SIGURNOST RAČUNALNIH SUSTAVA, 2021/2022

PRVA LABORATORIJSKA VJEŽBA: SIMETRIČNA KRIPTOGRAFIJA
15.03.2022.

UVOD

Korisničke zaporke su često najslabija točka mnogih sustava, a kada korisnici biraju jake različite zaporke za sve servise koje koriste, pojavljuje se potreba za alat koji će ih sigurno pohraniti umjesto da ih korisnik mora sve zapamtiti. U sklopu prve laboratorijske vježbe potrebno je dizajnirati i implementirati jednostavan i siguran prototip alata za pohranu zaporki (*password manager)* koristeći simetričnu kriptografiju.

FUNKCIONALNI ZAHTJEVI

Alat mora omogućavati korisniku sljedeće:

1. Inicijalizacija alata odnosno stvaranje prazne baze zaporki.
2. Pohrana para *adresa*, *zaporka*. Ako je već pohranjena zaporka pod istom adresom onda ju je potrebno zamijeniti sa zadanom.
3. Dohvaćanje pohranjene zaporke za zadanu adresu.

Alat je potrebno ostvariti da radi iz komandne linije, a podatke sprema na disk u nekom obliku koristeći pritom simetričnu kriptografiju kako bi osigurao povjerljivost i integritet spremljenih podataka. Podaci moraju biti zaštićeni putem *glavne zaporke* (*master password)* koju korisnik mora navesti prilikom svakog korištenja alata.

Interakcija s alatom može izgledati ovako:

```
$ ./tajnik init mAsterPasswrd
Password manager initialized.
$ ./tajnik put mAsterPasswrd www.fer.hr neprobojnAsifrA
Stored password for www.fer.hr.
$ ./tajnik get mAsterPasswrd www.fer.hr
Password for www.fer.hr is: neprobojnAsifrA.
$ ./tajnik get wrongPasswrd www.fer.hr
Master password incorrect or integrity check failed.
```

Radi jednostavnosti možemo pretpostaviti da će se adresa i zaporka sastojati od najviše 256 znakova te da će svi znakovi biti ispisivi neprazni ASCII znakovi (ASCII kodovi od 33 do 126 uključivo).

SIGURNOSNI ZAHTJEVI

Kada bi alat samo podatke zapisao na disk bez zaštite, napadač koji trajno ili privremeno dobije pristup disku može doći do svih zaporki. Stoga, pretpostavljamo sigurnosni model u kojem *napadač ima pristup disku te može po volji čitati i mijenjati podatke*. Dakle, sigurnost sustava mora na neki način počivati na glavnoj zaporki. Obratite pažnju da napadač tijekom vremena može prikupiti više verzija datoteka baze podataka alata na disku zaštićenih istom glavnom zaporkom. Također je moguće da korisnik pohrani u alat

parove adresa-zaporka po napadačevom izboru. Pod tim pretpostavkama potrebno je osigurati sljedeće sigurnosne zahtjeve:

1. *Povjerljivost zaporki*: napadač ne može odrediti nikakve informacije o zaporkama, čak niti njihovu duljinu, čak ni jesu li zaporke za dvije adrese jednake, čak ni je li nova zaporka jednaka staroj kada se promijeni.
2. *Povjerljivost adresa:* napadač ne može odrediti nikakve informacije o adresama, osim da zna koliko se različitih adresa nalazi u bazi.
3. *Integritet adresa i zaporki:* nije moguće da korisnik dobije od alata zaporku za određenu adresu, ako prethodno nije unio točno tu zaporku za točno tu adresu. Obratite pažnju na *napad zamijene*: napadač ne smije moći zamijeniti zaporku određene adrese zaporkom neke druge adrese.

ZADATCI

1. Samostalno istražite što su to funkcije za derivaciju ključa (*key derivation function*), koje sigurnosne zahtjeve moraju zadovoljavati, i kako se koriste kako bi od zaporke dobili jedan ili više kriptografskih ključeva.
2. Dizajnirajte i opišite alat za baratanje zaporkama koji zadovoljava gore opisane funkcionalne i sigurnosne zahtjeve. Dokumentirajte na koji se točno način podaci zaštićuju prilikom spremanja na disk i na koji se točno način provjerava zaštita prilikom čitanja s diska. Obratite pažnju i dokumentirajte postupke generiranja ključeva odnosno deriviranja ključeva iz zaporke.

IMPLEMENTACIJA

Laboratorijsku vježbu možete rješavati koristeći jedan od programskih jezika C++, C, Python, C# ili Java. Rješenje mora biti moguće pokrenuti koristeći standardne alate i prevoditelje na Ubuntu Linux 18.04 sustavu.

Za programski jezik Java, preporučamo da koristite standardne kriptografske biblioteke u sklopu Java SE 11 izdanja, a za programski jezik Python biblioteku pycryptodome. Niže su poveznice za pojedine kriptografske primitive koji bi vam mogli biti korisni prilikom izrade alata. Nije nužno koristiti sve od navedenog kako bi se ostvarilo ispravno rješenje.

Simetrična šifra:

- https://docs.oracle.com/en/java/javase/11/docs/api/java.base/javax/crypto/Cipher.html
- https://pycryptodome.readthedocs.io/en/latest/src/cipher/cipher.html

Kod za integritet poruke:

- https://docs.oracle.com/en/java/javase/11/docs/api/java.base/javax/crypto/Mac.html
- https://pycryptodome.readthedocs.io/en/latest/src/hash/hmac.html

Kriptografska funkcija sažetka:

- https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/security/MessageDigest.html
- https://pycryptodome.readthedocs.io/en/latest/src/hash/hash.html

Kriptografski generator slučajnih brojeva:

- https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/security/SecureRandom.html

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

Funkcije za derivaciju ključa:

- https://docs.oracle.com/en/java/javase/11/docs/api/java.base/javax/crypto/SecretKeyFactory.html
- https://docs.oracle.com/en/java/javase/11/docs/api/java.base/javax/crypto/spec/PBEKeySpec.html
- https://pycryptodome.readthedocs.io/en/latest/src/protocol/kdf.html

ZA ONE KOJI ŽELE VIŠE

1. Oblikujte alat tako da može baratati s jako puno zaporki. U tom slučaju nije moguće da se svi podaci zajedno zaštite, već je potrebno zaštititi svaki par zasebno i osmisliti mehanizam kojim se na temelju adrese može brzo dohvatiti odgovarajuća zaporka bez da se naruše sigurnosni zahtjevi.
2. Sigurnosni zahtjevi ne uključuju obranu protiv napada vraćanja stare verzije (*rollback attack)*. Osmislite i implementirajte mehanizam koji će onemogućiti ili otežati napadaču da pojedine zaporke ili cijelu bazu vrati na neku staru verziju.

PREDAJA

Potrebno je predati arhivu koja sadrži:

- Izvorni kod vašeg rješenja.
- Upute za prevođenje i pokretanje, idealno u obliku *shell* skripte koja će prilikom pokretanja prevesti vaše rješenje te ga nakon toga pokrenuti kako bi se demonstrirala sva funkcionalnost.
- Tekst datoteku koja sadrži opis vašeg sustava. Potrebno je u nekoliko rečenica opisati na koji način ste zaštitili zaporke i objasniti zašto su zadovoljeni sigurnosni zahtjevi.

Rok za predaju laboratorijske vježbe je **27.03.2022. u 23:59**.

Ako studenti iz bilo kojeg razloga ne stignu riješiti laboratorijsku vježbu do zadanog roka, još je uvijek mogu predati do **04.04.2022. u 23:59**. Ispravna rješenja poslana do tog roka ne donose bodove, ali omogućavaju studentima ispunjavanje minimuma i prolaz predmeta.

Nastavno osoblje će samostalno bodovati sva rješenja, a za svaku vježbu ćemo zatražiti od određenog broja studenata da u terminima laboratorijskih vježbi prezentiraju i objasne vlastito rješenje.

U slučaju problema ili nedoumica prilikom izrade vježbe molimo da pravovremeno kontaktirate nastavno osoblje putem mailing liste predmeta srs@fer.hr (isključivo koristeći fer.hr email adresu).

Važno: Dozvoljeno je i poželjno diskutiranje mogućih pristupa rješavanju vježbe između studenata. Međutim, samu laboratorijsku vježbu studenti moraju raditi samostalno. Nastavno osoblje će provesti provjere sličnosti predanih rješenja, a ponašanje koje nije u skladu s Kodeksom ponašanja studenata FER-a ćemo prijaviti Povjerenstvu za stegovnu odgovornost studenata te odrediti dodatne sankcije u sklopu predmeta. U slučaju problema ili nedoumica prilikom izrade vježbe molimo da pravovremeno kontaktirate nastavno osoblje.