Subject Alternative Name not present in certificate

Asked 5 years, 7 months ago Active 12 months ago Viewed 30k times



I have generated a CSR that includes the field subject alt names:

25

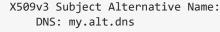
openssl req -out mycsr.pem -new -key mykey.pem -days 365



When I inspect this it looks as expected with a new field present:



4



However when I use this to sign a certificate that field is omitted for some reason.

I generate it with the following command:

```
openssl ca -out mycert.pem -infiles mycsr.pem
```

Can it be that my CA cert have to include the same Alt name for it to be included?

ssl openssl ssl-certificate

Share Improve this question Follow

asked Jun 22 '15 at 10:09

jimmy

1,683 • 2 • 15 • 27

Stack Overflow is a site for programming and development questions. This question appears to be off-topic because it is not about programming or development. See What topics can I ask about here in the Help Center. Perhaps Super User or Unix & Linux Stack Exchange would be a better place to ask. Also see Where do I post questions about Dev Ops?. – jww Jun 22 '15 at 16:00

Also see <u>How do you sign Certificate Signing Request with your Certification Authority?</u> – jww Jun 22 '15 at 16:03

@jww I can see why you say the question is off-topic but that seems to be the case for most SSL related questions on stack-overflow including the one you are linking:) – jimmy Jun 22 '15 at 16:10

Yeah, we (the community) do a poor job of keeping the site tidy at times. I do my best to tag all the new ones so folks citing them see they questions should be taken elsewhere. We **really** need that DevOps site for questions like this, questions about configuring Apache and Nginx, etc ... – jww Jun 22 '15 at 17:21

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up





You can use:



copy_extensions = copy



under your CA_default section in your openssl.cnf.



but only when you're sure that you can trust the extensions in the CSR as pointed out in this thread: http://openssl.6102.n7.nabble.com/subjectAltName-removed-from-CSR-when-signing-td26928.html

See also: How can I generate a self-signed certificate with SubjectAltName using OpenSSL?

Share Improve this answer Follow

edited May 23 '17 at 12:18



answered Jun 22 '15 at 10:21



@jimmy - be careful of $copy_extensions = copy$. You need to validate each signing request. A bad guy can set CA = TRUE and you will mint him a subordinate CA = jww Jun 22 '15 at 16:01

@jww Good advice. I will have to consider this. - jimmy Jun 22 '15 at 16:12

Then why isn't there a feature where you can whitelist which extensions to copy? :(– Steen Schütt Nov 29 '18 at 12:14



25

For everybody, who doesn't like to edit the system-wide <code>openssl.conf</code>, there's a native openssl CLI option for adding the SANs to the <code>.crt</code> from a <code>.csr</code>. All you have to use is openssl's <code>extfile</code> and <code>-extensions</code> CLI parameters.



Here's an example:



openssl x509 -req -days 3650 -in alice.csr -signkey aliceprivate.key -out alice.crt -extfile alice-csr.conf -extensions v3_req

This requires a alice-csr.conf file, which looks like this (fill in your appropriate data) and which was used to generate the <code>.csr</code> with the command <code>openssl req -new -key aliceprivate.key -out alice.csr -config alice-csr.conf:</code>

```
[req]
distinguished_name = req_distinguished_name
req_extensions = v3_req
prompt = no

[req_distinguished_name]
C = DE
ST = Thuringia
```

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up



```
[v3_req]
keyUsage = keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth
subjectAltName = @alt_names
[alt_names]
DNS.1 = server-alice
DNS.2 = localhost
```

Keep in mind, that the <code>-extensions v3_req</code> option corresponds to the <code>[v3_req]</code> section in the file <code>alice-csr.conf</code>, where you define you Subject Alternative Names aka the domains, which you want to issue your certificate to.

As I always appreciate fully comprehensible examples, where one could reproduce every step, I created an example project featuring Spring Boot microservices:

https://github.com/jonashackt/spring-boot-rest-clientcertificates-docker-compose

Share Improve this answer Follow

answered Dec 12 '17 at 19:03



- 4 i don't understand why this worked vs the other 20 things i tried but it did.. thanks. Flo Woo Mar 22 '19 at 5:56
- thank you very much... this should be the accepted answer b/c it doesnt require systemwide changes (thus could be scripted) Chad May 31 '19 at 0:30

This was the only thing I could get to work for me. Even changing the system .cnf file, as suggested in the accepted answer did not work. I made one small change to make this highly portable. I got rid of the [alt_names] section entirely and replaced subjectAltName = @alt_names with subjectAltName = \$ENV::SAN . Using this you can specify any subject alternative name by assigning the SAN environ variable. — Robert Kearns Dec 1 '19 at 13:16

This got me a little further along. You still have to use -config with the req command when building a CA it looks like. – John Ernest Aug 22 '20 at 23:34



Signing a CSR with alt names is described here well: https://www.feistyduck.com/library/openssl-cookbook/online/ch-openssl.html#creating-certificates-valid-for-multiple-hostnames



In short words, you create a something.ext file containing just the alt names:



```
subjectAltName = DNS:*.my.alt.dns, DNS:my.alt.dns
```

and then refer to this file in openss1 x509 -req ... command: -extfile something.ext . Note that it happens when signing the CSR, not when preparing it.

Share Improve this answer Follow

answered Jan 30 '20 at 9:15

Join Stack Overflow to learn, share knowledge, and build your career.



Join Stack Overflow to learn, share knowledge, and build your career.

Sign up