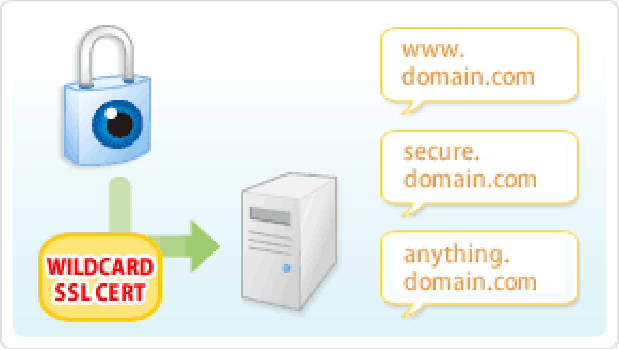
Howto create a (wildcard / multiple) SSL certificate

Author : admin



It's the first time I'm creating a wildcard ssl certificate. It appeared there is no fundamental difference between generating a normal SSL certificate and generating a wildcard certificate.

The procedure for generating a wildcard SSL certificate is as follows:

1. Generate an SSL key file

```
server:~# /usr/bin/openssl genrsa -des3 -out domain.com.key 2048
Enter pass phrase for domain.com.key:
```

Fill in any passphrase you like, the 2048 specifies the encryption level, 2048 is good enough and is the most commonly used as of today.
I've saw there is also an option to use 4096 bits encryption but I never tried that myself, I would be glad if somebody can share if he has succesfully established an SSL certificate with 4096 encryption.

2. Generate the certificate request file

```
server:~# /usr/bin/openssl req -new -key /home/hipo/domain.com.key -out /home/hipo/domain.com.csr
```

Further on it's necessery to fill in some info concerning the newly generated webserver SSL, e.g.:

```
Enter pass phrase for /home/hipo/domain.com.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) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:
Email Address []:
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Fill all the values according to your requirements, the only vital thing here is to fill in a proper **Common Name (eg, YOUR name) []:**

The **Common Name** should always be equal to ***.domain.com** , if something else is typed in the SSL certificate won't be considered a valid one when placed on the multiple subdomains.

The newly generated **domain.com.csr** file should be looking something similar to:

```
server:~# less -----BEGIN CERTIFICATE REQUEST-----
MIIC2jCCAcICAQAwgZQxCzAJBgNVBAYTAkdCMQ8wDQYDVQQIEwZMb25kb24xDzAN
BgNVBAcTBkxvbmRvbjEQMA4GA1UEChMHU2FudHJleDEWMBQGA1UECwwNKi5zYW50
cmV4Lm5ldDEWMBQGA1UEAwwNKi5zYW50cmV4Lm5ldDEhMB8GCSqGSIb3DQEJARYS
bWFzcmF3eXpAeWFob28uY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKC
AQEAlm9NwcQiA+AAVuVIjg8nCCn5nN14C3rSdcRNuE6oFo9E5uhl9yz8YDIg7wNx
FNQsmw0AwMzkao5Qv9yrmHqbht8qqPfG7YpcTiAoAuSaN6Nm25/zYrSu1uRsnc4C
9lINS8Va+n0Jt+CCQmomTKSarJqNfgo3j1ZU/HuOKcCEktIe0eKigMWxFKCM8wLh
CIdj6AwburckW1/ubOGlu2XKdcY5CbFe4cNGyME3rg33ft8b6v/ORWLSBMrt3QGP
bj42uZP6NoLaZCpsquJLeziLkT4rxdArUApdaTaEFrNMnwzGmUK10qmfx8SQodUl
QXmyd+PpQtaglymjIKN0L8Y36QIDAQABoAAwDQYJKoZIhvcNAQEFBQADggEBAIKe
UTXUt7XvaqVOesTmGCuVmv+Lz/GtGOEw+lfCNM4UFB950H975hHKo63YQr9Vqqqn
WlqZ0nXuwbZdfIh3xhTxzUqF/4m00OFQTbM9hwt6dyqLkmcc4J0rnTsvqjPkUsW5
U7iAIB/UIyDYUcAEky3gnokq3MLH42zXBViPM2+g/fkmJA4jaeoHGINbYMuxFh6Z
r2fIgAfGjms+hNaJvINDoBN5y6YUQbeJc+RoXMrG9clrDsUIGfmkTKCkG0BRJ2ki
Sdbm4IZMtQKU/C4a8vxZkFdleGqWeWL1SBtjjAnTtpb0uF+QmOPLcCoKnBwtEUU9
dPJlEzI+TCgAmKhZoXo=
-----END CERTIFICATE REQUEST-----
```

Next on this **BEGIN CERTIFICATE REQUEST** will have to be filled in to the certificate issuer website, whether it's requested, let's say in **GlobeSSL**.

Based on the certificate request a valid SSL certificate will be issued by the SSL provider.
Here one important note to make is that if your domain contians some prohibited keywords like, let's say

bank, finance, poker etc., a keywords which might be considered a fraud or forgery then probably the SSL certificate won't be issued by the SSL issuer and you will have to further contact the SSL cert provider and send them some more information related to the type of business the new website is going to run.
This kind of domain keyword filter, that is implemented by SSL certificate issuer companies is made to protect internet users from possible frauds or scam websites as well as reduce the level of potential dangerous pyramid like businesses that are so modern on the net these days.

Last step before the certificate will be visible in a browser is to set it for a domain name or virtualhost in Apache, lighttpd or whatever webser is used.
As I'm personally using it with Apache webserver, below I'll describe how to set it in Apache version 2.x.

3. Configure the newly configured SSL certificate to run on Apache virtualhost

Open up the virtualhost file which coresponds to the domain name to be secured with SSL, for example **/etc/apache/sites-available/www.domain.com**
Within the directives place in a code similar to:

```
SSLEngine on
# domain.com.crt cointains the wildcard SSL certificate generated and obtained by you from RapidSSL
SSLCertificateFile /etc/apache2/ssl/domain.com.pem
```

Here the file **/etc/apache2/ssl/domain.com.pem** should contain both the:

----BEGIN RSA PRIVATE KEY----- issued earlier in step one with *openssl* command, as well as:

-----BEGIN CERTIFICATE----- which will be issued by the SSL certificate reseller.

Finally it's necessery that Apache is restarted to load the new configured certificate:

```
server:~# /etc/init.d/apache2 restart
```

The above described steps need to be repeated for all the wildcard subdomains which will use the multiple SSL generated certificate and hopefully if all is well tuned, the certificates should start appearing to all the web domain subdomains immediately.