Elektrotehnički fakultet Banja Luka
Katedra za računarstvo i informatiku
Kriptografija i kompjuterska zaštita

Uputstvo za korištenje OpenSSL alata – rad sa digitalnim sertifikatima

Za rad sa digitalnim sertifikatima, OpenSSL alat koristi konfiguracionu datoteku koja se referencira odgovarajućom opcijom prilikom poziva svake komande. Iako je moguće navoditi sve opcije iz komandne linije, postojanje centralne konfiguracione datoteke pomaže u uspostavljanju organizacije PKI infrastrukture i lakšem održavanju takvog sistema.

Parametri unutar konfiguracione datoteke koji se odnose na zahtjeve (CSR)

U okviru sekcije `req`, definisane su opcije koje preciziraju način popunjavanja svakog novog zahtjeva. Opcija `x509_extensions` sadrži naziv sekcije u okviru konfiguracione datoteke u kojoj su navedene ekstenzije samopotpisanog sertifikata generisanog iz ovog zahtjeva.
Za dodavanje posebnih ekstenzija u zahtjev za potpisivanje potrebno je uključiti opciju `req_extensions`, nakon čega se u sekciji koja odgovara vrijednosti tog parametra nalaze ekstenzije koje se dodaju u zahtjev (naziv sekcije je podrazumijevano `v3_req`). Jedna od najznačajnijih ekstenzija je `keyUsage`, koja specifikuje dozvoljene upotrebe ključa koji se nalazi u sertifikatu. Moguće vrijednosti su: `digitalSignature`, `nonRepudiation`, `keyEncipherment`, `dataEncipherment`, `keyAgreement`, `keyCertSign`, `cRLSign`, `encipherOnly` i `decipherOnly`.
Sekcija `req_distinguished_name` sadrži opcije koje se koriste prilikom popunjavanja zahtjeva. Svaki blok unutar ove sekcije sadrži naziv polja kao i maksimalnu i minimalnu dužinu. Sve informacije u ovoj sekciji su vezane za vlasnika zahtjeva, a RFC3739 propisuje sljedeća polja:

```
domainComponent
countryName
commonName
surname
givenName
pseudonym
serialNumber
title
organizationName
organizationalUnitName
stateOrProvinceName
localityName
```

Ako je potrebno upisati više vrijednosti za isto polje, notacija koja ovo omogućava je redniBroj.nazivPolja (npr. `0.countryName` i `1.countryName`).
Sadržaj ovih ekstenzija ne obavezuje korisnika sertifikata (aplikaciju) da implementira kontrolu ispravne upotrebe.

Parametri unutar konfiguracione datoteke koji se odnose na sertifikate i potpisivanje

Da bi se OpenSSL iskoristio za rad sa digitalnim sertifikatima, potrebno je kreirati sljedeće:

- direktorijum za skladištenje novih sertifikata – u ovom primjeru `certs`,
- direktorijum za skladištenje kopija novih sertifikata – u ovom primjeru `newcerts`,
- direktorijum za čuvanje privatnog ključa CA – `private`,
- direktorijum za čuvanje liste povučenih sertifikata – `crl`,
- moguće je, po potrebi, kreirati i dodatne direktorijume za bolju organizaciju okruženja (npr. direktorijum za čuvanje zahtjeva itd.),
- datoteku sa spiskom generisanih sertifikata – `index.txt`,
- datoteku sa rednim brojem sljedećeg sertifikata – u ovom primjeru `serial`. Ovu datoteku je moguće kreirati ručno upisivanjem prvog rednog broja (npr. 01),
- datoteku sa rednim brojem crl liste – u ovom primjeru `crlnumber`.

Kreirane datoteke i fajlove je potrebno referencirati u konfiguracionom fajlu (najčešće openssl.cnf).

```
[ ca ]
default_ca = rootCA                                    # naziv sekcije sa opisom CA
####################################################################
[ rootCA ]                                             # sekcija u kojoj je opisan CA
dir               = ./rootCA                           # osnovni folder
certs             = $dir/certs                         # lokacija sertifikata
crl_dir           = $dir/crl                           # lokacija crl liste
database          = $dir/index.txt                     # spisak sertifikata
new_certs_dir     = $dir/newcerts                      # kopije novih sertifikata

certificate       = $dir/cacert.pem                    # CA sertifikat
serial            = $dir/serial                        # trenutni serijski broj
crlnumber         = $dir/crlnumber                     # trenutni broj crl liste
crl               = $dir/crl.pem                       # trenutna crl lista

private_key       = $dir/private/cacert.key            # privatni ključ
RANDFILE          = $dir/private/.rand                 # slučajni broj

default_days      = 365                                # trajanje sertifikata
```

Polje `default_ca` pokazuje naziv sekcije u kojoj su specifikovani parametri za rad CA. Osim podataka o lokacijama datoteka, u ovoj sekciji su definisani i parametri za rad sa ekstenzijama u okviru sertifikata, kao i politikama koje svaki zahtjev koji se potpisuje mora ispuniti.
Polje `x509_extensions` sadrži naziv sekcije u kojoj su navedene ekstenzije koje će se nalaziti u potpisanom sertifikatu (podrazumijevana vrijednost ovog polja je `usr_cert`). Sekcija `usr_cert` sadrži podatke o ekstenzijama koje će biti smještene u svaki potpisani sertifikat koji ne predstavlja CA. U okviru ove sekcije se takođe može naći polje `keyUsage` (sa istim značenjem kao u zahtjevu). Vrijednost ovog polja se upisuje u sertifikat, bez obzira šta je u zahtjevu navedeno.
Parametar `policy` određuje naziv sekcije u kojoj su definisani obavezni podaci o vlasniku sertifikata (dijelovi DN vrijednosti) koji moraju biti dostupni u zahtjevu. Samo oni zahtjevi koji poštuju odabranu politiku mogu biti potpisani. Za svako imenovano polje unutar `policy` sekcije dostupne su sljedeće vrijednosti:

- `match` – polje u zahtjevu mora imati istu vrijednost kao i istoimeno polje u CA sertifikatu,
- `supplied` – polje mora biti uneseno, ali nema ograničenja po pitanju vrijednosti,
- `optional` – polje ne mora biti upisano.

Komande za rad sa zahtjevima i sertifikatima

Nakon podešavanja konfiguracije, potrebno je generisati zahtjev za potpisivanje CA sertifikata, a zatim i sam sertifikat. Sertifikat će, u ovom slučaju, biti samopotpisan (tj. potpisan od strane autoriteta kojem se i izdaje).

Za generisanje zahtjeva za sertifikatom koristi se komanda req. Parametri komande su sljedeći:

- `new` – signalizira aplikaciji da se radi o novom zahtjevu,
- `x509` – opcija koja se koristi ako je potrebno odmah i potpisati kreirani zahtjev. Upotrebljava se za samopotpisane sertifikate, pri čemu okruženje mora biti pripremljeno (datoteka *serial* mora sadržavati sljedeći redni broj),
- `key` *`lokacijaKljuca`* – lokacija para ključeva na osnovu kojeg se generiše sertifikat,
- `out` *`nazivFajla`* – lokacija generisanog zahtjeva (ili sertifikata, ako je iskorištena opcija za samopotpisivanje),
- `config` *`lokacijaKonfiguracijonFajla`* – lokacija konfiguracione datoteke (openssl.cnf),
- `days` *`brojDana`* – traženi rok važenja sertifikata.

```
openssl req –new –key private.key –out request.csr –config openssl.cnf –days 3650
```

Generisani zahtjev je u PEM formatu. Na sličan način kao i ranije moguće je izvršiti konverziju u DER format.

```
openssl req -in request.pem -out request.der -inform PEM -outform DER
```

Za postpisivanje sertifikata se koristi komanda `ca`. Sljedeće opcije specifikuju lokaciju i način potpisivanja:

- `in` *`nazivFajla`* – naziv fajla sa zahtjevom,
- `config` *`nazivFajla`* – naziv datoteke sa konfiguracijom,
- `name` *`naziv`* – sekcija konfiguracionog fajla koja sadrži podešavanja (ako postoji samo jedna sekcija, parametar se može izostaviti),
- `out` *`lokacija`* – lokacija na koju se postavlja sertifikat (poželjno je da to bude lokacija navedena u openssl.cnf fajlu,
- `selfsign` - pokazuje da sertifikat treba biti samopotpisan. Ovako potpisan sertifikat će se pojaviti u bazi sertifikata kao regularan unos. Mora se koristiti u kombinaciji sa opcijom `keyfile`,
- `keyfile` *`lokacija`* - pokazuje na lokaciju ključa koji se koristi za potpisivanje samopotpisanog sertifikata,

Generisani sertifikat je u PEM formatu. Konverzija i ispis informacija o sertifikatu su mogući komandom x509.

```
openssl x509 –in rootCA/cacert.pem –out rootCA/cacert.der –inform PEM –outform DER
openssl x509 –in rootCA/cacert.pem –noout –text
openssl x509 –in rootCA/cacert.der –inform DER –noout –text
```

Iz generisanog sertifikata je moguće prikazati (i izdvojiti) javni ključ na sljedeći način:

```
openssl x509 -in cert.pem -pubkey -noout > javnikljuc.pem
```

Povlačenje sertifikata je dodatna mogućnost `ca` komande. Opcije koje se koriste prilikom povlačenja sertifikata su:

- `revoke` *`lokacija`* – lokacija sertifikata koji se povlači,
- `crl_reason` *`razlog`* – razlog povlačenja, pri čemu su dostupne vrijednosti `unspecified`, `keyCompromise`, `CACompromise`, `affiliationChanged`, `superseded`, `cessationOfOperation`, `certificateHold`,
- `config` *`konfiguracioniFajl`* – lokacija konfiguracionog fajla,
- `gencrl` – signalizira da je potrebno generisati listu povučenih sertifikata,
- `out` *`lokacija`* – lokacija liste.

```
openssl ca –revoke newcerts/newca.pem –crl_reason superseded –config openssl.cnf
```

Ako je za povlačenje korišten `certificateHold`, onda je sertifikat suspendovan. Suspendovani sertifikati se reaktiviraju ponovnim povlačenjem sa razlogom `removeFromCRL`. Korištenje ovog razloga ima smisla u okruženjima koja podržavaju izdavanje delta CRL liste (OpenSSL ne podržava ovu funkcionalnost). U slučaju OpenSSL-a, za reaktivaciju je potrebno samo modifikovati datoteku sa bazom sertifikata (podrazumijevano index.txt) i ponovo generisati CRL listu.

Generisanje CRL liste u PEM formatu:

```
openssl ca –gencrl –out lista.pem
```

Konverzija CRL liste iz PEM u DER format:

```
openssl crl –in crl.pem –out crl.der –inform PEM –outform DER
```

Prikaz informacija o CRL listi u tekstualnom obliku:

```
openssl crl –in crl.pem –noout -text
```

Digitalni sertifikat i ključ se mogu konvertovati u PKCS#12 format. Za konverziju se koristi komanda `pkcs12`. Korištene opcije:

- `export` – pokazuje da je potrebno generisati PKCS#12 datoteku kao rezultat,
- `out` *`lokacijaPfxFajla`* – izlazna datoteka (najčešće sa pfx ili p12 ekstenzijom),
- `inkey` *`fajlSaKljucem`* – privatni ključ korisnika,
- `in` *`fajlSaSertifikatom`* – sertifikat korisnika,
- `certfile` *`fajlSaSertifikatom`* – CA sertifikat.

```
openssl pkcs12 -export -out cert.pfx -inkey userkey.pem -in usercert.pem -certfile
```

```
cacert.pem
```

Osnovne informacije o sadržaju PKCS#12 datoteke:

```
openssl pkcs12 -in cert.pfx -noout -info
```

Izdvajanje privatnog ključa iz PKCS#12 datoteke:

```
openssl pkcs12 -in cert.pfx -nocerts -out kljuc.kriptovan.priv.key
```

Uz prethodnu komandu je moguće koristiti oznaku konkretnog algoritma kojim će ključ biti kriptovan, kao u slučaju komande `genrsa` (`-des`, `-des3`, `-aes128`, itd.).

Izdvajanje klijentskog sertifikata iz PKCS#12 datoteke:

```
openssl pkcs12 -in cert.pfx -nokeys -clcerts -out klijent.pem
```

Izdvajanje CA sertifikata iz PKCS#12 datoteke:

```
openssl pkcs12 -in cert.pfx -nokeys -cacerts -out ca.pem
```