NAME

pki --scepca - Get CA [and RA] certificate[s] from a SCEP server

SYNOPSIS

pki --scepca--url *url* [**--caout** *file*] [**--raout** *file*] [**--outform** *encoding*] [**--force**] [**--debug** *level*]

pki --scepca--options *file*

pki --scepca-h | **--help**

DESCRIPTION

This sub-command of **pki**(1) gets CA and RA certificates via http from a SCEP server using the *GetCACert* command of the Simple Certificate Enrollment Protocol (RFC 8894).

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.

-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*.

-r, --raout *file*

If present, path where the fetched RA certificate file is stored to. If multiple RA certificates are available, then the value of **--raout** is used as a template to derive unique filenames (*-2, etc.). If the **--raout** option is missing, then the value of **--caout** is used as a template to derive unique filenames (*-ra, *-ra-2, etc.) for the RA certificates. If a file suffix is missing, then depending on

the value of **--outform** either *.der* (the default) or *.pem* is automatically appended.

-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

A SCEP server sends a root CA and an intermediate CA certificate as well as an RA certificate:

```
pki --scepca --url http://pki.strongswan.org:8080/scep --caout myca.crt --raout myra.crt

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 cert is untrusted, 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'
RA cert "C=CH, O=strongSwan Project, CN=SCEP RA"
  serial: 74:f9:7e:72:7d:b8:fd:f2:c6:e5:1b:fa:37:f9:cb:87:bf:9c:ea:e3
  SHA256: 57:22:f3:13:69:2f:24:82:12:59:8e:05:63:0b:f5:a8:fb:4e:78:87:8d:68:d1:4c:c1:c4:b5:85:db:bb:64:df
  SHA1  : bc:d1:46:76:55:7f:8c:d1:c5:22:31:b9:d7:b1:49:b5:95:a4:f3:ea (vNFGdlV/jNHFIjG517FJtZWk8+o)
  using certificate "C=CH, O=strongSwan Project, CN=SCEP RA"
  using untrusted intermediate 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
RA cert is trusted, valid until Aug 10 15:51:34 2023, 'myra.crt'
```

The trusthworthiness of the root CA certificate 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 --scepca --url http://pki.strongswan.org:8080/scep --caout myca.crt --raout myra.crt \
        --outform pem --force
```

If the **--raout** option is omitted and the **--caout** template doesn't have a file suffix, then with **--outform** *pem* the following filenames are derived:

```
pki --scepca --url http://pki.strongswan.org:8080/scep --caout scep/myca --outform pem

Root CA cert "C=CH, O=strongSwan Project, CN=strongSwan Root CA"
  ...
Root CA cert is untrusted, valid until Aug 12 15:51:34 2032, written to 'scep/myca.pem'
Sub CA cert "C=CH, O=strongSwan Project, CN=strongSwan Issuing CA"
  ...
Sub CA cert is trusted, valid until Aug 12 15:51:34 2027, 'scep/myca-1.pem'
RA cert "C=CH, O=strongSwan Project, CN=SCEP RA"
  ...
RA cert is trusted, valid until Aug 10 15:51:34 2023, 'scep/myca-ra.pem'
```

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

```
pki --scepca --url http://pki.strongswan.org:8080/scep --raout myra.crt --outform pem > cacerts.pem
```

SEE ALSO

pki(1)