Kolegij Sigurnost računalnih sustava – prva laboratorijska vježba

- ## Opis predanog koda

Napisao sam Python kod koji služi za sigurno pohranjivanje i dohvaćanje lozinki za različite web adrese. Koristi se AES za enkripciju lozinki, HMAC za provjeru integriteta podataka te PBKDF2 za derivaciju ključeva iz master lozinke. Prilikom pokretanja aplikacije, provjerava se integritet baze podataka kako bi se spriječila neovlaštena promjena podataka. Nakon unosa master lozinke koja se koristi za pristup aplikaciji, lozinka se hashira kako bi se izbjegla pohrana u čistom obliku, čime se dodatno štiti privatnost korisnika. Provjera ispravnosti lozinke osigurava se usporedbom hash vrijednosti s prethodno pohranjenom vrijednošću u sistemu. Ključevi za enkripciju (AES) i provjeru integriteta (HMAC) izvode se iz unesene master lozinke i nasumično generiranog salt-a. Ova tehnika osigurava da čak i ako se pristupi hash vrijednosti master lozinke, neće biti moguće dekriptirati lozinke bez pristupa originalnoj lozinki. Korisnik može unositi nove lozinke za web adrese, a svaka lozinka se enkriptira pomoću AES algoritma i dodatno štiti HMAC provjerom integriteta prije pohrane u bazu podataka. U nastavku ću detaljno opisati funkcionalnost navedenih metoda.

calc_db_hash(file): Funkcija koja izračunava SHA-256 hash vrijednost datoteke baze podataka.

db_integrity(file, stored_hash): Funkcija koja provjerava integritet baze podataka uspoređujući izračunati hash s pohranjenim hashom.

mp_check(mp): Funkcija koja provjerava je li unesena master lozinka ispravna.

derive_keys(mp, salt): Funkcija koja derivira ključeve za AES i HMAC pomoću PBKDF2 algoritma. Ključ se dobiva tako što se na master lozinku primjenjuje PBKDF2 zajedno s salt-om.

encrypt(txt, key): Funkcija koja koristi AES za enkripciju teksta. Koristi se GCM (Galois/Counter Mode) za enkripciju i autentikaciju.

decrypt(ciphertext, key): Funkcija koja koristi AES za dekripciju šifriranog teksta.

compute_hmac(data, key): Funkcija koja računa HMAC pomoću SHA-256 hasha.

initialize_db(): Funkcija koja inicijalizira bazu podataka.

add_password(aes_key, hmac_key): Funkcija koja omogućuje korisniku dodavanje nove lozinke za određenu web adresu. Lozinka se enkriptira i HMAC-om štiti integritet podataka.

retrieve_password(aes_key, hmac_key): Funkcija koja omogućuje dohvaćanje lozinke za određenu web adresu. Prvo se provjerava integritet podataka korištenjem HMAC-a prije nego što se pokuša dekriptirati lozinka.

Literatura koja je korištena za pomoć pri pisanju koda navedena je u nastavku…

Literatura:

https://pycryptodome.readthedocs.io/en/latest/src/hash/hash.html

https://pycryptodome.readthedocs.io/en/latest/src/examples.html

https://pypi.org/project/pycryptodome/

https://pycryptodome.readthedocs.io/en/latest/src/hash/hmac.html

https://devcodef1.com/news/1121347/pycryptodome-secure-encryption-with-pbkdf2

https://security.stackexchange.com/questions/8015/what-should-be-used-as-a-salt

https://www.macobserver.com/tips/quick-tip/trust-password-managers-trick/

https://www.freecodecamp.org/news/why-a-little-salt-can-be-great-for-your-passwords/

https://www.howtogeek.com/devops/how-to-properly-store-passwords-salting-hashing-and-pbkdf2/

https://stackoverflow.com/questions/11520126/how-to-create-random-salt-hash-with-crypto

https://thewebdev.info/2022/02/09/how-to-create-random-salt-hash-with-crypto-with-node-js-and-javascript/

https://developer.mozilla.org/en-US/docs/Web/API/Crypto/getRandomValues

https://pycryptodome.readthedocs.io/en/latest/src/random/random.html