Uputstvo za generisanje zahteva, preuzimanje i postavljanje TCS sertifikata

Autori: Milica Kovinić (RCUB)

Apstrakt:

##Kratak opis rezultata

Sadržaj

1. Uvod

Da bi korisnici prilikom preuzimanja ili slanja osetljivih podataka na neki server imali zaštićenu komunikaciju, moraju da budu sigurni da su pristupili zaista onom serveru kojem su imali nameru da pristupe i da niko ne može da pročita/promeni podatke koji se šalju ili primaju. Upotreba digitalnih sertifikata u kombinaciji sa SSL tehnologijom omogućava pomenutu sigurnost.

AMRES je u saradnji sa TERENA uspostavio servis izdavanja serverskih sertifikata, gde TERENA ima ulogu sertifikacionog tela (CA – Certification Authority), a AMRES registracionog tela (RA – Registration Authority). Pravo na korišćenje ovih sertifikata za potrebe svojih servera i servisa imaju sve AMRES članice koje prethodno prođu kroz proces registracije.

Sertifikat TERENA SSL CA kojim se potpisuju svi serverski sertifikati za krajnje institucije izdati na ovaj način je potpisan od strane UserTrust, prelaznog sertifikacionog tela koje je potpisano od strane AddTrust External root sertifikacionog tela. Taj root sertifikat je preinstaliran u većini SSL klijenata (na primer, prilikom pristupanja sajtu preko https-a, koji se nalazi na web serveru koji ima TERENA sertifikat, nije potrebna intervencija korisnika kako bi sertifikat bio prihvaćen jer se odgovarajući Root CA sertifikat već nalazi preinstaliran u većini često korišćenih web browser-a: Internet Explorer, Mozilla Firefox, Google Chrome, Opera). Detaljnije o PKI, digitalnim sertifikatima i njihovoj primeni možete pročitati ovde.

Da bi registrovana institucija dobila sertifikat, potrebno je da uradi sledeće korake:

1. Registracija institucije
2. Kreiranje para ključeva i zahteva za sertifikatom
3. Podnošenje zahteva
4. Instalacija sertifikata i konfiguracija servera

1.1 Registracija institucije

AMRES članica dobija pravo korišćenja servisa registracijom. Registracijom, članica imenuje jednog do tri ovlašćena predstavnika za poslove dobijanja serverskih sertifikata. Registracija podrazumeva popunjavanje dokumenta u kome se imenuju ovlašćeni predstavnici koji će zastupati instituciju u postupcima zahtevanja, opozivanja, obnavljanja i dobijanja serverskih sertifikata. Imenuje se najmanje jedan, a najviše tri ovlašćena predstavnika. Takođe je potrebno da se potpiše „Saglasnost za korišćenje usluge izdavanja TERENA sertifikata – TCS“ između institucije i AMRES-a.

Saglasnost u kome su navedeni ovlašćeni predstavnici se šalju potpisani od strane ovlašćene osobe institucije i overeni pečatom institucije na adresu:

AMRES

Računarski Centar Univerziteta u Beogradu

Kumanovska bb

11000 Beograd

Posle provere podataka od strane AMRES-a, navedene ovlašćene osobe će mejlom biti obaveštene o rezultatu registracije.

Po uspešnoj registraciji, institucija, kao korisnik servisa, putem ovlašćenih osoba ostvaruje pravo na podnošenje zahteva i dobijanje serverskih sertifikata.

2. Kreiranje para ključeva i zahteva za sertifikatom

U ovom koraku se generišu privatni i javni ključ koji će se koristiti prilikom sigurnih transakcija i za formiranje serverskog sertifikata kao i zahtev za potpisivanje sertifikata (CSR – *Certificate Signing Request*) koji se šalje mejlom na adresu tcs@amres.ac.rs.

U nastavku je dato objašnjenje za kreiranje zahteva korišćenjem OpenSSL alata (za upotrebu sa Apache/mod_ssl serverom) na Linux-u, a uputstva za druge serverske platforme možete naći na adresi:

http://www.instantssl.com/ssl-certificate-support/csr_generation/ssl-certificate-index.html

Napomena: Zbog veće sigurnosti, AMRES je usvojio da minimalna dužina para asimetričnih ključeva, koji se generišu pri kreiranju zahteva za sertifikatom, mora biti 2048 bita.

2.1 Apache/mod_ssl

Sertifikat može da sadrži jednu ili više FQDN imena (maksimalno 100). Sertifikat sa više FQDN imena se koristi u slučaju da se na jednom serveru (jednoj IP adresi) nalazi više servisa sa različitim simboličkim adresama. Na primer, ako se više različitih web sajtova hostuje na jednom web serveru a različita FQDN imena sajtova se mapiraju u istu IP adresu servera, umesto da se na serveru nalaze posebni serverski sertifikati za svaki web sajt, moguće je da postoji samo jedan sertifikat koji bi važio za više FQDN imena.

U zavisnosti od toga da li sertifikat sadrži jedno ili više FQDN imena, mogu da se koriste dva predefinisana OpenSSL konfiguraciona fajla, *SCSreq.cnf* i *MultiSCSreq.cnf*. Samo jedan od njih se navodi u pozivu OpenSSL-a za kreiranje CSR zahteva. Ako se kreira zahtev za sertifikatom koji sadrži jedno FQDN ime servera, onda se poziva *SCSreq.cnf*, a u slučaju kreiranja zahteva sa više FQDN imena, poziva se *MultiSCSreq.cnf*.

Odgovarajući konfiguracioni fajl je potrebno prebaciti na server ili kopirati sdržaj konfiguracionog fajla u tekstualni fajl na serveru i sačuvati ga pod odgovarajućim imenom (SCSreq.cnf ili MultiSCSreq.cnf).

SCSreq.cnf – konfiguracioni fajl koji se poziva kada se kreira zahtev za sertifikatom koji sadrži jedno FQDN ime:

```
[ req ]
default_bits            = 2048
default_keyfile         = keyfile.pem
distinguished_name      = req_distinguished_name
#attributes              = req_attributes
#prompt                  = no
#output_password         = mypass
encrypt_key = no

[ req_distinguished_name ]
countryName                     = Oznaka zemlje (2 znaka)
countryName_default             = RS
countryName_min                 = 2
countryName_max                 = 2
localityName                    = Naziv lokacije (grad)
organizationName                = Pun naziv institucije
organizationName_default        = University of Belgrade
organizationalUnitName          = Naziv organizacione jedinice
organizationalUnitName_default = RCUB
commonName                      = FQDN adresa servera
commonName_max                  = 64
```

MultiSCSreq.cnf – konfiguracioni fajl koji se poziva kada se kreira zahtev za sertifikatom koji sadrži više od jednog FQDN imena:

```
[ req ]
default_bits            = 2048
default_keyfile         = keyfile.pem
distinguished_name      = req_distinguished_name
encrypt_key             = no
req_extensions          = v3_req

[ req_distinguished_name ]
countryName                     = Oznaka zemlje (2 znaa)
countryName_default             = RS
countryName_min                 = 2
countryName_max                 = 2

localityName                    = Naziv lokacije (grad)

organizationName                = Pun naziv institucije
organizationName_default        = University of Belgrade

organizationalUnitName          = Naziv organizacione jedinice
organizationalUnitName_default  = RCUB

0.commonName                    = FQDN adresa servera
0.commonName_max                = 64

[ v3_req ]
subjectAltName                  = @alt_names

[ alt_names ]
DNS.1                           =
DNS.2                           =
DNS.3                           =
```

U slučaju generisanja zahteva za sertifikatom koji sadrži više od jednog FQDN imena, u konfiguracioni fajl *MultiSCSreq.cnf* je potrebno dodatna imena upisati u delu [alt_names] kao vrednosti elemenata DNS.x. Dati konfiguracioni fajl predviđa tri dodatna FQDN imena, pri čemu taj broj, u zavisnosti od potrebe, može da bude manji ili veći. Ako je broj dodatnih FQDN imena manji od tri potrebno je obrisati suvišne DNS.x elemente (obrisati celu liniju koja počinje sa DNS.x).

Kada je kreiran konfiguracioni fajl prelazi se na generisanje zahteva za sertifikatom. Da bi ovaj zahtev mogao da bude kreiran, potrebno je da bude instaliran *OpenSSL* softverski paket.

Pre zadavanja *openssl* komande, *unmask* komandom se definišu *read* privilegije za sve što će biti kreirano posle zadate komande. Na ovaj način se štiti privatni ključ servera, koji je tajni i ne sme da bude javno dostupan.

Pri pozivanju *openssl*-a zadaje se putanja do jednog od gore pomenutih konfiguracionih fajlova (komanda se zadaje u bilo kom direktorijumu). U dole navedenoj komandi *myserver.key* i *server.csr* predstavljaju nazive fajlova u koje će biti smešteni privatni ključ servera i zahtev za sertifikatom, redom, pri čemu imena ovih fajlova mogu da se promene i zgodno je da nose ime samog servera.

```
umask 0377
openssl req -new -config SCSreq.cnf -utf8  -keyout myserver.key -out server.csr
```

Pokretanjem navedene *openssl* naredbe, tražiće se unos dodatnih informacija, pre samog generisanja zahteva (slika 1). Tražene podatke je potrebno tačno popuniti u skladu sa podacima koji su podneti u procesu registracije.

```
Generating a 2048 bit RSA private key
......+++
.....................+++
writing new private key to 'myserver.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Oznaka zemlje (2 znaka) [RS]:
Naziv lokacije (grad) []:Belgrade
Naziv institucije [University of Belgrade]:
Naziv organizacione jedinice [RCUB]:
FQDN ime servera []:imeservera@rcub.bg.ac.rs
```

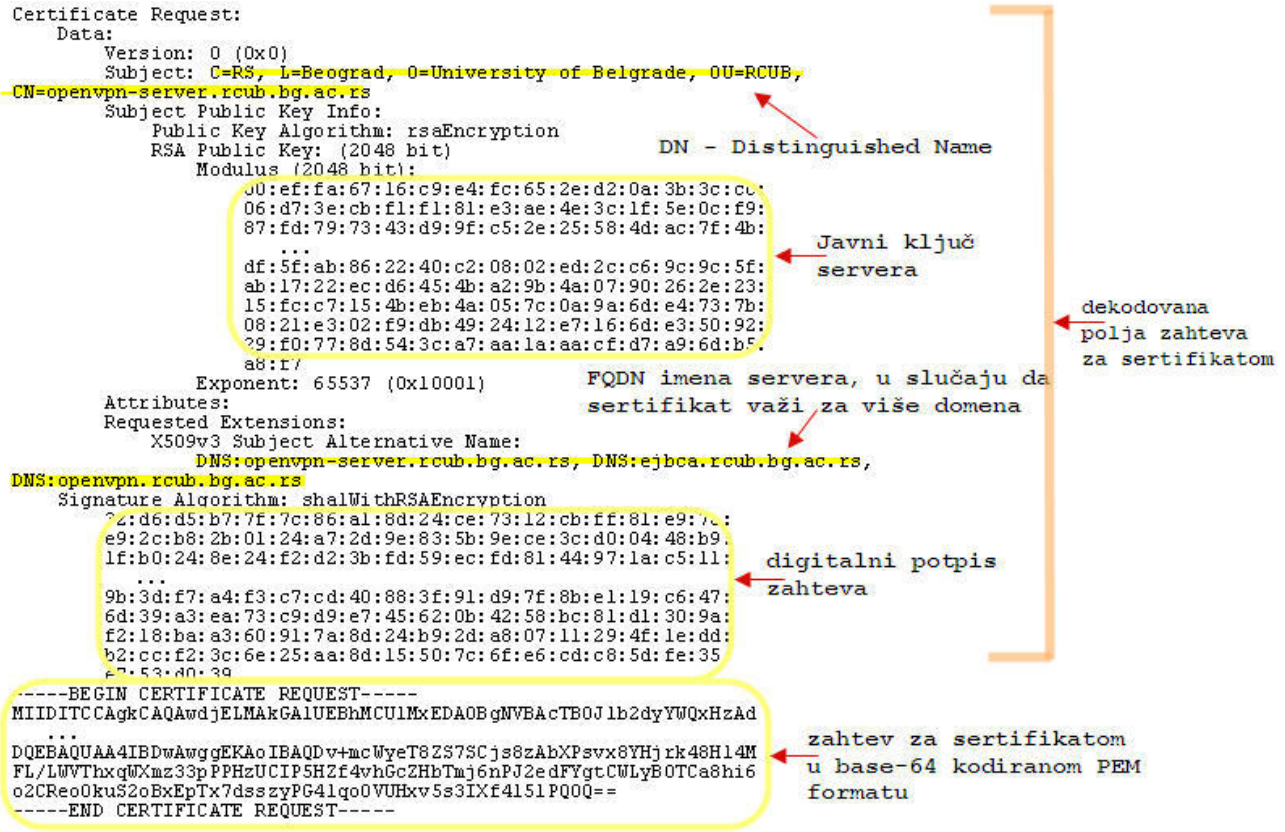
Slika 1

Na ovaj način je generisan par ključeva (privatni ključ se smešta u fajl *myserver.key* a javni ključ se pakuje u sam zahtev) i zahtev za potpisivanje sertifikata koji u sebi sadrži podatke o instituciji, jedno ili više FQDN imena i javni ključ servera.

Podaci koji treba da se nalaze u sertifikatu se digitalno potpisuju privatnim ključem servera i u obliku zahteva za sertifikatom se automatski smeštaju u fajl *server.csr*. Privatni ključ, koji se smešta u fajl *myserver.key*, ostaje na serveru i ne sme da bude javno dostupan. U nastavku teksta su navedeni preporučeni direktorijumi u kojima se čuvaju sertifikati i ključevi.

Kreirani zahtev za sertifikatom može da se proveri sledećom komandom:

```
openssl req -in myserver.csr -text
```



Slika 2

3. Podnošenje zahteva

Zahtev za sertifikatom podnosi ovlašćena osoba institucije određena u toku registracije institucije. Prilikom podnošenja zahteva potrebno je sadržaj fajla *myserver.csr* kopirati kao tekst u telo poruke mejla i poslati na adresu tcs@amres.bg.ac.rs. Fajl *.csr* može da se otvori bilo kojim tekstualnim editorom. Zahtev se nalazi u *base-64* kodiranom PEM formatu.

U nastavku je dat primer sadržaja fajla *myserver.csr*:

```
-----BEGIN CERTIFICATE REQUEST-----
MIIDITCCAgkCAQAwdjELMAkGA1UEBhMCUlMxEDAOBgNVBAcTB0Jlb2dyYWQxHzAd
BgNVBAoTFlVuaXZlcnNpdHkgb2YgQmVsZ3JhZGUxDTALBgNVBAsTBFJDVUIxJTAj
BgNVBAMTHG9wZW52cG4tc2VydmVyLnJjdWIuYmcuYWMucnMwggEiMA0GCSqGSIb3
DQEBAQUAA4IBDwAwggEKAoIBAQDv+mcWyeT8ZS7SCjs8zAbXPsvx8YHjrk48H14M
+Yf9eXND2Z/FLiVYTax/S59YuFKi1vlmkxEFusspaDCnPs8dQovX2UYHZt9tNGXS
fzk2x7rviI/mG1y3o15Y0QH96Ov6+R2aGBAPcimjLtWh17KaAE0Xon4V6QWExNU6
0TkP73/krflXTehJh2GdT7OvPCbJnwXUTN/RxLqETyL/BlbQr0mmi7Kqdy3xQLJM
ng5kBQ+fkd9fq4YiQMIIAu0sxpycX6sXIuzWRUuim0oHkCYuIxX8xxVL60oFfAqa
OiQyonjCG7ZJXngh7OtESeJEEtCniy+fzzweedv62kLTjMx5Osxw6FzUMuXCugzg
8kDH3DS607iv4Gn3IhYk/aV6H6hJtwUZaA/vssst6MvM6SJOeePZbpvGbYbnUAXU
FL/LWVThxqWXmz33pPPHzUCIP5HZf4vhGcZHbTmj6nPJ2edFYgtCWLyB0TCa8hi6
o2CReo0kuS2oBxEpTx7dsszyPG4lqo0VUHxv5s3IXf4151PQOQ==
-----END CERTIFICATE REQUEST-----
```

Takođe, u mejlu je potrebno navesti i sledeće podatke:

- pun naziv institucije pod kojim je registrovana na registru domena ac.rs (https://registar.ac.rs), ulica i broj, grad
- jedno ili više FQDN ime servera (ako se navodi više simboličnih imena u jednom sertifikatu, naglasiti koje ime je primarno)
- period važenja sertifikata – 1, 2 ili 3 godine
- serverski softver korišćen za generisanje zahteva
- podatke o ovlašćenoj osobi koji su navedeni prilikom registracije
- e-mail adresu na koju će biti poslat sertifikat

Posle uspešne provere zahteva od strane AMRES-a, sertifikat će biti izdat i poslat na mejl koji je naveden prilikom zahtevanja sertifikata.

4. Instalacija sertifikata

Sertifikat servera i *ca-chain* stižu na mejl poručioca u jednom zip fajlu. Dobijeni fajl ima naziv u obliku sedmocifrenog broja (na primer 9026687.zip) ili kao samo FQDN ime servera koje je definisano u sertifiktu.. Pri instalaciji dobijenog sertifikata potrebno je konfigurisati server da pored svog sertifikata šalje i *ca-chain* kojim se uspostavlja lanac poverenja. TERENA SSL CA sertifikat kojim se potpisuju serverski sertifikati je potpisan od strane *UserTrust* prelaznog sertifikacionog tela koje je potpisano od strane *AddTrust External root* sertifikacionog tela. Prelazni i *root* sertifikat su smešteni zajedno u *ca-chain* fajlu. Većina ssl klijenata ima preinstaliran root sertifikat, ali da bi sertifikat servera mogao da se proveri, klijentu je potreban i prelazni sertifikat koji je potpisan CA *root* sertifikatom.

4.1 Web server - Apache/mod_ssl

Korišćenjem https protokola omogućava se sigurna razmena podataka između web servera i klijenta. Pomoću sertifikata servera vrši se autentifkacija servera (klijent može da bude siguran da je pristupio traženom serveru), a zatim se sva dalja komunikacija odvija preko zaštićenog kanala korišćenjem enkripcije.

Dobijeni .zip fajl je prvo potrebno raspakovati:

```
unzip 9026687.zip
```

Pošto je fajl raspakovan dobijaju se dva fajla sa *.crt* i *.ca-bundle* ekstenzijama. Fajl sa *crt* ekstenzijom predtavlja sertifikat servera, dok se u *ca-bundle* fajlu nalazi prelazni i *root* sertifikat. Za Linux distribucije nastale od Redhat-a (CentOS, Fedora, Mandriva) je uobičajeno da se sertifikati i ključevi čuvaju na sledećim lokacijama, redom:

```
/etc/pki/tls/certs/
/etc/pki/tls/private/
```

U *ssl.conf* konfiguracionom falju potrebno je uneti odgovarajuće putanje do serverskog i prelaznog sertifikata kao i ključa servera:

```
SSLCertificateFile /etc/pki/tls/certs/9026687.crt                    #(putanja do
sertifikata servera)
SSLCertificateKeyFile  /etc/pki/tls/private/myserver.key             #(putanja  do
ključa servera)
SSLCertificateChainFile /etc/pki/tls/certs/9026687.ca-bundle         #(putanja do
prelaznog sertifikata)
```

Na kraju je potrebno restartovati web server:

```
/etc/init.d/httpd stop
/etc/init.d/httpd start
```

U slučaju da želite da vidite instaliran sertifikat, to možete da uradite pomoću sledeće komande:

```
openssl x509 -in 9026687.crt -text
```

4.2 RADIUS server

Standard 802.1x omogućava autentifikaciju klijenata pre nego što im se dozvoli pristup mrežnim resursima. Kao framework protokol se koristi EAP (Extensible Authentication Protocol) u kombinaciji sa TLS (Transport Layer Security), TTLS (Tunneled TLS) ili u okviru PEAP (Protected EAP) protokola. Ovo uputstvo razmatra instalaciju i upotrebu serverskih sertifikata na RADIUS serveru (koji je realizovan na FreeRadius platformi) kod EAP-TTLS protokola pri eduroam servisu.

Sertifikat instaliran na RADIUS serveru se u toku procesa autentifikacije šalje klijentu. Sam sertifikat sadrži javni ključ servera i digitalni potpis ovlašćene institucije kojoj se veruje (CA – Certificate Authority). Klijent verifikuje identitet servera proveravanjem digitalnog sertifikata i tada može da koristi serverov javni ključ za kriptovanje kredencijala koje će mu slati (videti dodatak A za detaljnije objašnjenje EAP-TTLS protokola).

Prvo je potrebno obezbediti serverski sertifikat na način opisan u prethodnom delu ovog uputstva (npr. dobijeni sertifikat se zove 8866644.zip). Taj fajl je potrebno prebaciti na sam server (npr. korišćenjem programa WinnSCP). Nakon toga se fajl prebacuje na lokaciju gde se drže sertifikati vezani za FreeRadius (folder /etc/raddb/certs). U isti folder je potrebno prebaciti i privatni serverski ključ koji je generisan u procesu dobijanja samog serverskog sertifikata (npr. zove se privatnikljuc.key):

```
cp 8866644.zip /etc/raddb/certs/8866644.zip
#(prebacivanje zip fajla)

cp privatnikljuc.key /etc/raddb/certs/privatnikljuc.key
#(prebacivanje privatnog ključa)
```

zatim se fajl raspakuje:

```
unzip 8866644.zip
```

Kada se fajl raspakuje (komanda unzip), dobijaju se dva fajla, jedan sa .crt drugi sa .ca-bundle ekstenzijom. Sertifikat sa .crt ekstenzijom se prebacuje u .pem format preko sledećih komandi:

```
openssl x509 -in 8866644.crt -out 8866644.der -outform DER
openssl x509 -in 8866644.der -inform DER -out 8866644.pem -outform PEM
```

Privatni ključ servera mora da ima „r--------„ (read only) dozvolu. U slučaju da nema potrebno je da se dozvola promeni sa:

```
chmod 400 /etc/raddb/certs/privatnikljuc.key
```

Sada je ove sertifikate potrebno ubaciti u konfiguraciju vezanu za tls (jer za funkcionisanje ttls autentifikacije je potrebno prvo konfigurisati tls) u okviru eap.conf fajla:

```
certdir = /etc/raddb/certs
cadir = /etc/raddb/certs
private_key_file = /etc/raddb/certs/privatniključ.key
certificate_file = /etc/raddb/certs/8866644.pem
CA_file = /etc/raddb/certs/8866644.ca-bundle
```

Sada je potrebno restartovati RADIUS proces da bi promene u konfiguraciji postale aktivne:

```
killall radiusd
radiusd
```

Da bi klijent uspešno proverio sertifikat servera, mora da instalira sam sertifikat na svoj računar.

4.3 Email server

Mejl server koristi sertifikat kako bi se autentifikovao klijentima i kako bi ostvario sigurnu vezu sa klijentima korišćenjem enkripcije putem SSL/TLS protokola.

Procedura postavljanja sertifikata je ista kao i kod *web* servera, samo što se konfiguriše drugi konfiguracioni fajl. U konfiguracionom fajlu je potrebno uneti odgovarajuće putanje do serverskog i prelaznog sertifikata kao i ključa servera:

U slučaju **imap** i **pop3** servera, koji se koriste za primanje mejlova, konfiguracioni fajl je ***dovecot.conf*** *(/etc/dovecot.conf):*

```
ssl_cert_file= /etc/pki/tls/certs/8866644.crt  #(putanja do sertifikata servera)

ssl_key_file = /etc/pki/tls/private/myserver.key
#(putanja do privatnog ključa servera)

ssl_ca_file = /etc/pki/tls/certs/8866644.ca-bundle
#(putanja do prelaznog sertifikata)
```

Za smtp server, koji se koristi za slanje mejlova, konfiguracioni fajl je main.cf (/etc/postfix/main.cf).:

```
smtpd_use_tls = yes

smtpd_tls_CAfile = /etc/pki/tls/certs/8866644.ca-bundle
#(putanja do prelaznog sertifikata)

smtpd_tls_CApath = /etc/pki/tls/certs
#(putanja do direktorijuma u kome su sertifikati)

smtpd_tls_cert_file = /etc/pki/tls/certs/8866644.crt
#(putanja do sertifikata servera)

smtpd_tls_key_file = /etc/pki/tls/private/myserver.key
#(putanja doprivatnog ključa servera)
```

4.4 Java Web server (Tomcat, JBoss...)

Ukoliko je potrebno generisati sertfikat za Java web server kao što je Tomcat, koristi se alat keytool.

Generisanje zahteva za sertifikat je opisano na

http://www.instantssl.com/ssl-certificate-support/csr_generation/ssl-certificate-java.html

i zamenjuje poglavlje 1.2 ovog dokumenta. Dobijeni CSR fajl se koristi kao što je opisano u poglavlju 1.3, s tim što se za serverski softver navodi "Java web server".

Za potrebe instalacije sertifikata takođe se koristi keytool kao što je opisano na adresi:

https://support.comodo.com/index.php?_m=knowledgebase&_a=viewarticle&kbarticleid=275&nav=0,1,88

Obratiti pažnju da se prilikom poslednjeg importa mora koristiti isti alias kao i prilikom generisanja zahteva za sertifikat, nakon čega keytool potvrđuje uparivanje sa informacijom:

"*Certificate reply was installed in keystore*".