NAME

pki --estca - Get CA certificate[s] from an EST server

SYNOPSIS

pki --estca--url *url* [**--label** *label*] **--cacert** *file* [**--caout** *file*] [**--outform** *encoding*] [**--force**] [**--debug** *level*]

pki --estca--options *file*

pki --estca-h | **--help**

DESCRIPTION

This sub-command of **pki**(1) gets CA certificates via https from an EST server using the */cacerts* operation of the Enrollment over Secure Transport protocol (RFC 7030).

OPTIONS

-h, --help

Print usage information with a summary of the available options.

-v, --debug *level*

Set debug level, default: 1.

-+, --options *file*

Read command line options from *file*.

-u, --url *url*

URL of the SCEP server.

-l, --label *label*

Label in the EST server path.

-C, --cacert *file*

CA certificate in the trust chain used for EST TLS server signature verification. Can be used multiple times.

-c, --caout *file*

If present, path where the fetched root CA certificate file is stored to. If several CA certificates are downloaded, then the value of **--caout** is used as a template to derive unique filenames (*-1, *-2, etc.) for the intermediate or sub CA certificates. If a file suffix is missing, then depending on the value of **--outform** either *.der* (the default) or *.pem* is automatically appended. If the **--caout** option

is missing and **--outform** is set to *pem* then a PEM-encoded CA certificate bundle is written to *stdout*.

-f, --outform *encoding*
Encoding of the created certificate file. Either *der* (ASN.1 DER) or *pem* (Base64 PEM), defaults to *der*.

-F, --force
Force overwrite of existing files.

EXAMPLES

To save some typing work the following command line options are stored in a *est.opt* file:

```
--url https://pki.strongswan.org:8443
--cacert tlsca.crt
--cacert tlsca-1.crt
```

NOTE: For a successful HTTPS connection, trust must be established into the EST server certificate. The TLS trust chain including the root CA certificate and optionally intermediate CA certificates must be given using multiple **--cacert** options.

An EST server sends a root CA and an intermediate CA certificate:

```
pki --estca --options est.opt --caout myca.crt

negotiated TLS 1.3 using suite TLS_AES_256_GCM_SHA384
received TLS server certificate 'C=CH, O=strongSwan Project, CN=pki.strongswan.org'
  using certificate "C=CH, O=strongSwan Project, CN=pki.strongswan.org"
  using trusted intermediate ca certificate "C=CH, O=strongSwan Project, CN=strongSwan Issuing CA"
  using trusted ca certificate "C=CH, O=strongSwan Project, CN=strongSwan Root CA"
  reached self-signed root ca with a path length of 1
Root CA cert "C=CH, O=strongSwan Project, CN=strongSwan Root CA"
  serial: 65:31:00:ca:79:da:16:6b:aa:ac:89:e2:a8:f9:49:c3:10:ab:64:54
  SHA256: 96:70:50:51:cd:b9:e7:94:6b:04:f6:15:45:80:fc:90:85:01:71:2a:f6:4f:d1:1b:2d:a1:7e:eb:bf:dd:be:86
  SHA1  : 8e:f3:78:b0:34:a6:c1:6a:7b:c6:f5:91:eb:e5:46:9b:0d:0a:a7:ba (jvN4sDSmwWp7xvWR6+VGmw0Kp7o)
Root CA equals trusted TLS Root CA
Root CA cert is trusted, valid until Aug 12 15:51:34 2032, 'myca.crt'
Sub CA cert "C=CH, O=strongSwan Project, CN=strongSwan Issuing CA"
  serial: 74:f9:7e:72:7d:b8:fd:f2:c6:e5:1b:fa:37:f9:cb:87:bf:9c:ea:e2
  SHA256: a3:5b:4b:12:d5:8f:68:7b:05:11:08:27:f5:42:62:b8:b5:01:1b:19:37:9c:28:78:5d:37:08:69:6a:8c:07:bf
```

SHA1 : 8c:e6:67:67:c2:23:89:7b:d0:bc:b1:50:d2:1c:bc:8d:8d:69:15:11 (jOZnZ8IjiXvQvLFQ0hy8jY1pFRE)
using certificate "C=CH, O=strongSwan Project, CN=strongSwan Issuing CA"
using trusted ca certificate "C=CH, O=strongSwan Project, CN=strongSwan Root CA"
reached self-signed root ca with a path length of 0
Sub CA cert is trusted, valid until Aug 12 15:51:34 2027, 'myca-1.crt'

NOTE: The trusthworthiness of the root CA certificate is either verified automatically if the Root CA certificate of the TLS trust chain is the same as that of the Issuing CA. Otherwise trust has to be established manually by verifying the SHA256 or SHA1 fingerprint of the DER-encoded certificate that is e.g. listed on the official PKI website or by some other means.

The stored certificate files in DER format can be overwritten by PEM-encoded versions with:

pki --estca --options est.opt --caout myca.crt --outform pem --force

A CA certificate bundle in PEM format is written to *stdout*:

pki --estca --options est.opt --outform pem > cacerts.pem

SEE ALSO

pki(1)