cecilemuller / 2019-https-localhost.md

Last active 2 days ago • Report abuse

Star

Code Revisions 5 Stars 554 Forks 188

How to create an HTTPS certificate for localhost domains

2019-https-localhost.md

How to create an HTTPS certificate for localhost domains

This focuses on generating the certificates for loading local virtual hosts hosted on your computer, for development only.

Do not use self-signed certificates in production ! For online certificates, use Let's Encrypt instead (tutorial).

Certificate authority (CA)

Generate `RootCA.pem` , `RootCA.key` & `RootCA.crt` :

```
openssl req -x509 -nodes -new -sha256 -days 1024 -newkey rsa:2048 -
keyout RootCA.key -out RootCA.pem -subj "/C=US/CN=Example-Root-CA"
openssl x509 -outform pem -in RootCA.pem -out RootCA.crt
```

Note that `Example-Root-CA` is an example, you can customize the name.

Domain name certificate

Let's say you have two domains `fake1.local` and `fake2.local` that are hosted on your local machine for development (using the `hosts` file to point them to `127.0.0.1`).

First, create a file `domains.ext` that lists all your local domains:

```
authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
```

```
keyUsage = digitalSignature, nonRepudiation, keyEncipherment,
dataEncipherment
subjectAltName = @alt_names
[alt_names]
DNS.1 = localhost
DNS.2 = fake1.local
DNS.3 = fake2.local
```

Generate `localhost.key`, `localhost.csr`, and `localhost.crt`:

```
openssl req -new -nodes -newkey rsa:2048 -keyout localhost.key -out
localhost.csr -subj "/C=US/ST=YourState/L=YourCity/O=Example-
Certificates/CN=localhost.local"
openssl x509 -req -sha256 -days 1024 -in localhost.csr -CA RootCA.pem -
CAkey RootCA.key -CAcreateserial -extfile domains.ext -out
localhost.crt
```

Note that the country / state / city / name in the first command can be customized.

You can now configure your webserver, for example with Apache:

```
SSLEngine on
SSLCertificateFile "C:/example/localhost.crt"
SSLCertificateKeyFile "C:/example/localhost.key"
```

Trust the local CA

At this point, the site would load with a warning about self-signed certificates. In order to get a green lock, your new local CA has to be added to the trusted Root Certificate Authorities.

Windows 10: Chrome, IE11 & Edge

Windows 10 recognizes `.crt` files, so you can right-click on `RootCA.crt > Install` to open the import dialog.

Make sure to select "Trusted Root Certification Authorities" and confirm.

You should now get a green lock in Chrome, IE11 and Edge.

Windows 10: Firefox

There are two ways to get the CA trusted in Firefox.

The simplest is to make Firefox use the Windows trusted Root CAs by going to `about:config`, and setting `security.enterprise_roots.enabled` to `true`.

The other way is to import the certificate by going to `about:preferences#privacy` > `Certificats` > `Import` > `RootCA.pem` > `Confirm for websites`.

Load earlier comments...

BraxtonI commented on 28 Nov 2020

I followed a few other tutorials which just gave me errors, and this one finally worked out, and I got my localhost certificate, but I did not have https on my localhost. I decided to delete everything I created and follow this tutorial start to finish again, and I'm getting the following error when I try to generate the localhost.key, etc:

```
problem creating object tsa_policy1=1.2.3.4.1
13632:error:08064066:object identifier routines:OBJ_create:oid
exists:crypto\objects\obj_dat.c:698:
```

matteogll commented on 30 Nov 2020

@superfein: it's an issue related to Git Bash on Windows.
Try with adding `MSYS_NO_PATHCONV=1` in order to disable PATH conversion:
`MSYS_NO_PATHCONV=1 openssl req -x509 -nodes -new -sha256 -days 1024 -newkey rsa:2048 -keyout RootCA.key -out RootCA.pem -subj "/C=US/CN=Example-Root-CA" Generating a RSA private key`

hossamhamedm… commented on 8 Dec 2020

Make a root CA:

```
openssl req -new -x509 -keyout server.key -out server.pem -days 3650 -nodes
openssl x509 -outform pem -in server.pem -out server.crt
```

ajaysbugatti commented on 28 Dec 2020

**make sure you make following changes -adding the generated RootCA.crt file to Chrome in the Authorities tab at chrome://settings/certificates.
credit -**@chrisk**
**

studious commented on 11 Jan 2021

Has anyone got this to run as a wildcard setup something like `*.test`?

I am using `CN=*.test` and domains.ext looks like this

```
authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
subjectAltName = @alt_names
[alt_names]
DNS.1 = *.test
```

In Brave I see

```
This server could not prove that it is app.test; its security certificate is from
*.test.
```

In safari I see

```
"*.test" certificate is not standards compliant
```

The CA and cert looks similar to my production ones, I just don't know the bit that is missing. I'd rather not need to make a certificate per local service.

gallusenrico commented on 21 Jan 2021

Amazing instruction! Works like a charm

studious commented on 21 Jan 2021

I found out Safari doesn't see certificates over 825days as valid. Now I get the same error essentially about the certificate using *.test

I've heard about issues using `.test` on Mac OS, but I'm wondering if wildcards need to be a minimum depth `*.example.com` rather than `*.test`

dhawal1248 commented on 23 Feb 2021

💯

BonBonSlick commented on 27 Feb 2021 • edited

Certificate Import Error
The Private Key for this Client Certificate is missing or invalid
or
The file contained one certificate, which was not imported:
localhost.local: Not a Certification Authority

BonBonSlick commented on 27 Feb 2021

Not working, it is still http even after enabling in nginx

Rendez commented on 4 Mar 2021

Thanks so much!

For **OSX** (tested in Big Sur) it's also possible to add the trusted certificate via CLI for your localhost project, here are all the steps:

```
openssl req -x509 -nodes -new -sha256 -days 1024 -newkey rsa:2048 -keyout RootCA.key -out
```

```
openssl x509 -outform pem -in RootCA.pem -out RootCA.crt
```

```
echo -n "authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
subjectAltName = @alt_names
[alt_names]
DNS.1 = localhost" > domains.ext
```

```
openssl req -new -nodes -newkey rsa:2048 -keyout localhost.key -out localhost.csr -subj "
```

```
openssl x509 -req -sha256 -days 1024 -in localhost.csr -CA RootCA.pem -CAkey RootCA.key -
```

```
sudo security add-trusted-cert -d -r trustAsRoot -k /Library/Keychains/System.keychain lo
```

Extra steps for cleanup

```
echo -n "localhost.crt\nlocalhost.key" >> .gitignore # optional !
rm domains.ext localhost.csr RootCA.* # optional !
```

ProgrammingPl… commented on 7 Mar 2021

hi guys it didn't work for me. kindly help me.

my setup -
Windows 10 OS
apache2(on ubuntu server) installed in VirtualBox VMS.
generated for domain linux.vm
in my windows hosts file - I added an entry for vm IP to linux.vm domain.
also i install rootCA.crt file in windows 10. by right click -> install

Note - my apache2 is accessible on windows 10's chrome without https, but when i tried https, it gives error that "**this certificate can't be verified upto a trusted authority.**"

elliott-fwdsec commented on 11 Mar 2021

You can also add `IP.1 = 127.0.0.1` under `[alt_name]`

terrylinooo commented on 21 Apr 2021

Thank you. Works like a charm.

pstanton commented on 26 Apr 2021

I'm trying to get local tomcat working via ssl and following the above i have

localhost.crt
localhost.csr
localhost.key

but i need key.pem, cert.pem and chain.pem

```
<Connector port="443" protocol="org.apache.coyote.http11.Http11AprProtocol"
           maxThreads="150" SSLEnabled="true" >
    <UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol" />
    <SSLHostConfig>
        <Certificate certificateKeyFile="conf/localhost-rsa-key.pem"
                     certificateFile="conf/localhost-rsa-cert.pem"
                     certificateChainFile="conf/localhost-rsa-chain.pem"
                     type="RSA" />
    </SSLHostConfig>
</Connector>
```

What have i missed?

VladTitSf9 commented on 8 May 2021

> First, create a file domains.ext that lists all your local domains:

where this file should be created?

tranthaihoang commented on 13 May 2021

> Need an easy solution: https://github.com/FiloSottile/mkcert

good! easy and fast, thank you!

hasoxy commented on 25 May 2021

The only one that worked after trying multiple solutions, easy and clear, thanks man

Robin-Sch commented on 5 Jul 2021

Is it possible to renew (and add new domains), or would you have to redo everything?

jordygrunn commented on 19 Aug 2021

Thank you for sharing! It works great for my local development environment

GitHub-Mike commented on 20 Aug 2021

> > First, create a file domains.ext that lists all your local domains:
>
> where this file should be created?

In the same directory where the openssl command is executed.

web-bert commented on 6 Oct 2021

Thank you for this, I had to reduce the validity for the `localhost.crt` to `397` days as that is now the maximum validity period. See this post about it, it might be worth updating the command with the new maximum.

rinogo commented on 15 Oct 2021

> Need an easy solution: https://github.com/FiloSottile/mkcert

Thank you so much! For a quick fix on local development environments, this beats messing with `openssl`.

perki commented on 2 Nov 2021

If you need ready to use SSL certificates for localhost you can have a look at https://github.com/pryv/rec-la
https://.rec.la/ => https://localhost/

praveeenpatel commented on 2 Nov 2021

Certificate import error

The Private Key for this Client Certificate is missing or invalid

when i import certificate in chrome browser.
https://prnt.sc/1y2i7ev

serkanalgur commented on 4 Nov 2021

Still Working!. Thanks **@cecilemuller**

GeoffCapper commented on 17 Dec 2021

Note that if you are generating for localhost, in the commands for "Generate localhost.key, localhost.csr, and localhost.crt:" the CN in the first command should be ".../CN=localhost", not ".../CN=localhost.local" otherwise Chrome (and maybe others) won't like it.

jaami commented on 7 Jan

Command: openssl x509 -req -sha256 -days 1024 -in localhost.csr -CA RootCA.pem -CAkey RootCA.key -CAcreateserial -extfile domains.ext -out localhost.crt

OutPut: C:\xampp\htdocs\SSL>openssl x509 -req -sha256 -days 1024 -in localhost.csr -CA RootCA.pem -CAkey RootCA.key -CAcreateserial -extfile domains.ext -out localhost.crt
Can't open "domains.ext" for reading, No such file or directory
78030000:error:80000002:system library:BIO_new_file:No such file or directory:crypto\bio\bss_file.c:67:calling fopen(domains.ext, r)
78030000:error:10000080:BIO routines:BIO_new_file:no such file:crypto\bio\bss_file.c:75:

for me all steps work smoothly otherwise useless effort wont help because things are new and confusing.

GitHub-Mike commented on 8 Jan

@jaami: The domains.ext file must be located in the directory where the command is executed. Also check the file permissions for the executing user.

rajan-31 commented on 19 Jan

I followed this https://ritesh-yadav.github.io/tech/getting-valid-ssl-certificate-for-localhost-from-letsencrypt/

It's kind of a hack with heroku.

If you are going to follow that then some tips:

- you have to implement the route asked in certbot console output, in your heroku app (not the app running on localhost. So, that letsencrypt can access that from their server)
- add domain in `/etc/hosts` at the end