Using Self-Signed Certificates with mkcert

Overview

We do not recommend using self-signed certificates with Element Enterprise On-Premise, however, we recognize that there are times when self-signed certificates can be the fastest way forward for demo or PoC purposes. It is in this spirit that these directions are provided.

Steps

The following instructions will enable you to use a tool called mkcert to generate self-signed certificates. Element does not ship this tool and so these directions are provided as one example of how to get self-signed certificates.

Ubuntu:

```
sudo apt-get install wget libnss3-tools
```

EL:

```
sudo yum install wget nss-tools -y
```

Both EL and Ubuntu:

```
wget -O mkcert "https://dl.filippo.io/mkcert/latest?for=linux/amd64"
sudo mv mkcert /usr/bin/
sudo chmod +x /usr/bin/mkcert
```

Once you have mkcert executable, you can run:

```
mkcert -install
The local CA is now installed in the system trust store! ⚡
```

Now, you can verify the CA Root by doing:

```
mkcert -CAROOT
/home/element-demo/.local/share/mkcert
```

Your output may not be exactly the same, but it should be similar. Once we’ve done this, we need to generate self-signed certificates for our hostnames.

You can either do this by generating a wildcard certificate that works for all subdomains or you can do this per domain.

The following is an example for how to build a wildcard cert for `element.local`. You will only need to run this once and then you can use the generated certificate for all hostnames that require a certificate:

```
mkcert *.element.local element.local 192.168.122.39 127.0.0.1


Created a new certificate valid for the following names - "*.element.local"
 - "element.local"
 - "192.168.122.39"
 - "127.0.0.1"


Reminder: X.509 wildcards only go one level deep, so this won't match a.b.element.local ℹ


The certificate is at "./_wildcard.element.local+3.pem" and the key at "./_wildcard.element.local+3-key.pem"
It will expire on 5 July 2025
```

The following is an example of how to do it for `element.local`. You will need to do this for all of the aforementioned hostnames, including the `fqdn.tld`.

The run for the element fqdn looks like this:

```
mkcert element.local element 192.168.122.39 127.0.0.1


Created a new certificate valid for the following names
- "element.local"
- "element"
- "192.168.122.39"
- "127.0.0.1"


The certificate is at "./element.local+3.pem" and the key at
"./element.local+3-key.pem"


It will expire on 1 May 2024
```

Once you have self-signed certificates, you need to rename them for each host with the form of `fqdn.crt` and `fqdn.key`.

Using our above example, these are the commands we would need to run from the installer directory just for the `element.local` certificate: (We ran `mkcert` in that directory as well.)

```
cp element.local+3.pem  element.local.crt
cp element.local+3-key.pem  element.local.key
```

In the case of the wildcard certificate, we could run:

```
cp ./_wildcard.element.local+3.pem wildcard.element.local.crt
cp ./_wildcard.element.local+3-key.pem wildcard.element.local.key
```

and then use this file where needed in the graphical installer for a crt/key pair.

Revision #4
Created 28 July 2022 19:17:37 by Karl Abbott
Updated 6 November 2024 12:49:27 by Kieran Mitchell Lane