

# SIGURNOST RAČUNALNIH SUSTAVA, 2021/2022

## DRUGA LABORATORIJSKA VJEŽBA: AUTENTIFIKACIJA UPOTREBOM LOZINKI 08.04.2022.

### UVOD

Lozinke (korisničke zaporkе) najčešći su i najjednostavniji način autentifikacije – kako za implementaciju, tako i za korištenje. Međutim, lozinke istovremeno imaju dosta ranjivosti koje prijetnje mogu iskoristiti. U sklopu ove laboratorijske vježbe potrebno je napraviti jednostavnu aplikaciju za prijavu korištenjem lozinki vodeći računa o prijetnjama i ranjivostima.

### FUNKCIONALNI ZAHTJEVI

U sklopu vježbe potrebno je implementirati dva alata, jedan koji omogućava upravljanje lozinkama i korisničkim imenima i namijenjen je administratorima te drugi koji služi za prijavu korisnika. Alat za upravljanje korisničkim imenima (neka se zove `usermgmt`) treba omogućiti:

1. Dodavanje novog korisničkog imena.
2. Promjenu lozinke postojećeg korisničkog imena.
3. Forsiranje promjene lozinke korisničkog imena.
4. Uklanjanje postojećeg korisničkog imena.

Drugi alat služi za prijavu (neka se zove `login`) na sustav te treba omogućavati:

1. Upisivanje korisničkog imena i lozinke pri čemu lozinka ne smije biti vidljiva tijekom upisa.
2. Promjenu lozinke nakon uspješne prijave – ako je tako postavio administrator.

Alati se koriste iz komandne linije, a podatke treba zapisivati u datoteku proizvoljnog formata. Alat koji služi za prijavu nakon uspješne prijave korisnika treba samo ispisati je li prijava bila uspješna ili ne.

Interakcija s alatima može izgledati ovako:

```
$ ./usermgmt add sgros
Password:
Repeat Password:
User add failed. Password mismatch.
```

```
$ ./usermgmt add sgros
Password:
Repeat Password:
User add successfully added.
```

```
$ ./usermgmt passwd sgros
Password:
Repeat Password:
Password change failed. Password mismatch.
```

```
$ ./usermgmt passwd sgros
```

```
Password:
Repeat Password:
Password change successful.

$ ./usermgmt forcepass sgros
User will be requested to change password on next login.

$ ./usermgmt del sgros
User successfully removed.

$ ./login sgros
Password:
Login successful.

$ ./login sgros
Password:
New password:
Repeat new password:
Login successful.

$ ./login sgros
Password:
Username or password incorrect.
Password:
Username or password incorrect.
Password:
Username or password incorrect.
```

Primijetite da alat ne daje informaciju što je točno problem prilikom prijave, nepostojeće korisničko ime ili kriva lozinka. Razmislite zašto je to tako.

Radi jednostavnosti možemo pretpostaviti da će se adresa i zaporka sastojati od najviše 256 znakova te da će svi znakovi biti ispisivi ASCII znakovi (ASCII kodovi od 33 do 126 uključivo), dakle nije potrebno podržavati UNICODE znakove.

## SIGURNOSNI ZAHTJEVI

Tajnost lozinki mora biti očuvana. Kako bi ispravno predvidjeli mehanizme zaštite pretpostavite sljedeći model prijetnje koji je identificiran tijekom faze dizajna sustava:

Naredba login pokreće se na takav način da se prijavljivati mogu samo legitimni korisnici. Svima njima se apsolutno vjeruje da neće zloupotrebljavati svoje ovlasti, a naredbu usermgmt ionako može pokretati isključivo administrator. Zbog načina kako će se pohranjivati datoteka s korisničkim imenima i lozinkama postoji način na koji bi ona mogla biti otuđena, tj. pasti u ruke napadača pri čemu napadači imaju vrlo napredne metode pogađanja lozinki i pristup značajnim količinama računalnih resursa.

Prijetnje i ranjivosti lozinki navedeni su u predavanju te su potom navedeni mehanizmi koji se koriste kako bi se uklonile ili ublažile ranjivosti.

## ZADATCI

1. Proučiti materijale s predavanja.
2. Dizajnirate i implementirate alate za upravljanje zaporkama te prijavu koji zadovoljavaju gore opisane funkcionalne zahtjeve te sadrže potrebne zaštitne mehanizme.
3. Ostvarite alat koristeći programski jezik po izboru.

Za one koji žele više:

1. Napišite dodatni alat koji će pokušati pogađati lozinke.
2. Pokušajte koristiti jedan od alata za pogađanje lozinki spomenutih na predavanju.

## IMPLEMENTACIJA

Laboratorijsku vježbu možete rješavati koristeći programski jezik po vlastitom izboru. Preporučamo programskih jezik više razine, npr. Java ili Python. Rješenje mora biti moguće pokrenuti koristeći standardne alate i prevoditelje na Linux operacijskom sustavu.

## PREDAJA

Potrebno je predati arhivu koja sadrži:

- Izvorni kod vašeg rješenja.
- Upute za prevođenje i pokretanje, idealno u obliku *shell* skripte koja će sve prilikom pokretanja prevesti vaše rješenje te ga nakon toga pokrenuti kako bi se demonstrirala sva funkcionalnost.
- Tekst datoteku koja sadrži opis vašeg sustava te za svaku zaštitu koja je opisana na predavanjima navesti jeste li ju implementirali ili ne te argumentirati odluku.

Rok za predaju laboratorijske vježbe je **19.04.2022. u 23:59.**

Ako studenti iz bilo kojeg razloga ne stignu riješiti laboratorijsku vježbu do zadanog roka, još je uvijek mogu predati do **08.05.2022. 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 nedoumice prilikom izrade vježbe molimo da pravovremeno kontaktirate nastavno osoblje putem mailing liste predmeta [srs@fer.hr](mailto:srs@fer.hr) (isključivo koristeći fer.hr email adresu) ili putem platforme MS Teams.

**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.