hmac funkciu. Salt a zahashované heslo ukladáme v databáze v hex formáte.

Pri prihlásení dochádza k rovnakému procesu – načítame používateľské heslo z prihlasovacieho formuláru, načítame salt vytvorený pri registrácii a zavoláme funkciu *verify_password(), ktorá salt prevedie z hex formátu,* salt pridá k heslu, heslo so saltom zahashuje a porovná s hashom uloženým pri registrácii v databáze. Ak sa hashe zhodujú a je správne aj používateľské meno, prihlásenie je úspešné, inak zvýši počet neúspešných pokusov o prihlásenie. Ochranu voči brute-force útokom pri prihlasovaní sme zabezpečili pomocou počítania neúspešných pokusov a nastavením blokovania účtu na určitý čas. V databáze ukladáme počet neúspešných prihlásení k danému účtu. AK toto číslo presiahne 5 neúspešných pokusov, účet ostane zablokovaný na 5 minút.