Linux Cert Management

The easy way to manage certificates is navigate to chrome://settings/search#ssl. Then click on the “Manage Certificates” button. This will load a built-in interface for managing certificates.

On Linux, Chromium uses the NSS Shared DB. If the built-in manager does not work for you then you can configure certificates with the NSS command line tools.

Details

Get the tools

- Debian/Ubuntu: `sudo apt-get install libnss3-tools`
- Fedora: `su -c "yum install nss-tools"`
- Gentoo: `su -c "echo 'dev-libs/nss utils' >> /etc/portage/package.use && emerge dev-libs/nss"` (You need to launch all commands below with the `nss` prefix, e.g., `nsscertutil`.)
- Opensuse: `sudo zypper install mozilla-nss-tools`

List all certificates

```
certutil -d sql:$HOME/.pki/nssdb -L
```

Ubuntu Jaunty error

Above (and most commands) gives:

```
certutil: function failed: security library: invalid arguments.
```

Package version 3.12.3.1-0ubuntu0.9.04.2

List details of a certificate

```
certutil -d sql:$HOME/.pki/nssdb -L -n <certificate nickname>
```

Add a certificate

```
certutil -d sql:$HOME/.pki/nssdb -A -t <TRUSTARGS> -n <certificate nickname> \
-i <certificate filename>
```

The TRUSTARGS are three strings of zero or more alphabetic characters, separated by commas. They define how the certificate should be trusted for SSL, email, and object signing, and are explained in the certutil docs or Meena's blog post on trust flags.

For example, to trust a root CA certificate for issuing SSL server certificates, use

```
certutil -d sql:$HOME/.pki/nssdb -A -t "C,," -n <certificate nickname> \
-i <certificate filename>
```

To import an intermediate CA certificate, use

```
certutil -d sql:$HOME/.pki/nssdb -A -t ",," -n <certificate nickname> \
-i <certificate filename>
```

Note: to trust a self-signed server certificate, we should use

```
certutil -d sql:$HOME/.pki/nssdb -A -t "P,," -n <certificate nickname> \
-i <certificate filename>
```

This should work now, because NSS bug 531160 is claimed to be fixed in a related bug report. If it doesn't work, then to work around the NSS bug, you have to trust it as a CA using the “C,,” trust flags.

Add a personal certificate and private key for SSL client authentication

Use the command:

```
pk12util -d sql:$HOME/.pki/nssdb -i PKCS12_file.p12
```

to import a personal certificate and private key stored in a PKCS #12 file. The TRUSTARGS of the personal certificate will be set to “u,u,u”.

Delete a certificate

```
certutil -d sql:$HOME/.pki/nssdb -D -n <certificate nickname>
```