Rameez Kakodker Follow
Mar 5, 2019 · 5 min read · Listen

Save

[How-to] Simple way of generating Wildcard/SAN SSL CSRs for Product Managers

I recently came across a situation where I had to generate CSRs for a single, wildcard & SAN SSL certificates. And while I maintain that as Product Managers we have to prioritize our workload, sometimes, to speed things up, we need to get our hands dirty.

Before we jump into the details, we should know the differences amongst SSL types:

1. Single-name SSL Certificates

Protects a single subdomain/hostname.

Example: If you purchase single-name SSL Certificate for www.xyz.com, it doesn't mean you can secure mail.xyz.com.

2. Wildcard SSL Certificates

Protects an unlimited number of subdomains for a single domain.

Example: If you purchase a certificate for www.xyz.com, it will secure career.xyz.com, help.xyz.com, etc. It will work on any subdomain. However, it will not secure abc.pro.xyz.com.

3. Unified SSL Certificates/Multi-Domain SSL Certificates/SAN Certificates

It allows customers to protect up to 250 domains with the help of the same certificate. They are specially designed to secure Microsoft Exchange and Office Communications environments. It protects different domains with a single certificate with the help of the SAN extension.

Note: The instructions are for Mac. Running Mojave.

Prerequisite — Installing Openssl on your system

Using Homebrew for Mac

(Don't have Homebrew? Follow this).

This is the simplest way to do this. Simply open your terminal (iTerm?)and type this:

```
brew install openssl
```

If you're stuck on homebrew update, you can bypass the update by typing this:

```
HOMEBREW_NO_AUTO_UPDATE=1 brew install openssl
```

However, it is always advisable that you keep your formulas (formulae?) updated.

Generating a single domain CSR for SSL

This is for only a single domain, like www.websiteurl.com. So, all we need to do is open Terminal and enter the following command:

```
openssl req -new -newkey rsa:2048 -nodes -keyout private.key -out
generated.csr
```

After pressing enter, you'll be prompted with the following:

1. **Country Name (2 letter code)**
   Use your 2 char country code (USA is US, India is IN, UAE is AE etc.)
2. **State or Province Name (full name)**
   State in which your org is in... Dubai, Texas, Maharashtra etc.
3. **Locality Name (eg, city)**
   City name.
4. **Organization Name (eg, company)**
   Company name — usually this has to be the same as the domain. E.g. if you're making a CSR for Nike, the organization name should have Nike in it.
5. **Organizational Unit Name (eg, section)**
   Your team in the organization. Could be "IT dept", "Product Team" etc.
6. **Common Name (eg, fully qualified host name)**
   Domain name. In our case, websiteurl.com.
7. **Email Address**
   Your email address — try to use your official email id here.

8. **Password**

   Leave it blank.

After this, your screen should be like this:

```
RameezKs-MacBook-Pro:Downloads rameezk$ openssl req -new -newkey rsa:2048 -nodes -keyout private.key -out generated.csr
Generating a 2048 bit RSA private key
..............................+++
.....................................................................+++
writing new private key to 'private.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.
-----
Country Name (2 letter code) []:AE
State or Province Name (full name) []:Dubai
Locality Name (eg, city) []:Dubai
Organization Name (eg, company) []:Your Organization
Organizational Unit Name (eg, section) []:Your Team
Common Name (eg, fully qualified host name) []:www.websiteurl.com
Email Address []:youremail@email.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:password
RameezKs-MacBook-Pro:Downloads rameezk$
```

A simple **ls -l** will show you the folder contents. You'll find a 'generated.csr' file, which you'll need to upload to the SSL provider to generate the final certificate. The private.key will have to be uploaded eventually to the server where you'll install the certificate.

Generating a wildcard domain CSR for SSL

Follow the above steps, but when you specify the 'Common Name', enter *.websiteurl.com.

Note that your wildcard SSL **will not support** multiple sub-domains, i.e., the SSL certificate will verify **bar**.websiteurl.com but not **foo.bar**.websiteurl.com. That's the issue with wildcard SSLs — they say wildcard, but really it's only one level down.

Generating a SAN CSR for SSL

This requires a little bit of work. Follow each step, strictly.

Step 1 — Create a configuration file

To create a .conf file, first create a new folder. Open Terminal:

```
$ mkdir san
$ cd san
$ touch ssl.conf
$ open -a TextEdit ssl.conf
```

A file would have opened in TextEdit. Enter the values below:

```
[ req ]
default_bits = 4096
distinguished_name = req_distinguished_name
```



```
stateOrProvinceName_default = Dubai
localityName = Dubai
localityName_default = Dubai
organizationName = YourOrganizationName
organizationName_default = YourOrganizationName
commonName = websiteurl.com
commonName_max = 64
commonName_default = websiteurl.com

[ req_ext ]
subjectAltName = @alt_names

[alt_names]
DNS.1 = websiteurl.com
DNS.2 = www.websiteurl.com
DNS.3 = foo.websiteurl.com
DNS.4 =  bar.foo.websiteurl.com
DNS.5 =  websiteurl.net
DNS.6 =  foo.websiteurl.net
DNS.7 =  bar.websiteurl.net
```

Notes:

1. SAN certificates cover more than just your domain. You can add other domains, up to a max of 250. Use the DNS.# to add all possible domains & sub-domains.

2. The common name (CN) is the main domain you want to verify. Ensure that this domain is also under [alt_names] (DNS.#).

Save this file.

Step 2 — Generate private key

Go back to your terminal and enter the following (making sure you're in the right directory):

```
$ openssl genrsa -out private.key 4096
```

This generates the private.key fo

Step 3 — Generate CSR

Time to generate the CSR:

```
$ openssl req -new -sha256 -out private.csr -key private.key
-config ssl.conf
```

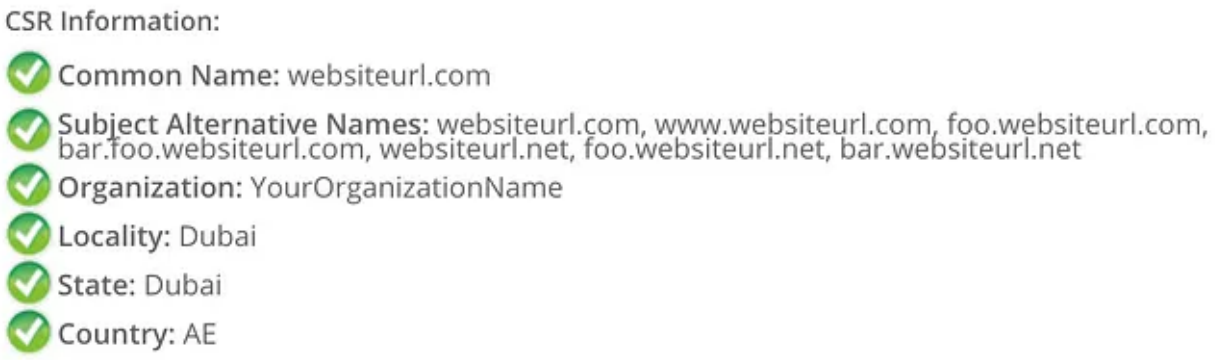
Again, you'll get the same options as earlier. Since your *ssl.conf* has the values already setup, keep pressing enter.

The CSR (private.csr) will now be generated.

```
RameezKs-MacBook-Pro:san rameezk$ openssl genrsa -out private.key 4096
Generating RSA private key, 4096 bit long modulus
.....................++
.....................................................................++
e is 65537 (0x10001)
RameezKs-MacBook-Pro:san rameezk$ openssl req -new -sha256     -out private.csr     -key private.key
  -config ssl.conf
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.
-----
AE [AE]:
Dubai [Dubai]:
Dubai [Dubai]:
YourOrganizationName [YourOrganizationName]:
websiteurl.com [websiteurl.com]:
RameezKs-MacBook-Pro:san rameezk$ ls -l
total 24
-rw-r--r--  1 rameezk  staff  1927 Mar  4 23:52 private.csr
-rw-r--r--  1 rameezk  staff  3247 Mar  4 23:52 private.key
-rw-r--r--@ 1 rameezk  staff   696 Mar  4 23:51 ssl.conf
RameezKs-MacBook-Pro:san rameezk$
```

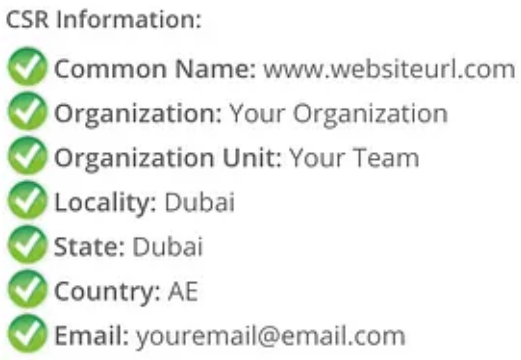
Verification of CSR

If you get a CSR from some other team, before you pass it on, do take some time to verify it. Use — https://www.sslshopper.com/csr-decoder.html. This is the information you'll see:

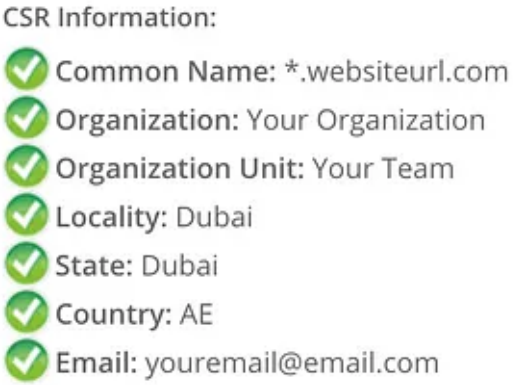


Essential to note the Common Name & the Subject Alternative Name (for SAN) — making sure that the SAN has the Common Name in it.

For a single SSL:



For wildcard:



And that's that!

Thank you for reading. Do note that this isn't the technical version of the process — you might have errors due to some specific SSL generation request that might require some specific requirements. Consult your nearest developer!

Ssl Product Management Command Line Wildcard Technology

You likey?

Me likey if you subscribe to receive articles from me in your inbox! I write only on the weekends, so you'll get, at max, 2 articles over the weekend. Think about it.

Emails will be sent to baskar.deepak3@gmail.com. Not you?

Subscribe