README.md

Laboratorijske vježbe iz predmeta Sigurnost računalnih sustava, FER, Sveučilište u Zagrebu

Password manager

U sklopu laboratorijske vježbe razvijen je Password manager u programskom jeziku Python. Program je razvijen koristeći biblioteku pycriptodome i namjenjen je za korištenje u naredbenom retku.

Zahtjevi

Zahtjevi koje rješenje mora zadovoljavati su sljedeći:

1. Povjerljivost lozinki
2. Povjerljivost adresa
3. Integritet adresa i lozinki

Zahtjevi su ispunjeni korištenjem autentificirane šifre i funkcije za derivaciju ključa iz lozinke.

Funkcija za derivaciju ključa

Funkcija za derivaciju ključa korištena je za stvaranje ključa za enkripciju/dekripciju na temelju master passworda. Dodatno, korišten je 16-bajtni nasumično generirani *salt* koji se koristi samo za jedan par enkripcija-dekripcija i čuva se na disku. Korištenje salta povećava entropiju izvora za generiranje ključa i nužan je kako bi algoritam bio potpuno funkcionalan i siguran.

Generira se jedan 32-bajtni ključ za 256-bitnu implementaciju AES algoritma.

Autentificirana šifra

Za povjerljivost podataka i zaštitu integriteta korištena je autentificirana šifra. AES algoritam autentificirane šifre kao ulaz prima nekriptirani tekst (JSON bazu podataka koja sadrži adrese i lozinke) i ključ za kriptiranje. Rezultati izvođenja algoritama su kriptirani tekst, oznaka i random generirani inicijalizacijski vektor (u implementaciji - nonce). Oznaka se izračunava kao HMAC kriptiranog teksta.

Na disk se pohranjuju :

1. inicijalizacijski vektor
2. oznaka
3. kriptirani tekst

Iako sam algoritam kriptiranja i HMAC koriste svaki svoj vlastiti ključ, u korištenoj implementaciji autentificirane šifre potrebno je dostaviti samo jedan.

Dekriptiranje teksta započinje generiranjem ključa na temelju master passworda i spremljenog salta. Time je on jednak ključu korištenom za posljednje kriptiranje.

Kako bi dekriptiranje se dekriptiranje smatralo uspješnim i obavilo potraživanu operaciju nad podacima potrebno je:

1. Validirati oznaku uz pomoć ključa koji je istovjetan onom korištenom za kriptiranje i spremljenog inicijalizacijskog vektora. Validaciju sačinjava ponovno izračunavanje HMAC-a od kriptiranog teksta sa diska i provjera da su oznaka sa diska i nanovo izračunata oznaka sadržajno identične. Ukoliko se pokaže da nisu identične, uzročnik tome su ili pogrešan unos master passworda ili neželjena izmjena sadržaja diska koja nije napravljena kroz sustav enkripcije i dekripcije.
2. Ukoliko je validacija uspješna, dekriptirati šifrat i izvršiti potrebnu operaciju nad podacima

Bitno je napomenuti da je inicijalizacijski vektor jedinstven za pojedini par ključ-čisti tekst. Dakle za svako pojedino kriptiranje će on biti nanovo inicijaliziran i spremljen na disk. Šifrat će se stoga nužno razlikovati od prošloga za svako kriptiranje, neovisno o izmjeni nad podacima. Time, uz činjenicu da čistom tekstu može se pristupiti samo uz ispravni master password, su zadovoljeni zahtjevi povjerljivosti za lozinke i za adrese.

Validacija oznake, osim zahtjeva za povjerljivost, osigurava integritet adresa i lozinki. Ukoliko su podaci na disku izmjenjeni, validacija će sigurno biti bezuspješna i korisnik će biti obavješten neispravnom stanju sustava.

Korištenje

Program se koristi iz naredbenog retka te korisnik mora imati instaliran python3 i biblioteku pycriptodome. Korištenje se sastoji od 3 koraka: inicijalizacije (init), postavljanja lozinke za adresu (put) te dohvata lozinke za zadanu adresu (get).

1. Na samom početku korisnik treba odabrati svoj master password koji će mu služiti za pristup bazi. Pomoću njega izvršava inicijalizaciju password managera naredbom:

   $ ./passManager.sh init < master_password >

   Navedena naredba predviđena je za izvršavanje na samom početku korištenja te će njezino ponavljanje uzrokovati brisanje baze i ponovnu inicijalizaciju.

2. Za dodavanje lozinke za pojedinu adresu potrebno je izvesti naredbu:

   $ ./passManager.sh put < master_password > < adresa > < lozinka >

3. Za dohvati lozinke adrese potrebno je izvesti naredbu:

   $ ./passManager.sh get < master_password > < adresa >