Infrastruktura javnih ključeva

Namena sistema

PKI (Public Key Infrastructure) je sistem za upravljanje digitalnim sertifikatima koji omogućava sigurnu autentifikaciju korisnika i uređaja, kao i zaštitu komunikacije. Podržava izdavanje, pregled i povlačenje sertifikata, upravljanje korisnicima, šablonima i automatsko obnavljanje sertifikata. Drugi deo sistema je deljeni menadžer lozinki sa enkripcijom. Sistem koristi HTTPS, višefaktorsku autentifikaciju i audit logove radi visokog nivoa bezbednosti.

Korisničke uloge

- **Administrator (Admin aplikacije)**
- **CA korisnik (npr korisnik iz kompanije koja poseduje CA sertifikat)**
- **Običan korisnik (korisnik koji će zahtevati end-entity sertifikat)**

Prava i funkcionalnosti po ulogama

Administrator

- Može da dodaje nove CA korisnike.
    - Prilikom registracije novih CA korisnika, sistem (backend) generiše nasumičnu lozinku koju šalje CA korisniku na mejl. Prilikom prvog pristupa sistemu, ovi korisnici moraju da promene podrazumevanu lozinku.
- Može da dodaje CA sertifikate za ove korisnike (za njihove sisteme, servere itd).
- Može da izdaje sve tipove sertifikata:
    - Root (samopotpisani)
    - Intermediate
    - End-Entity (EE)
- Ima uvid u sve sertifikate u sistemu.
- Može da koristi bilo koji CA sertifikat iz bilo kog lanca za dalje izdavanje.
- Može da vidi, preuzme ili povuče sve sertifikate.

CA korisnik

- Može da izdaje intermediate i end-entity sertifikate **samo za svoju organizaciju**, koristeći bilo koji sertifikat iz svog lanca.
- Može da vidi i preuzme samo sertifikate iz svog lanca.
- Može da kreira i koristi šablone vezane za svoje sertifikate.

Običan korisnik

- Može da:
    - uploaduje CSR i privatni ključ radi izdavanja sertifikata,
    - preuzme dobijeni sertifikat,
    - pregleda sertifikat,
    - povuče sertifikat uz navođenje razloga (po X.509 standardu).

Funkcionalnosti sistema

Registracija

Samo obični korisnici mogu da se registruju, na početnoj stranici putem forme za registraciju. Običan korisnik od podataka mora da unese *email* adresu, lozinku, ime, prezime i organizaciju. Lozinka mora da se unese dva puta, i mora da ispoštuje minimalne zahteve. Dodatno, potrebno je ugraditi estimator jačine lozinke. Korisnik mora da potvrdi svoj identitet nakon registracije, klikom na aktivacioni link iz mejla. Link je vremenski ograničen i može da se iskoristi samo jednom.

Prijava na sistem

Korisnici se prijavljuju na sistem unošenjem *email* adrese i lozinke. Dodatni korak prilikom prijave je rešavanje reCAPTCHE. CAPTCHA podrazumeva rešavanje prostih aritmetičkih operacija ili prepoznavanje oblika koje čoveku oduzimaju malo vremena a mašina trenutno ne ume dovoljno brzo da ih reši. Možete se osloniti na postojeća CAPTCHA rešenja poput Google reCAPTCHA ili Cloudflare.

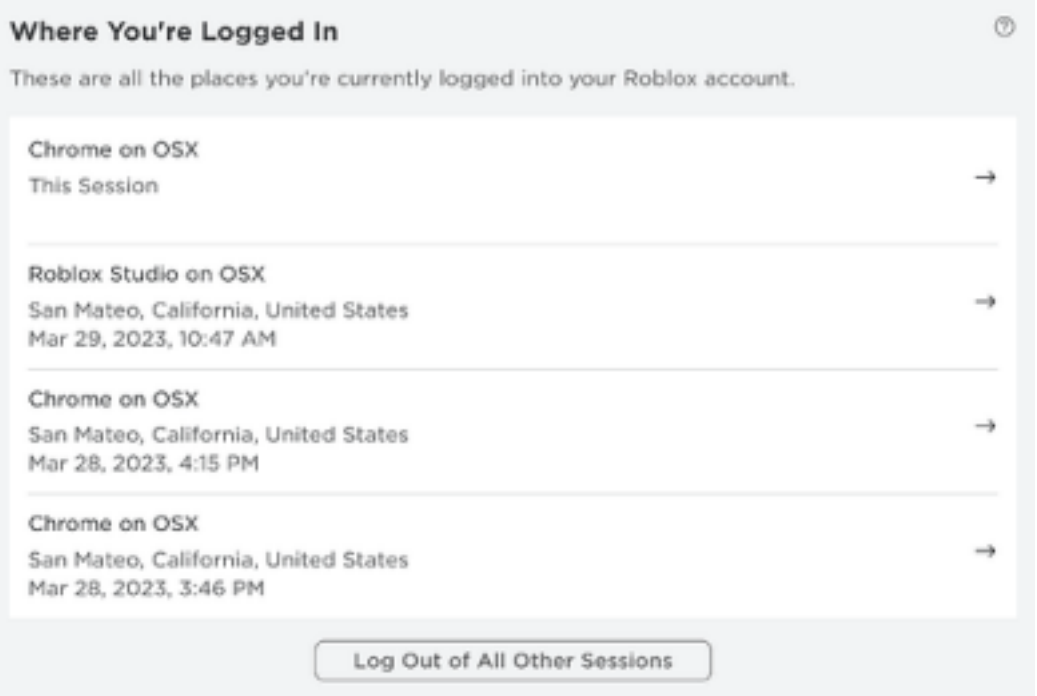
Oporavak naloga

Ukoliko korisnik zaboravi lozinku, može da povrati pristup nalogu tako što će pristupiti linku za oporavak naloga koji mu se šalje na *email*. Ovaj link ima ista svojstva kao i aktivacioni link - jednokratni je i vremenski ograničen.

Praćenje aktivnih tokena

Korisnik može iz korisničkog profila videti listu aktivnih JWT tokena – tj. na kojim uređajima i u kojim internet pregledačima je trenutno prijavljen. Za svaku sesiju se prikazuju osnovni podaci poput IP adrese, tipa uređaja i vremena poslednje aktivnosti. Korisnik može pojedinačno opozvati bilo koji token (npr. ako je zaboravio da se odjavi sa javnog računara), čime se korisnik trenutno odjavljuje sa tog uređaja. Ne čuvati tokene u bazi već ih identifikovati preko nekog polja.

Primer:



Izdavanje sertifikata

Sistem treba da podrži centralizovano izdavanje sertifikata za digitalne entitete svih nivoa. Potrebno je omogućiti izdavanje proizvoljnog broja *intermediate* sertifikata u jednom lancu. Za generisanje sertifikata korisnik unosi podatke o vlasniku (pogledati X500Name), definiše trajanje sertifikata i ekstenzije (npr. keyCertSign, BasicConstraints, …) i bira CA sertifikat koji će biti izdavalac. Prilikom generisanja sertifikata, potrebno je voditi računa o validnosti sertifikata izdavaoca. Validnost treba posmatrati u kontekstu perioda važenja, ispravnosti digitalnog potpisa i statusa povučenosti.

Čuvanje CA sertifikata

Za svaki izgenerisani CA sertifikat, potrebno je sačuvati njegov privatni ključ i kompletan lanac u *keystore*. Pošto je keystore zaštićen lozinkom, lozinku treba izgenerisati nasumično i čuvati je enkriptovanu u bazi ili keychain-u na nivou operativnog sistema. Ključ za enkripciju lozinke može da se izgeneriše pomoću PBKDF2 ili AES algoritma. Da biste olakšali generisanje i čuvanje ključeva za enkripciju, možete koristiti jedan simetrični ključ po CA korisniku i tim istim ključem možete enkriptovati sve lozinke za keystorove u kojima se čuvaju sertifikati kojima taj CA korisnik ima pristup.

Čuvanje End-entity sertifikata

Potrebno je čuvati EE sertifikate u PKI sistemu radi prikaza i opcije za preuzimanje. S obzirom da se za EE sertifikat ne sme čuvati odgovarajući privatni ključ, ovi sertifikati mogu da se čuvaju u keystore kao *TrustedCertificateEntry* ili u nekom drugom enkodovanom formatu (npr. der ili pem).

CSR (Certificate Signing Request)

CSR je zahtev koji pravi entitet kako bi zatražio sertifikat od sertifikacionog tela (CA). CSR

sadrži sve podatke o vlasniku sertifikata (pogledati X500Name), javni ključ i ekstenzije. End-entity korisnici prave CSR putem eksterne generacije - korisnik sam generiše ključeve i koristi alat za generisanje csr-a, poput openssl-a. Tako generisani csr se potom čuva u enkodovanom .pem formatu i *upload*-uje kroz formu na PKI sistemu. Pored *upload*-a je potrebno omogućiti korisniku da odabere CA za digitalni potpis. Na osnovu odabranog CA, korisnik može da unese trajanje sertifikata pri čemu se mora poštovati trajanje sertifikata odabranog CA.

Pregled i pristup sertifikatima

- Administrator vidi sve sertifikate.
- CA korisnik vidi samo sertifikate iz svog lanca.
- Korisnik vidi samo svoje EE sertifikate (može da ih ima proizvoljno mnogo).

Povlačenje sertifikata (Revocation)

Svaki korisnik ima mogućnost povlačenja sertifikata uz obavezno navođenje razloga iz X.509 standarda. Povučeni sertifikati ne bi smeli da se ponude dalje za izdavanje novih sertifikata (isto važi i za privatni ključ vezan za povučeni sertifikat). Za proveru povučenosti, potrebno je obezbediti Revocation servis (CRL proveru) pomoću CRL Distribution Point ekstenzije u sertifikatu ili podršku za OCSP.

Šabloni (Templates) za sertifikate

Šablon definiše ekstenzije i politiku sertifikata. Prilikom izdavanja sertifikata, potrebno je olakšati CA korisniku popunjavanje forme time što će mu se u zavisnosti od odabranog issuera, ponuditi (i popuniti) odgovarajuće ekstenzije sa predefinisanim vrednostima. Automatsko definisanje ekstenzija na osnovu odabranog sertifikata izdavaoca predstavlja šablon. Šablon se kreira tako što CA korisnik definiše sledeće vrednosti:

- **Naziv šablona**
- **CA issuer -** CA koji će dalje izdavati sertifikate na osnovu ovog šablona
- **Common Name (CN)** – regularni izraz za validaciju (npr. `.*\.ftn\.com`)
  - ovo podrazumeva da će se prilikom upotrebe šablona CN novog sertifikata validirati na osnovu unetog regularnog izraza
- **Subject Alternative Names (SANs)** – definiše regularni izraz za SAN ekstenziju (isto pravilo validacije važi kao i za CN)
- **TTL (vreme važenja)** – maksimalno trajanje
- **Key Usage** – podrazumevana vrednost ove ekstenzije
- **Extended Key Usage** – podrazumevana vrednost ove ekstenzije

CA korisnik može ali i ne mora da iskoristi unapred definisani šablon za izdavanje novog sertifikata. Takođe, korisnik može da izabere i druge ekstenzije koje nisu obuhvaćene šablonom, ali tako da ukupan skup odabranih ekstenzija ne prevazilazi politiku sertifikata koji se koristi za potpisivanje.

Funkcionalnost za deljeni password manager

Ova funkcionalnost omogućava korisnicima da bezbedno čuvaju i dele poverljive podatke kao što su lozinke i pristupni nalozi za sajtove i servise, koristeći javni i privatni ključ koji korisnik već poseduje u okviru PKI sistema.

Password manager je dostupan samo EE korisnicima koji posedu sopstveni par ključeva (privatni i javni), koji se generiše kroz postojeće funkcionalnosti sistema (CSR i izdavanje sertifikata). Kada korisnik želi da sačuva poverljivu informaciju (npr. lozinku za pristup sajtu), unosi sledeće podatke: naziv sajta, korisničko ime i lozinku. Lozinka se zatim šifruje pomoću korisnikovog javnog ključa i tako šifrovana čuva u bazi. Samo vlasnik koji poseduje odgovarajući privatni ključ može kasnije dešifrovati i pročitati tu lozinku. Privatan ključ se nikada ne čuva na serveru, već ga korisnik selektuje lokalno (na frontendu) kada želi da pristupi poverljivim podacima gde se i vrši dekripcija pomoću WEB Crypto API-ja. Primeri

Deljenje lozinki sa drugim korisnicima

Korisnik može izabrati da podeli neku od svojih lozinki sa drugim korisnikom. Lozinka se prvo dešifruje na frontendu, a zatim se ponovo enkriptuje javnim ključem korisnika sa kojim želi da je podeli (potrebno je omogućiti preuzimanje javnih ključeva EE korisnika). Rezultat tog šifrovanja se zatim šalje backendu i čuva uz originalnu lozinku. Time se omogućava da više korisnika vidi isti tajni podatak, ali tako da svaki korisnik može da dešifruje samo svoju verziju ključa. Ispod je dat primer strukture lozinke sačuvane na serveru (nije neophodno da baza bude identična). Obavezno je čuvanje metapodataka kao što su created_at, created_by i slični.

```
{
 "entry": {
 "id": 1,
 "site_name": "Gmail",
 "username": "alice2000",
 "owner_id": 1 // Alice
 },
 "shares": [
 {
 "user_id": 1, // Alice
 "encrypted_password": "MIIBIjANBgkqhkiG9w0BAQEFA...=="  },
 {
 "user_id": 2, // Bob
 "encrypted_password": "MIIBIjANBgkqhkiG9w0BAQEFG...=="  }
 ]
}
```

Primer loznike u bazi

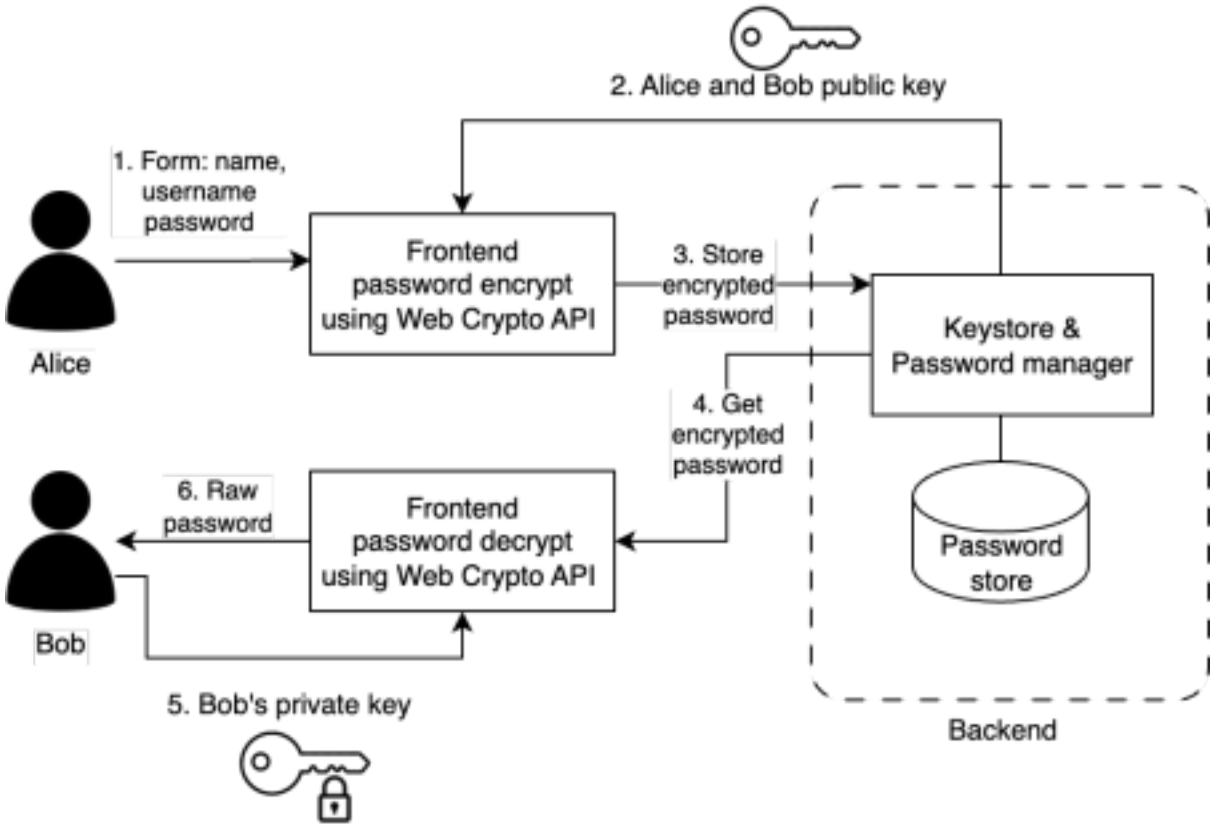


Diagram čuvanja i čitanja lozinke **Opšti zahtevi**

Potrebno je obezbediti komunikaciju klijentskog i serverskog dela aplikacije putem HTTPS protokola. To podrazumeva da izgenerišete nov sertifikat, i iskonfigurišete aplikaciju tako da iskoristi taj sertifikat za HTTPS komunikaciju između svih servisa aplikacije. Ukoliko tim izgeneriše novi sertifikat pomoću svog PKI sistema, dobiće dodatne bodove za ovu funkcionalnost.

Neophodno je implementirati autentifikaciju i kontrolu pristupa svih korisnika. Korisnici koji nisu autentifikovani nemaju prava pristupa ni jednoj stranici, osim stranici za registraciju i prijavu na sistem. Takođe nemaju prava pristupa nikakvim podacima sistema. Potrebno je obezbediti zaštitu pristupa za svaki ulaz u sistem (engl. endpoint) i na klijentskoj i na serverskoj strani aplikacije.

Potrebno je implementirati logging mehanizam koji ispunjava kompletnost, pouzdanost, upotrebljivost i konciznost. Log zapis treba da sadrži dovoljno informacija da dokaže neporecivost. Svaki događaj od bezbednosnog značaja za sistem treba da bude zabeležen. Format loga treba da prati standard i da bude u skladu sa najboljom praksom. Takođe, treba voditi računa o memorijskom zauzeću logova i napraviti mehanizam za rotaciju logova.

Zaštita od poznatih napada

Neophodno je da sistem bude otporan na poznate bezbednosne ranjivosti kao što su SQL injection i XSS (Cross-Site Scripting).

- Svi unosi korisnika moraju biti validirani i/ili sanitizovani pre korišćenja ● Zabranjeno je

direktno interpoliranje korisničkog inputa u SQL upite (koristiti prepared statements ako framework to ne handluje sam)
- Prikaz korisničkih podataka na frontend-u mora se vršiti tako da se spreči izvršavanje HTML/JS (XSS napadi) koda (escape-ovanje, DOM sanitizacija)
- Obratiti pažnju na verzije bibloteka koje se koriste i proveriti da li postoje neke poznate ranjivosti

Funkcionalnosti za stare studente koji rade samostalno

● generisanje sertifikata svih nivoa (obavezno voditi računa o trajanju sertifikata, digitalnom potpisu i ekstenzijama), bez omogućavanja lanca proizvoljne dubine ● cuvanje u keystore bez vodjenja racuna o lozinkama za keystore
● autentifikacija i autorizacija za administratora
● password manager bez deljenja poverljivih informacija (administrator generiše par ključeva i te ključeve koristi za enkripciju / dekripciju poverljivih informacija) ● logging
● HTTPS

Dodatne funkcionalnosti

autentifikacija pomoću keycloaka
2FA upotrebom neke autentikator aplikacije na mobilnom
analiza ranjivosti pomoću sonar/codacy/snyk šta god da ima besplatno i u skladu sa skeniranjem da se srede ranjivosti