# Certifikati za servise

Kriptografski certifikati, kao što već znate, omogućuju sigurnu autentikaciju, a time i komunikaciju između dvije točke na Internetu, ali i sigurnu identifikaciju suprotne strane. Ovo omogućuje asimetrična kriptografija, no teorijom se u ovom članku nećemo baviti. Iako postoje tvrtke koje prodaju certifikate i time nedvojbeno potvrđuje da je web site vaš, certifikati koje sami potpisujete (*self-signed*) omogućuju isto što i komercijalni. Jedina razlika su dosadni *pop-up* prozori u *browseru* koji će stalno pitati korisnika vjeruje li dobivenom certifikatu ili ne.

Već neko vrijeme, CARNet nudi "prave" certifikate, u vidu DigiCert certifikata za akademsku zajednicu. Nažalost, neke institucije članice CARNeta nemaju pravo na ove certifikate, zato ćemo obraditi kreiranje i *self-signed* certifikata za one koji nemaju pravo na DigiCert certifikate.



---

### 1. DigiCert certifikati

Na web stranicama http://certifikati.carnet.hr [1]možete pronaći detaljne upute kako doći do ovih certifikata. Kontakt e-mail je tcs-ra@carnet.hr [2], a možete se obratiti i telefonom na broj 01/6661-770.

Sve članice CARNET-a imaju pravo na DigiCert certifikat, a ne samo punopravne članice. No, prije svega morate registrirati svoju ustanovu [3].

Kako su na gore navedenim stranicama detaljno napisane sve upute, ovdje nećemo ponavljati iste informacije. Možete konzultirati točku 2.b) kako bi vidjeli kako se kreira zahtjev za certifikatom s više FQDN [4]-ova, ali inače možete odmah skočiti na točku 3.

Još ćemo samo napomenuti:
- nemojte kriptirati ključ! (stavite "encrypt_key = no" u *.cnf* datoteci)
- rabite datoteku MultiSCSreq.cnf [5], jer je vrlo izvjesno da ćete trebati više FQDN-ova u istom certifikatu

### 2. Samopotpisani (*self-signed*) certifikati

Samopotpisani certifikati omogućuju sigurnu komunikaciju, ali će korisnik morati potvrditi "da vjeruje vašem certifikatu" svaki put kad posjeti vaš site preko SSL-a. Kod drugih servisa ovaj problem nije tako izražen, odnosno krajni korisnici ne vide problem (HTTP, IMAP, POP3...). Popup prozor će se pojavljivati sve dok korisnik trajno ne prihvati certifikat izdan od strane vaše institucije.

Možemo napraviti:

a) certifikat za jedan DNS unos (FQDN - Fully Qualified Domain Name, puno ime poslužitelja [4])
b) certifikat za višestruke FQDN-ove

Preporučujemo da odmah izradite certifikate za sve poslužitelje koje imate ili koje ćete imati.

Kako možete znati koje ćete poslužitelje imati? Možete jednostavno staviti kakva generička imena (npr. geografske pojmove), ili ime poslužitelja podesiti prema njegovoj funkciji (mail, www, student i slično, a prijedloge smo dali u točki 2.b).

Zar ne postoje *wildcard* certifikati (*.domena.hr)? Postoje, ali njihova uporaba nije preporučljiva iz sigurnosnih razloga. Treba uzeti u obzir i da nijedna akademska institucija u HR neće imati više od nekoliko javno dostupnih servisa, stoga nema ni prevelike potrebe za *wildcardovima*.

### 2. a) certifikati za jedan FQDN

Certifikat možete kreirati za samo jedan FQDN na jednostavan način (navest ćemo tri primjera):

#### I) Najbrži način (postupak vrijedi samo za paket apache2-cn!)

```
# find /etc/apache2 -name server.\* | xargs rm
# dpkg-reconfigure apache2-cn
```

Preko apachejeve install skripte generiramo nove certifikate, koji se obično nalaze u /etc/apache2/ssl.* direktorijima (inače, svi se certifikati mogu prebaciti u /etc/ssl/certs direktorij). Ovakav postupak će kreirati potpuno novi certifikat, ali će uzeti u obzir samo osnovni FQDN. Dakle, ako vaš poslužitelj ima ime "server.domena.hr", i certifikat će glasiti na to ime, te primjerice https://www.domena.hr neće raditi (samo https://server.domena.hr).

Gornji primjer za apache2 je specifičan (jer interno rabi posebnu skriptu carnet-generate-ssl), pa ćemo prikazati druge, generičke, slučajeve. Iako ovaj postupak rabe i drugi paketi, navest ćemo primjer za **dovecot** (dok se za primjerice postfix rabi certifikat snakeoil.pem iz paketa ssl-cert):

```
# cd /etc/ssl/certs
# rm `openssl x509 -noout -hash < dovecot.pem`.0
# rm dovecot.pem ../private/dovecot.pem
# dpkg-reconfigure dovecot-common
```

#### II) Uporaba skripte iz paketa apache2-cn (postupak vrijedi samo ukoliko imate CARNetov paket apache2-cn!)

Kako automatsko generiranje certifikata uzima u obzir samo osnovno ime računala, maloprije spomenuta skripta **carnet-generate-ssl** isporučena uz CARNetov paket apache2-cn vam pomaže da zadate **bilo koje ime** koje će se naći u certifikatu (uz uvjet da je unešeno u DNS). Najčešće će to biti "www.domena.hr".

```
# /usr/share/apache2-cn/carnet-generate-ssl /etc/apache2 www.domena.hr \
```

```
        root@domena.hr "Vasa Institucija"
```

**III) Potpuno manualni način (ovaj postupak vrijedi za sve servise!)**

Sve postupke iz gornjih točaka možete napraviti i ručno, uz možda malo više tipkanja:

```
# umask 0377
# cd /etc/ssl/certs
# rm `openssl x509 -noout -hash < dovecot.pem`.0
# rm dovecot.pem
# openssl req -new -x509 -days 3650 -nodes -out dovecot.pem \
   -keyout dovecot.key
Generating a 1024 bit RSA private key
.............................++++++
.....++++++
writing new private key to 'dovecot.key'
-----
You are about to be asked to enter information that
...
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:HR
State or Province Name (full name) [Some-State]:.
Locality Name (eg, city) []:Grad
Organization Name (eg, company) [Company]:Neka Institucija
Organizational Unit Name (eg, section) []:dom
Common Name (eg, YOUR name) []:server.domena.hr
Email Address []:root@domena.hr
# ln -sf dovecot.pem `openssl x509 -noout -hash < dovecot.pem`.0
```

Na gore opisani način možete generirati (odnosno obnoviti) certifikat za bilo koji servis.

**2.b) certifikati za višestruke FQDN-ove**

Za self-signed certifikate možete uporabiti istu datoteku kao i za kreiranje DigiCert certifikata, MultiSCSreq.cnf [5]. U nju, u sekciji "[ req ]" dodajte redak "encrypt_key = no". Ona će tada izgledati poput ovog:

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

[ req_distinguished_name ]
countryName = Oznaka zemlje (dvoznakovni kod)
countryName_default = HR
countryName_min = 2
countryName_max = 2

localityName = Naziv lokacije (npr. grad)
```

```
organizationName = Naziv organizacije
organizationName_default = naziv CARNet clanice

organizationalUnitName = Naziv organizacijske jedinice (npr. odjel)
organizationalUnitName_default = Internet

commonName = FQDN adresa posluzitelja
commonName_max = 64

[ v3_req ]
subjectAltName                  = @alt_names

[ alt_names ]
DNS.1 = FQDN alias posluzitelja 1

DNS.2 = FQDN alias posluzitelja 2

DNS.3 = FQDN alias posluzitelja 3
```

Brojeve (DNS.1, DNS.2..., 3...) možete povećavati do broja poslužitelja koji vam treba, no nemojte pretjerivati. Predlažemo sljedeća imena (uz osnovno ime poslužitelja), no to naravno ovisi samo o vašim potrebama:

1. www
2. mail (pop, imap, smtp)
3. student
4. webmail
5. forum
6. sluzba_x (studentska služba ili slično) itd.

Više od 10-ak imena vam sasvim sigurno neće trebati, pogotovo ako rabite osnovno ime poslužitelja za više servisa (npr. "server.domena.hr" za mail, web i ostalo).

### 3. *Ugrađivanje* certifikata u mrežne servise

Preostaje samo certifikate uključiti u svaki servis koji mislite rabiti. Ovo ovisi o konkretnom načinu konfiguracije svakog servisa, a mi ćemo prikazati sve standardne servise koji se mogu susresti na CARNetovim poslužiteljima.

Certifikate je potrebno zaštititi prije instaliranja. To ćemo napraviti s naredbom (naravno, treba promijeniti ime datoteke po vašim potrebama):

```
# chmod 640 /etc/ssl/certs/dovecot.*
# chown root:root /etc/ssl/certs/dovecot.*
```

Drugim riječima, certifikat ne smije biti dostupan nikome osim rootu i korisniku pod čijim se UID-om servis vrti.

### Apache 2.x

Konfiguracijska datoteka nije nužno u **/etc/apache/conf.d/ssl.conf**, kako je to uobičajeno, nego može biti u konfiguraciji bilo kojeg virtualnog poslužitelja u direktoriju sites-available ili samostalno,

primjerice u datoteci **/etc/apache2/sites-available/ssl**. Ukoliko je konfiguracija u samostalnom VHOST-u, provjerite je li taj virtualni poslužitelj sa SSL-om uključen:

```
# a2ensite ssl
Site ssl installed; run /etc/init.d/apache2 reload to enable.
```

Konfiguracija:

```
SSLEngine On
SSLCertificateFile              /etc/ssl/certs/mojserver.cert
SSLCertificateKeyFile           /etc/ssl/private/mojserver.key
SSLCertificateChainFile         /etc/ssl/certs/mojserver.chain
```

Ključ bi trebao biti u svom direktoriju, nedostupan običnim okrisnicima i zaštićen ovim pravima pristupa (chmod 600):

```
-rw------- 1 root ssl-cert 1675 Dec  5  2012 /etc/ssl/private/mojserver.key
```

Treba još samo restartati Apache sa naredbama:

```
# /etc/init.d/apache2 stop
# /etc/init.d/apache2 start
```

Napomena: apache obično zahtijeva potpuno zaustavljanje i ponovno pokretanje kako bi počeo rabiti certifikat.

**Postfix**

Konfiguracijske datoteke su **/etc/postfix/main.cf** i **/etc/postfix/master.cf**. Varijable **smtp_*** se rabe za konfiguraciju dijela za slanje pošte od vašeg serverra, dok se varijable **smtpd_*** rabe za konfiguraciju dijela za zaprimanje pošte (samo jedno slovo razlike, ali ta razlika je bitna). U /etc/postfx/main.cf:

```
# stara varijabla, ali se jos moze rabiti:
#smtp_use_tls = yes
# od Postfixa 2.3, treba rabiti (zakomentirati varijablu smtp*_use_tls):
smtp_tls_security_level = may
smtp_tls_cert_file = /etc/ssl/certs/mojserver.cert
smtp_tls_key_file = /etc/ssl/private/mojserver.key
smtp_tls_CAfile = /etc/ssl/certs/mojserver.chain
# stara varijabla, ali se jos moze rabiti:
#smtpd_use_tls = yes
# od Postfixa 2.3, treba rabiti:
smtpd_tls_security_level = may
smtpd_tls_cert_file = /etc/ssl/certs/mojserver.cert
smtpd_tls_key_file = /etc/ssl/private/mojserver.key
smtpd_tls_CAfile = /etc/ssl/certs/mojserver.chain
```

Ovime ste omogućili STARTTLS na portu 25, što je dovoljno za sigurno slanje maila.

Ukoliko ipak želite omogućili **preporučeni** port 587 (submission) za **sigurno** i **autenticirano** slanje maila, u datoteci **/etc/postfix/master.cf** treba odkomentirati/dopisati sljedeće:

```
submission inet n       -       -       -       -       smtpd
#        -o smtpd_enforce_tls=yes
        -o smtpd_tls_security_level=encrypt
        -o smtpd_sasl_auth_enable=yes
        -o smtpd_client_restrictions=permit_sasl_authenticated,reject
```

*Opcija **smtpd_tls_security_level** (Postfix 2.3 i noviji) zamjenjuje opcije **smtpd_enforce_tls** i **smtpd_use_tls** (< Postfix 2.3). Stara opcija je navedena jer se učestalo pojavljuje u raznim uputama na Internetu, pa smo htjeli naglasiti da se u Squeezeu i dalje ne treba rabiti.*

Port za SMTPS (465) koji se nekada rabio bi trebalo zakomentirati, jer je prenamijenjen za nešto drugo (multicast). No, u praksi se i dalje rabi za SMTPS, pa je odluka na vama:

```
#smtps     inet  n       -       n       -       -       smtpd
#         -o smtpd_tls_wrappermode=yes
#         -o smtpd_sasl_auth_enable=yes
```

Na kraju, treba restartati postfix servis s naredbom:

```
# /etc/init.d/postfix reload
```

**Dovecot**

~~**Dovecot 1.0 / 1.2 (squeeze)**~~

~~Konfiguracijska datoteka je i dalje /etc/dovecot/dovecot.conf, ali je potrebno navesti i prijelazni certifikat.~~

```
ssl_disable = no
ssl_cert_file = /etc/ssl/certs/mojserver.cert
ssl_key_file = /etc/ssl/private/mojserver.key
ssl_ca_file = /etc/ssl/certs/mojserver.chain
```

~~Za **dovecot 1.2 (squeeze)**, parametar ssl_disable ne postoji, nego treba umjesto njega rabiti:~~

```
ssl = yes
```

~~I ovdje naravno treba restartati servis sa:~~

```
# /etc/init.d/dovecot reload
```

~~**Dovecot 2.1.7 (wheezy)**~~

~~U izdanju **wheezy** konfiguracijska datoteka je razlomljena na više manjih. Konfiguracija SSL-a se nalazi u datoteci **/etc/dovecot/conf.d/10-ssl.conf**, a ukoliko ste obavili nadogradnju s paketom **carnet-upgrade**, nalazi se i u datoteci **95-cn7-upgrade.conf**.~~

~~*Uvijek vrijedi definicija koja dolazi u "alfanumerički novijoj" datoteci, pa možete upisati retke i u*~~

~~*vlastitu datoteku, primjerice 99-local.conf. Ta će konfiguracija onda vrijediti bez obzira što piše u prethodnim datotekama (ovo vrijedi za bilo koji konfiguracijski parametar).*~~

~~Relevantni retci se ponešto razlikuju:~~

```
ssl_ca = </etc/ssl/certs/mojserver.chain
ssl_cert = </etc/ssl/certs/mojserver.cert
ssl_key = </etc/ssl/private/mojserver.key
```

~~**Napomena**: inačica dovecota u wheezyu samopotpisane certifikate stvara unutar direktorija /etc/dovecot, odnosno ključeve u /etc/dovecot/private. Vidjeti~~ http://www.dovecot.org/doc/README.Debian-wheezy ~~[6].~~

**Dovecot 2.2.13 (jessie)**

U izdanju **jessie** konfiguracijska datoteka je razlomljena na više manjih, kao i u prethodnoj inačici za **wheezy**. Konfiguracija SSL-a se nalazi u datoteci **/etc/dovecot/conf.d/10-ssl.conf**, a ukoliko ste obavili nadogradnju s paketom **carnet-upgrade**, nalazi se i u datoteci **95-cn7-upgrade.conf**.

*Uvijek vrijedi definicija koja dolazi u "alfanumerički novijoj" datoteci, pa možete upisati retke i u vlastitu datoteku, primjerice 99-local.conf. Ta će konfiguracija onda vrijediti bez obzira što piše u prethodnim datotekama (ovo vrijedi za bilo koji konfiguracijski parametar).*

Relevantni retci se ponešto razlikuju od inačice u jessie, ali samo po tome da se očekuje da je certifikat spojen s prijelaznim certifikatom:

```
ssl_cert = </etc/ssl/certs/mojserver-cert-chain.pem
ssl_key = </etc/ssl/private/mojserver.key
```

**Napomena**: Certifikat i prijelazni certifikat spajate na ovaj način:

```
$ cat certifikat.pem chain.pem >> mojserver-cert-chain.pem
```

Redoslijed certifikata je bitan, pa tako mora ići prvo vaš certifikat, pa onda prijelazni certifikat. Naziv nastale datoteke nije bitan, može biti .pem ili .crt ili .cert - svejedno je.

**Vsftpd**

Konfiguracijska datoteka je /etc/vsftpd.conf.

```
ssl_enable=YES
allow_anon_ssl=NO
force_local_data_ssl=NO
force_local_logins_ssl=NO
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
rsa_cert_file=/etc/ssl/certs/mojserver-key-cert.pem
```

~~*Specifično je da starije inačice Vsftpda (do Lennyja) zahtijevaju da* ***ključ i certifikat budu u jednoj datoteci****, što možete napraviti sa*~~

```
# cat mojserver.pem mojserver.key >> mojserver-key-cert.pem
```

~~*Napravili smo posebnu datoteku kako ne bi potencijalno ometali druge servise koji traže da se ključ i certifikat nalaze u posebnim datotekama.*~~

~~*Sad možete restartati servis sa:*~~

```
# /etc/init.d/vsftpd reload
```

**Novije inačice vsftpda** (distribucija Lenny i dalje) imaju dodatnu direktivu **rsa_private_key_file** gdje **morate** navesti ključ:

```
rsa_private_key_file=/etc/ssl/private/mojserver.key
```

Kako bi funkcionirao cijeli lanac povjerenja, spojite prijelazni certifikat s osnovnim:

```
# cat certifikat.pem chain.pem >> mojserver-cert-chain.pem
```

U direktivi rsa_cert_file sada treba biti navedena ova spojena datoteka:

```
rsa_cert_file=/etc/ssl/certs/mojserver-chain-cert.pem
```

Sad možete restartati servis sa:

```
# systemctl reload vsftpd
```

Proftpd

Konfiguracijska datoteka je /etc/proftpd.conf.

```
  <IfModule mod_tls.c>
    TLSEngine on
    TLSLog /var/log/proftpd/tls.log
    TLSProtocol SSLv23
    # Kod problema, stavite:
    # TLSProtocol TLSv1

    # Are clients required to use FTP over TLS when talking to this server?
    TLSRequired off

    # Server's certificate
    TLSRSACertificateFile /etc/ssl/certs/mojserver.cert
    TLSRSACertificateKeyFile /etc/ssl/private/mojserver.key
    # CA the server trusts - ako imate Comodo certifikat
    TLSCACertificateFile /etc/ssl/certs/mojserver.chain
    # Authenticate clients that want to use FTP over TLS?
    TLSVerifyClient off

    # Allow SSL/TLS renegotiations when the client requests them, but
    # do not force the renegotations.
    TLSRenegotiate required off
  </IfModule>
```

Servis nakon ovoga treba restartati sa:

```
# /etc/init.d/proftpd reload
```

4. Provjera ispravnosti instaliranih certifikata

```
# openssl s_client -connect mojserver.mojadomena.hr:443 -showcerts -CApath /dev/null
| less
```

Port 443 je port na kojem HTTPS servis radi, te ga morate promijeniti u odgovarajući za ostale servise (primjerice 993 za IMAPS, 443 za HTTPS, 995 za POP3S, itd).

Ispravan rad servisa označavaju DVA ulančana certifikata (pojavljuju se DVA zaglavlja -----BEGIN CERTIFICATE----- i -----END CERTIFICATE-----), poslužiteljski i prijelazni pomoću kojega SSL klijenti mogu uspostaviti lanac povjerenja.

5. Dodatne napomene

Datoteke certifikata i ključeva mogu imati različite nastavke (ekstenzije), no sve su one u PEM formatu (ostali mogu biti DER i PKCS12). Tako, u *.pem datoteci može biti ključ, certifikat, ili čak i certifikat i ključ zajedno. Status možemo provjeriti otvaranjem dateoteka:

```
# grep BEGIN mojserver.pem
-----BEGIN CERTIFICATE-----
# grep BEGIN mojserver.key
-----BEGIN RSA PRIVATE KEY-----
# grep BEGIN mojserver-key-cert.pem
-----BEGIN RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
```

Dakle, datoteke mogu imati različite nastavke, a sadržavati iste stvari. Ovo je bila samo brza provjera, detaljnije informacije možete naći uporabom naredbe openssl:

```
# openssl x509 -in /etc/ssl/certs/imapd.pem -noout -text | grep Subject:
        Subject: O=Dovecot mail server, OU=server.,
        CN=server.domena.hr/emailAddress=root@domena.hr
```

Dakle, radi se certifikatu za Dovecot za poslužitelj "server.domena.hr". Ako izostavite naredbu *grep*, dobit ćete punu informaciju o certifikatu.

Za certifikate s višestrukim FQDN-ima, pod retkom "X509v3 Subject Alternative Name" bi trebali pisati svi vaši poslužitelji koje ste odabrali.

Izgled generiranog zahtjeva za certifikatom (za provjeru prije nego ga pošaljete TCS timu) možete provjeriti s naredbom:

```
# openssl req -in mojserver.csr -text | less
```

Ovime smo prošli sve što CARNet sistemcu može zatrebati oko certifikata. Naravno, poželjno je dalje samostalno istražiti informacije.

---

Ovaj članak u potpunosti zamjenjuje članak http://sistemac.carnet.hr/node/48 [7]

OSVJEŽENO: 2019-06-13

- Logirajte [8] se za dodavanje komentara

čet, 2007-11-08 00:03 - Željko Borš**Kuharice:** Linux [9]
**Kategorije:** Servisi [10]
**Vote:** 4.5

Vaša ocjena: Nema Average: 4.5 (2 votes)

**story_tag:** certifikat [11]
certifikati [12]
certifikati za servise [13]
DigiCert [14]
DigiCertCA [15]

**Source URL:** https://sysportal.carnet.hr/node/315

**Links**
[1] http://certifikati.carnet.hr/
[2] mailto:tcs-ra@carnet.hr
[3] https://certifikati.carnet.hr/registracija_ustanove/
[4] http://en.wikipedia.org/wiki/FQDN
[5] https://certifikati.carnet.hr/static/docs/MultiTCSreq.cnf
[6] http://www.dovecot.org/doc/README.Debian-wheezy
[7] https://sysportal.carnet.hr/node/48
[8] https://sysportal.carnet.hr/sysportallogin
[9] https://sysportal.carnet.hr/taxonomy/term/17
[10] https://sysportal.carnet.hr/taxonomy/term/28
[11] https://sysportal.carnet.hr/taxonomy/term/114
[12] https://sysportal.carnet.hr/taxonomy/term/115
[13] https://sysportal.carnet.hr/taxonomy/term/116
[14] https://sysportal.carnet.hr/taxonomy/term/117
[15] https://sysportal.carnet.hr/taxonomy/term/118