Open in app ↗ Sign up Sign in

Medium Search

How to generate a self-signed SSL certificate for an IP address

4 min read · Apr 25, 2020

Dimitri Witkowski Follow

Listen Share

Generating a self-signed certificate for a hostname is easy, but it gets more complicated if you would like to do the same for an IP address.

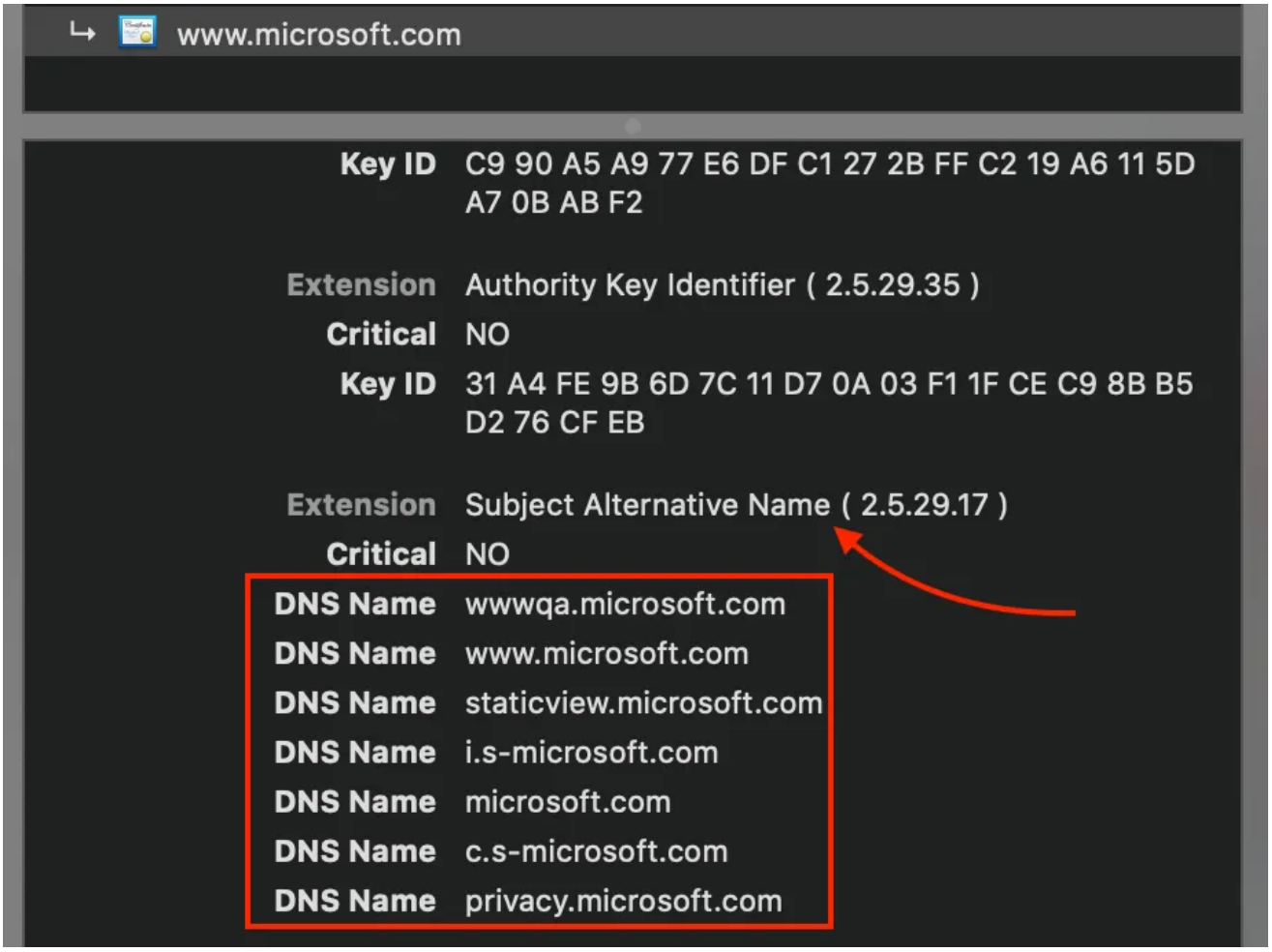
Although it's not very common, issuing a certificate for an IP address is possible.

Subject Alternative Name extension

Subject Alternative Name extension is an extension of the X.509 specification described in RFC 5280, section 4.2.1.6 as follows:

> *The subject alternative name extension allows identities to be bound to the subject of the certificate. These identities may be included in addition to or in place of the identity in the subject field of the certificate. Defined options include an Internet electronic mail address, a DNS name, an IP address, and a Uniform Resource Identifier (URI).*

For example, here's how Microsoft is using it:



SAN on www.microsoft.com

SAN can be used to issue certificates not only for multiple hostnames, but also for IP addresses.

Generating a self-signed certificate with OpenSSL

To generate a certificate with SAN extension using OpenSSL, we need to create a config first. Here's what it can look like:

```
[req]
default_bits  = 2048
distinguished_name = req_distinguished_name
req_extensions = req_ext
x509_extensions = v3_req
prompt = no

[req_distinguished_name]
countryName = XX
stateOrProvinceName = N/A
localityName = N/A
organizationName = Self-signed certificate
commonName = 120.0.0.1: Self-signed certificate
```

```
[req_ext]
subjectAltName = @alt_names

[v3_req]
subjectAltName = @alt_names

[alt_names]
IP.1 = 127.0.0.1
```

An important part here is the last one, where the IP address is set. It's also possible to add additional IP addresses and hostnames in this section.

Get Dimitri Witkowski's stories in your inbox

Join Medium for free to get updates from this writer.

Enter your email

Subscribe

Remember me for faster sign in

Save this config as `san.cnf` and pass it to OpenSSL:

```
openssl req -x509 -nodes -days 730 -newkey rsa:2048 -keyout key.pem -
out cert.pem -config san.cnf
```

This will create a certificate with a private key. Let's inspect it:

```
openssl x509 -in cert.pem -text -noout
```

The output should contain the IP address from the config:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 12776076004935403308 (0xb14dbd6f9991072c)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=XX, ST=N/A, L=N/A, O=Self-signed certificate,
CN=127.0.0.1: Self-signed certificate
        Validity ...
        Subject: C=XX, ST=N/A, L=N/A, O=Self-signed certificate,
CN=127.0.0.1: Self-signed certificate
        Subject Public Key Info: ...
        X509v3 extensions:
            X509v3 Subject Alternative Name:
```

```
                IP Address:127.0.0.1
        Signature Algorithm: sha256WithRSAEncryption ...
```

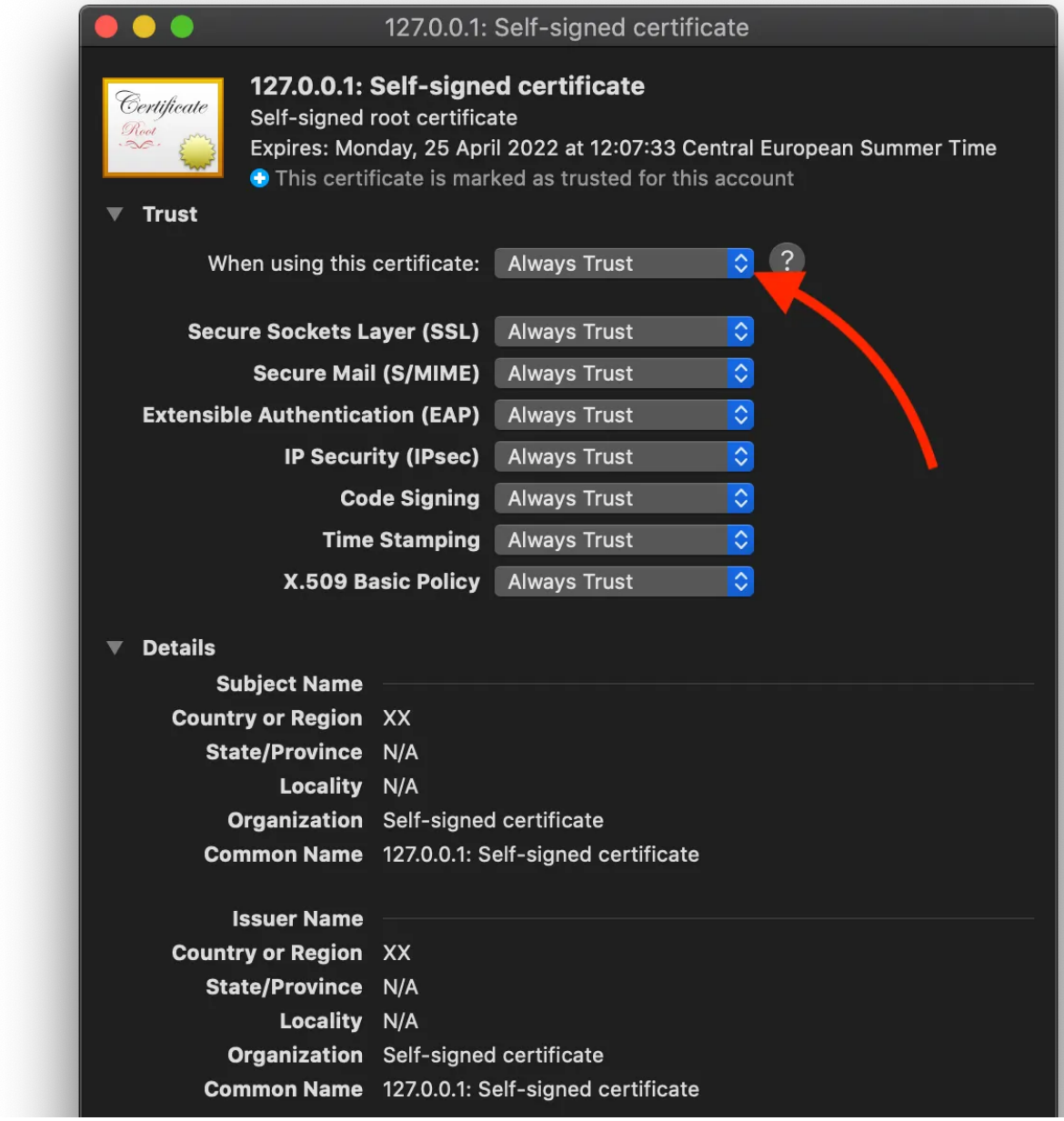
Using the certificate

Now you can install and use the certificate in the same way you would use any other SSL certificate, for example, an easy way to serve static files is using http-server npm package:

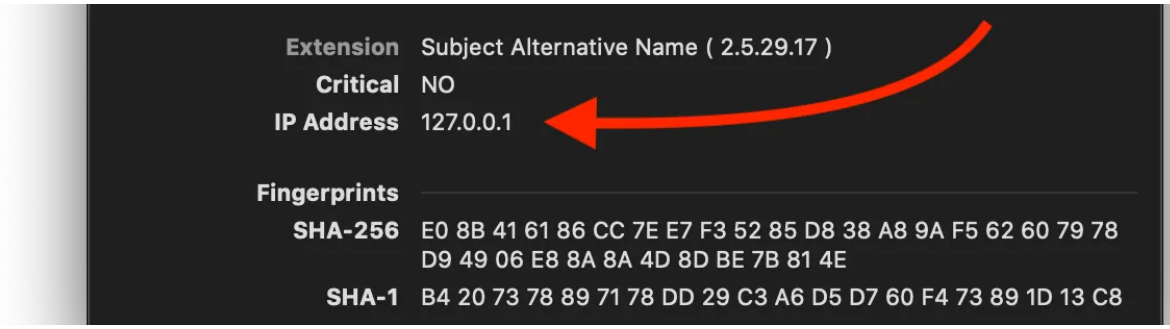
```
npx http-server --ssl
```

It reads `cert.pem` and `key.pem` automatically if `--ssl` option is specified.

To avoid adding a security exception in browsers, you can trust the certificate. On macOS the certificate looks like this:

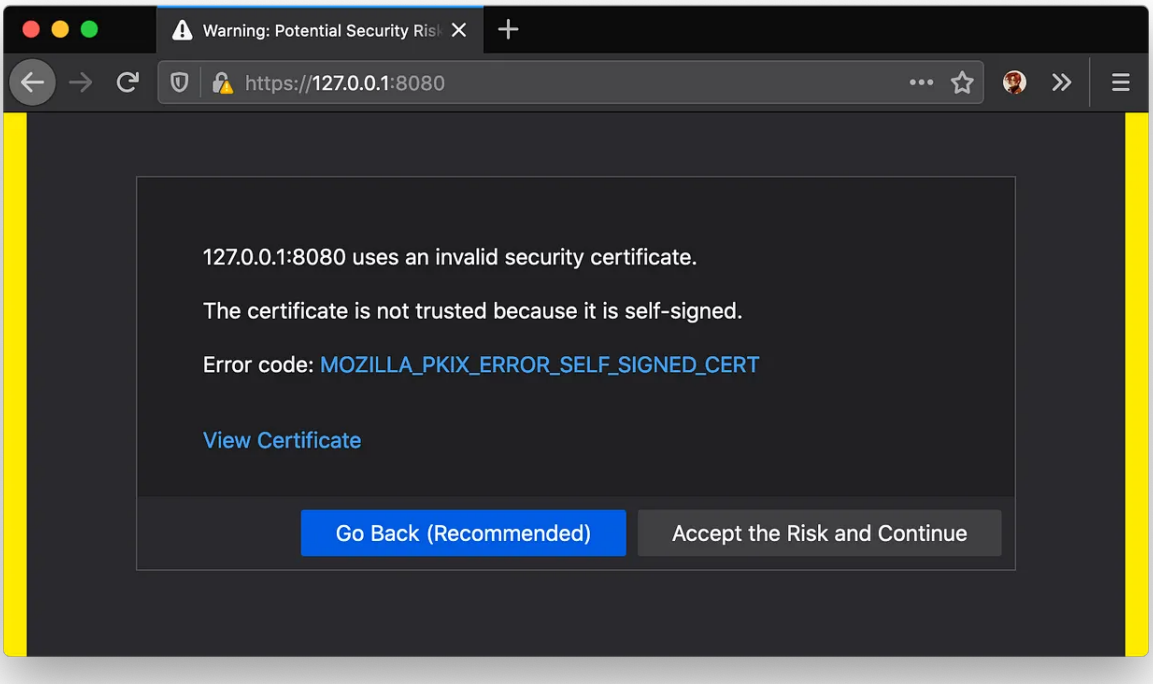


Self-signed certificate on macOS



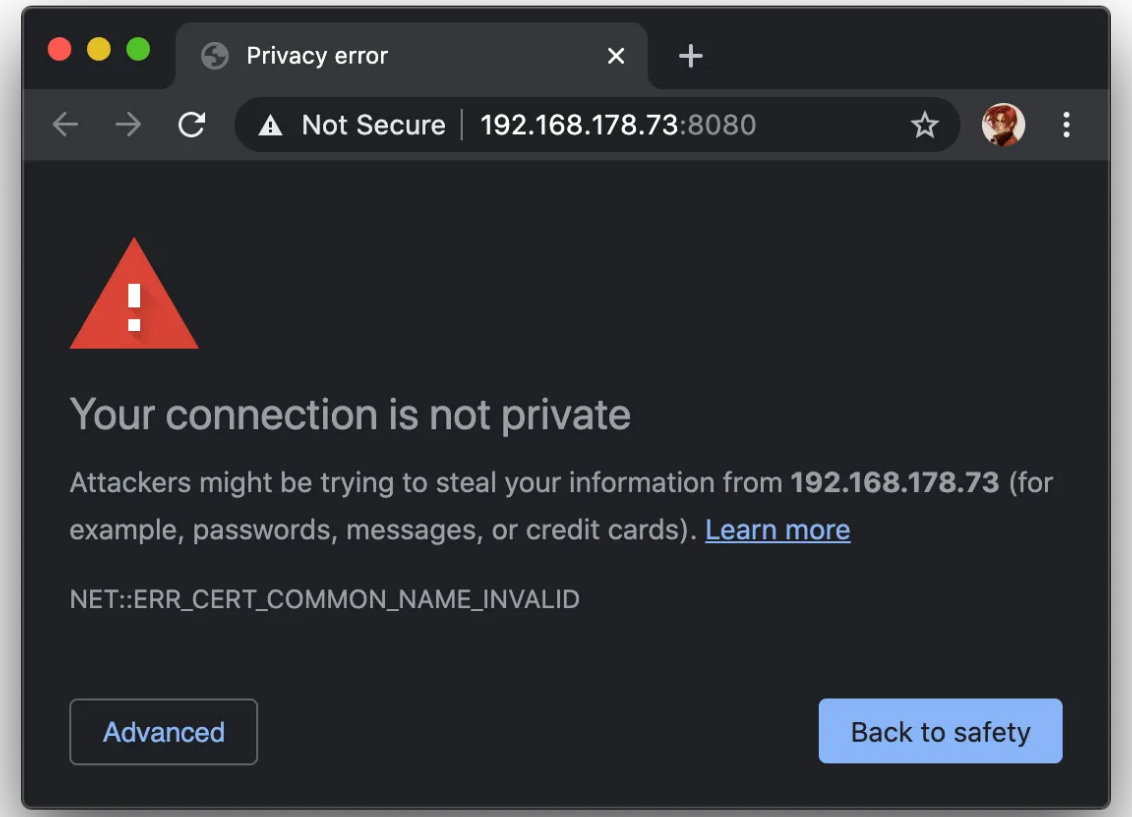
SAN extension of the certificate

If the certificate is trusted in the Keychain, Chrome and Safari won't complain about it. Firefox shows an additional warning about self-signed certificates:



Self-signed certificate warning in Firefox

If you try to load a website with the same certificate from another IP, you should get an error like this:



Using a SAN certificate from another IP

Putting it all together

To make it easier to generate a certificate for an IP address, I put it all together in one script that can be found on GitHub. It will create a config, put your IP address there, and generate a certificate with a private key for you:

```
curl -sS https://raw.githubusercontent.com/antelle/generate-ip-cert/master/generate-ip-cert.sh | bash -s 127.0.0.1
```

References

If you're curious about SAN, these articles can give some very basic info about it:

SAN Certificates: Subject Alternative Name - Multi-Domain (SAN)

The Subject Alternative Name field lets you specify additional host names (sites, IP addresses, common names, etc.) to...



www.digicert.com



What is the Subject Alternative Name (SAN)? - DNSimple Help

The Subject Alternative Name (SAN) is an extension to the X.509 specification that allows users to specify additional...

support.dnsimple.com

Know about SAN Certificate and How to Create With OpenSSL

A powerful cloud-based vulnerability scanner that finds security flaws in your digital estate, explains the risks &...

geekflare.com



Openssl | Self Signed Certificate | Https | Ssl | Ssl Certificate



Follow

Written by Dimitri Witkowski

11 followers · 1 following

Full stack software developer from the Netherlands. https://antelle.net