SVN client certificate guide

Author: Jan Dittberner <jandd@cacert.org>
Version: 0.1
Date: 2010-06-20

Contents

Getting a client certificate **1**
Generating a private key and CSR using OpenSSL 1
Generating a private key and CSR using GNUTLS 1
Creating a PKCS#12 key store **2**
Export PKCS#12 key store from Mozilla Firefox 2
Generate PKCS#12 key store using OpenSSL 2
Generate PKCS#12 key store using GNUTLS 2
Configure clients to use certificates **3**
Command line SVN client 3
TortoiseSVN 3
Eclipse 3

Getting a client certificate

Before using client certificate authentication you obviously need a client certificate. There are several ways to get one from CAcert.org.

1. use a capable web browser like Mozilla Firefox using the client certicate at the client certificate URL
2. generate a private key, and a CSR (certificate signing request) either using OpenSSL or GNUTLS, I will describe both variants in the two sections below

Generating a private key and CSR using OpenSSL

OpenSSL supports a one shot operation to generate both a private key and a CSR:

```
openssl req -new -newkey rsa:2048 -keyout clientcert.key \
  -out clientcert.csr \
  -subj '/CN=Jan Dittberner/emailAddress=jandd@cacert.org'
```

you need to enter a PEM passphrase that is used to protect the private key twice. The CSR is put into cliencert.csr and can be pasted into the CSR field at the bottom of the page at the client certificate URL.

You should copy the generated certificate block into a file `clientcert.pem` that we will use later.

Generating a private key and CSR using GNUTLS

GNUTLS comes with `certtool` that can be used to generate a private key and CSR as follows:

```
certtool --generate-privkey --outfile clientcert.key
echo -e 'cn="Jan Dittberner"\nemail="jandd@cacert.org"\ntls_www_client' \
  > cert.cfg
certtool --generate-request --load-privkey clientcert.key \
  --outfile clientcert.csr --template cert.cfg
```

You can also specify the cn and email fields manually but I prefer to use the template approach. The CSR is put at the bottom of cliencert.csr and can be pasted into the CSR field at the bottom of the page at the client certificate URL.

You should copy the generated certificate block into a file `clientcert.pem` that we will use later.

Creating a PKCS#12 key store

Subversion (SVN) uses PKCS#12 keystores for client certificate storage. Depending on how your subversion client is built you should use either OpenSSL or GNUTLS to create your PKCS#12 file. You can also export a PKCS#12 file from Mozilla Firefox (or possibly other browsers). I will describe these three variants here and assume that you used the instructions above for creating your private key and client certificate.

Export PKCS#12 key store from Mozilla Firefox

Open Firefox and perform the following steps:

- Tools
- Options ...
- Advanced
- Encryption
- View certificates
- Your certificates
- select your CAcert.org client certificate and click on "Backup ..."
- choose a file name and save the .p12 file

Generate PKCS#12 key store using OpenSSL

Use the following command line to generate a PKCS#12 file from your existing `clientcert.key` and `clientcert.pem`:

```
openssl pkcs12 -export -inkey clientcert.key -in clientcert.pem \
  -out clientcert.p12
```

Enter the passphrase for the private key and a new passphrase for your key store.

Generate PKCS#12 key store using GNUTLS

Use the following command line to generate a PKCS#12 file from your existing `clientcert.key` and `clientcert.pem`:

```
certtool --to-p12 --load-privkey clientcert.key \
  --load-certificate clientcert.pem \
  --outraw --outfile clientcert.p12
```

You have to enter a name for the key (i.e. your name) and a password to protect the key store.

Configure clients to use certificates

Command line SVN client

The command line SVN client uses a file `servers` inside its application setting directory (`$HOME/.subversion` on Unix like operating systems and `%APPDATA%\Subversion` on Microsoft Windows).

The file should contain something like this:

```
[groups]
cacert = svn.cacert.org

[cacert]
ssl-client-cert-file = /absolute/path/to/clientcert.p12
ssl-authority-files = /absolute/path/to/cacert.pem,/absolute/path/to/root.pem
```

If you don't want the subversion client to store your key store's password you should also put the following line into the `auth` section of `$HOME/.subversion/config` (or its Windows equivalent):

```
[auth]
store-passwords = no
store-auth-creds = no
```

TortoiseSVN

TortoiseSVN uses the same configuration as the command line client. You can edit the `servers` file from TortoiseSVN's "Settings" -> "Network" dialog.

Eclipse

To use Subversion in Eclipse you should use the Subversive SVN Team provider from http://www.eclipse.org/subversive/ with the JavaHL or SVNKit connector. You get asked to install a connector when you first try to access a subversion repository.

After installing Subversive and a SVN connector you can add the Subversion repository by opening the SVN Repository Explorer perspective ("Window" -> "Open Perspective" -> "Other ..." -> "SVN Repository Exploring" -> "OK") and pressing the "New repository location" button in the "SVN Repositories" view.

In the Dialog:

- on Tab "General"
    - enter the Repository URL in "URL"
- on Tab "SSL Settings"
    - check "Enable Client Authentication"
    - select your .p12-File using the "Browse ..." button and file dialog
    - enter the passphrase to test the connection
    - optionally check "Save passphrase" (be aware that this is not a good idea)
- click "Finish"