NAME

pki --ocsp - OCSP request parser and OCSP responder.

SYNOPSIS

pki --ocsp[**--in** *file*] [**--cacert** *file*] [**--debug** *level*]

pki --ocsp--respond [**--in** *file*] **--cacert** *file* [**--key** *file*|**--keyid** *hex***]** [**--cert** *file*|**--certid** *hex***]** [**--index** *file*] [**--lifetime** *minutes*] [**--digest** *digest*] [**--rsa-padding** *padding*] [**--debug** *level*]

pki --ocsp--options *file*

pki --ocsp-h | **--help**

DESCRIPTION

This sub-command of **pki**(1) parses an **Online Certificate Status Protocol** (OCSP) request as defined by RFC 6960 and with the **--respond** option generates an OCSP response based on the OCSP request. The certificate status for this may be provided by plugins so if not using the **--index** option, this currently requires the **openxpki** and **mysql** plugins in order to directly access the internal **certificate** database of an **OpenXPKI** (https://openxpki.org) server.

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*.

-i, --in *file*

OCSP request. If not given, the OCSP request is read from *STDIN*.

-C, --cacert *file*

CA certificate corresponding to one of the issuer hashes contained in the OCSP request. If the OCSP request is signed, a CA certificate forming the trust chain. Can be used multiple times.

-k, --key *file*

OCSP signer key. Can be used multiple times.

-K, --keyid *hex*
Smartcard or TPM 2.0 OCSP signer key object handle. Can be used multiple times.

-c, --cert *file*
OCSP signer certificate (if it is not a CA certificate). Can be used multiple times.

-X, --certid *hex*
Smartcard or TPM 2.0 OCSP signer certificate object handle. Can be used multiple times.

-x, --index *file*
OpenSSL-style index.txt providing information about the status of certificates issued by the CA certificate loaded immediately before. Can be used multiple times if the status for multiple CAs should be provided, just make sure to pass each index.txt file right after the corresponding CA certificate.

See below for a description of the structure of these files.

-l, --lifetime *minutes*
Validity in minutes of the OCSP response (if missing, nextUpdate is omitted).

-g, --digest *digest*
Digest to use for signature creation. One of *md5*, *sha1*, *sha224*, *sha256*, *sha384*, or *sha512*, *sha3_224*, *sha3_256*, *sha3_384*, *sha3_512*. The default is determined based on the type and size of the OCSP signing key.

-R, --rsa-padding *padding*
Padding to use for RSA signatures. Either *pkcs1* or *pss*, defaults to *pkcs1*.

INDEX.TXT DESCRIPTION

Each line in an index.txt file consists of six columns that are separated by tab characters:

The first column denotes the certificate status, which can be either "V" (for valid), "E" (for expired, treated like valid), or "R" (for revoked).

The second column contains the certificate’s expiration date and time in UTC in the format YYMMDDHHMMSSZ. This field is ignored by the command but must not be empty.

The third column is the revocation date and time in UTC in the format YYMMDDHHMMSSZ and an optional revocation reason that immediately follows it, separated by a comma. Valid reasons are "keyCompromise", "CACompromise", "affiliationChanged", "superseded", "cessationOfOperation",

"certificateHold", and "removeFromCRL", any other value or omitting a reason results in "unspecified".

The fourth column contains the certificate's serial number in hexadecimal encoding.

The fifth and sixth columns are both ignored by the command, so they may be omitted completely. They can contain a path to the certificate (usually set to "unknown") and the certificate's subject DN with slashes separating the RDNs.

Example index.txt:

```
V   310930122422Z       03  unknown /C=CH/O=strongSwan/CN=moon...
V   310930122422Z       04  unknown /C=CH/O=strongSwan/CN=sun...
R   310930122422Z   231002122422Z,keyCompromise 88
V   Z       05
```

Note that the fields are separated by tabs. So if a certificate is valid, two tabs follow after the expiration date. The third line in this example only specifies the relevant first four columns, the fourth even uses a dummy expiration date.

EXAMPLES

Show the raw content of an OCSP request:

```
pki --ocsp --in req_ca.der
```

```
nonce:              5b:14:e3:cc:d5:b2:65:ec:c4:0d:c3:11:37:6a:9d:71
  issuerKeyHash:    b6:76:79:95:b5:58:...:06:93:f3:39:79:19 (no match)
  issuerNameHash:   af:25:78:ce:fc:15:..:67:95:81:31:a3:4d (no match)
  serialNumber:     4f:33:21:1d:4d:fd:9b:db
  issuerKeyHash:    b6:76:79:95:b5:58:...:06:93:f3:39:79:19 (no match)
  issuerNameHash:   af:25:78:ce:fc:15:..:67:95:81:31:a3:4d (no match)
  serialNumber:     68:f2:93:10:65:d0:5e:d1
```

Show the content of the same OCSP request if the issuer certificate is given:

```
pki --ocsp --in req_ca.der --cacert cacert.pem
```

```
nonce:              5b:14:e3:cc:d5:b2:65:ec:c4:0d:c3:11:37:6a:9d:71
issuer:           "C=CH, O=strongSwan Project, CN=strongSwan Root CA"
  issuerKeyHash:    b6:76:79:95:b5:58:...:06:93:f3:39:79:19 (ok)
```

```
  issuerNameHash:    af:25:78:ce:fc:15:..:67:95:81:31:a3:4d (ok)
  serialNumber:      4f:33:21:1d:4d:fd:9b:db
issuer:            "C=CH, O=strongSwan Project, CN=strongSwan Root CA"
  issuerKeyHash:     b6:76:79:95:b5:58:..:06:93:f3:39:79:19 (ok)
  issuerNameHash:    af:25:78:ce:fc:15:..:67:95:81:31:a3:4d (ok)
  serialNumber:      68:f2:93:10:65:d0:5e:d1
```

Respond to the OCSP request above, with the OCSP response signed by the CA itself:

```
pki --ocsp --respond --in req_ca.der --cacert cacert.pem --key cakey.pem \
        --lifetime 10 > rsp_ca.der

nonce:               5b:14:e3:cc:d5:b2:65:ec:c4:0d:c3:11:37:6a:9d:71
issuer:            "C=CH, O=strongSwan Project, CN=strongSwan Root CA"
  issuerKeyHash:     b6:76:79:95:b5:58:..:06:93:f3:39:79:19 (ok)
  issuerNameHash:    af:25:78:ce:fc:15:..:67:95:81:31:a3:4d (ok)
  serialNumber:      4f:33:21:1d:4d:fd:9b:db
  thisUpdate:        Oct 19 15:54:15 UTC 2023
  nextUpdate:        Oct 19 16:04:15 UTC 2023
  certValidation:    GOOD
issuer:            "C=CH, O=strongSwan Project, CN=strongSwan Root CA"
  issuerKeyHash:     b6:76:79:95:b5:58:..:06:93:f3:39:79:19 (ok)
  issuerNameHash:    af:25:78:ce:fc:15:..:67:95:81:31:a3:4d (ok)
  serialNumber:      68:f2:93:10:65:d0:5e:d1
  thisUpdate:        Oct 19 15:54:15 UTC 2023
  nextUpdate:        Oct 19 16:04:15 UTC 2023
  certValidation:    GOOD
trusted signer:    "C=CH, O=strongSwan Project, CN=strongSwan Root CA"
ocspResponseStatus:  successful
```

Respond to a signed OCSP request providing the complete trust chain:

```
pki --ocsp --respond --in req_signed.der --cacert cacert.pem --cacert issuer1.pem \
        --key signerKey1.pem --cert signerCert1.pem --lifetime 10 > rsp_signed.der

requestor:         "C=CH, O=strongSwan Project, CN=vpn.strongswan.org"
  using certificate "C=CH, O=strongSwan Project, CN=vpn.strongswan.org"
  using trusted intermediate ca certificate "C=CH, O=strongSwan Project, CN=strongSwan Issuing CA 1"
  using trusted ca certificate "C=CH, O=strongSwan Project, CN=strongSwan Root CA"
  reached self-signed root ca with a path length of 1
```

```
requestor is trusted
nonce:              a8:0f:29:0f:08:9c:29:c1:0d:a8:cb:b0:21:fa:e1:f7
issuer:             "C=CH, O=strongSwan Project, CN=strongSwan Issuing CA 1"
  issuerKeyHash:    5a:1b:ec:17:f0:6d:..:a2:c8:e7:6a:84:20 (ok)
  issuerNameHash:   df:1e:24:71:96:e6:..:b9:82:18:45:e7:09 (ok)
  serialNumber:     04:ff:cc:8d:36:91:cb:35:d7:c4
  thisUpdate:       Oct 19 16:30:54 UTC 2023
  nextUpdate:       Oct 19 16:40:54 UTC 2023
  certValidation:   REVOKED
  revocationTime:   Mar 26 06:41:54 UTC 2023
  revocationReason: superseded
trusted signer:     "C=CH, O=strongSwan Project, CN=OCSP signer of strongSwan Issuing CA 1"
ocspResponseStatus: successful
```

Respond to an OCSP request containing two items from different known issuers having an OCSP signer each. The issuer of the first request item determines the OCSP signer used to sign the OCSP response:

```
pki --ocsp --respond --in req.der --cacert issuer1.pem --cacert issuer2.pem \
        --key signerKey1.pem --cert signerCert1.pem \
        --key signerKey2.pem --cert signerCert2.pem \
        --lifetime 10 > rsp_trusted.der
```

```
nonce:              a1:33:aa:bc:96:60:69:76:f3:bc:9c:88:3b:07:50:47
issuer:             "C=CH, O=strongSwan Project, CN=strongSwan Issuing CA 2"
  issuerKeyHash:    72:41:ca:f9:35:87:..:d3:83:ab:d5:89:7b (ok)
  issuerNameHash:   5e:b2:b4:42:e1:a5:..:b2:c3:9a:38:4f:cd (ok)
  serialNumber:     29:ff:36:d9:9a:21:49:61:91:1d
  thisUpdate:       Oct 19 16:02:35 UTC 2023
  nextUpdate:       Oct 19 16:12:35 UTC 2023
  certValidation:   REVOKED
  revocationTime:   Sep 22 13:13:04 UTC 2023
  revocationReason: superseded
issuer:             "C=CH, O=strongSwan Project, CN=strongSwan Issuing CA 1"
  issuerKeyHash:    5a:1b:ec:17:f0:6d:..:a2:c8:e7:6a:84:20 (ok)
  issuerNameHash:   df:1e:24:71:96:e6:..:b9:82:18:45:e7:09 (ok)
  serialNumber:     10:ff:45:9a:6d:ee:4c:ec:7c:97
  thisUpdate:       Oct 19 16:02:35 UTC 2023
  nextUpdate:       Oct 19 16:12:35 UTC 2023
  certValidation:   FAILED
```

```
there are multiple known issuers
trusted signer:     "C=CH, O=strongSwan Project, CN=OCSP signer of strongSwan Issuing CA 2"
ocspResponseStatus:  successful

Repeat the OCSP response above but with a self-signed OCSP signing certificate

pki --ocsp --respond --in req.der --cacert issuer1.pem --cacert issuer2.pem \
        --key signerKey.pem --cert signerCert.pem --lifetime 10 > rsp_self_signed.der

nonce:              a1:33:aa:bc:96:60:69:76:f3:bc:9c:88:3b:07:50:47
issuer:            "C=CH, O=strongSwan Project, CN=strongSwan Issuing CA 2"
  issuerKeyHash:     72:41:ca:f9:35:87:..:d3:83:ab:d5:89:7b (ok)
  issuerNameHash:    5e:b2:b4:42:e1:a5:..:b2:c3:9a:38:4f:cd (ok)
  serialNumber:      29:ff:36:d9:9a:21:49:61:91:1d
  thisUpdate:        Oct 19 16:13:23 UTC 2023
  nextUpdate:        Oct 19 16:23:23 UTC 2023
  certValidation:    REVOKED
  revocationTime:    Sep 22 13:13:04 UTC 2023
  revocationReason:  superseded
issuer:            "C=CH, O=strongSwan Project, CN=strongSwan Issuing CA 1"
  issuerKeyHash:     5a:1b:ec:17:f0:6d:..:a2:c8:e7:6a:84:20 (ok)
  issuerNameHash:    df:1e:24:71:96:e6:..:b9:82:18:45:e7:09 (ok)
  serialNumber:      10:ff:45:9a:6d:ee:4c:ec:7c:97
  thisUpdate:        Oct 19 16:13:23 UTC 2023
  nextUpdate:        Oct 19 16:23:23 UTC 2023
  certValidation:    GOOD
there are multiple known issuers
self-signed signer: "C=CH, O=strongSwan Project, CN=strongSwan OCSP signer"
ocspResponseStatus:  successful
```

SEE ALSO

pki(1)