Apache - How to create a multi-domain SSL certificate

Introduction

By defult, an SSL certificate is valid for only a single domain name.

Using wildcards, you can match all the subdomains with a single certificate. However, a wildcard certificate is valid only for the subdomains, and not for the main domain: so, a certificate for **.example.com* is valid for http://www.example.com , *foo.example.com* and *bar.example.com*, but not for http://example.com .

If you need to match both the main domain and subdomains, or even different domains (i.e. http://example.com and http://example.net), you need a **multi-domain SSL certificate**.

How to create the multi-domain SSL certificate

To create the multi-domain SSL certificate you need the **openssl** libraries and application on your PC.

Basically, the commands to create a **multi-domain SSL certificate** are almost the same to create a single-domain certificate.

In this case it's required to generate a **Certificate Signing Request** (CSR) using a customized version of the OpenSSL configuration file, including in it the list of domain names (SubjectAltName) and, optionally, IP addresses.

Customize the *openssl.conf* file

Make a copy of the *openssl.conf* file (usually located in */etc/ssl/openssl.cnf*) into the working directory. You can name this file *openssl_copy.cnf*, for example.

```
cp /etc/ssl/openssl.cnf /etc/ssl/openssl_copy.cnf
```

Then open the new file with a text editor

```
vi /etc/ssl/openssl_copy.cnf
```

Search for the *[req]* section

```
/ [ req ]
```

Uncomment the *req_extensions* line removing the hash (#) on the first column:

```
req_extensions = v3_req # The extensions to add to a certificate request
```

Then search for the *[v3_req]* section

```
/ [ v3_req ]
```

add the *subjectAltName* parameter

```
subjectAltName = @alt_names
```

Finally, add at the end of the file a new section *[alt_names]* that contains all the domain names and/or IP addresses you want to include in the SSL certificate:

```
[ alt_names ]
DNS.1 = example.com
DNS.2 = www.example.com
DNS.3 = *.third.example.com
DNS.4 = example.net
DNS.5 = *.example.net
IP.1 = 1.2.3.4
IP.2 = 5.6.7.8
```

In this example, the SSL certificate will be valid for http://example.com , http://www.example.com , all the subdomains of *third.example.com* (but not for *third.example.com* itself), and http://example.net including *all* its subdomains (only the third-levels).

The certificate will be valid also for IP addresses *1.2.3.4* and *5.6.7.8*: it could be useful if the server is accessible directly via the IP address, instead of using a domain name.

Create the Certificate Signing Request (CSR)

Now you can create the key and the CSR file:

```
openssl req -newkey rsa:2048 -nodes -keyout ca.key -out ca.csr -config /etc/ssl/openssl_copy.cnf
```

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) [XX]:US
State or Province Name (full name) []:OH
Locality Name (eg, city) [Default City]:LOVELAND
Organization Name (eg, company) [Default Company Ltd]:SFL SERVICES LLC
Organizational Unit Name (eg, section) []:IT DEPT
Common Name (eg, your name or your server's hostname) []:docs.sflservicesllc.com
Email Address []:sales@sflservicesllc.com
```

Please enter the following 'extra' attributes
to be sent with your certificate request

```
A challenge password []:
An optional company name []:
```

This will create a 2048-bits key: if you need longer keys, change *rsa:2048* with the value you prefer.

You can verify the CSR file content to be sure the multiple domain names have been included:

```
openssl req -text -noout -in ca.csr
```

Finally, you can create the self-signed multi-domain SSL certificate:

```
openssl x509 -req -days 365 -in ca.csr -signkey ca.key -out ca.crt -extensions v3_req -extfile
/etc/ssl/openssl_copy.cnf
```

Copy the certificates to the default location

```
cp ca.crt /etc/pki/tls/certs
cp ca.key /etc/pki/tls/private/ca.key
cp ca.csr /etc/pki/tls/private/ca.csr
```

Apache configuration

The last step is the virtual host configuration on **Apache**:

```
<VirtualHost 1.2.3.4:443>
  ServerName www.example.com
  DocumentRoot /www
  ErrorLog logs/www.example.com-error.log
  CustomLog logs/www.example.com-access.log
  combined SSLEngine on
  SSLCertificateFile /etc/pki/tls/certs/ca.crt
  SSLCertificateKeyFile /etc/pki/tls/private/ca.key
</VirtualHost>
```

You can then restart Apache to make the changes effective.

From https://www.wizlab.it/code/apache-create-multi-domain-ssl-certificate.html

Nice to have https://serverfault.com/questions/648534/accidently-removed-localhost-crt-ssl-in-centos-6-what-can-i-do

Revision #1
Created 2 September 2024 20:26:20 by Steve Ling
Updated 2 September 2024 20:28:08 by Steve Ling