NAME

pki - Simple public key infrastructure (PKI) management tool

SYNOPSIS

pki*command* [*option* ...]

pki-h | **--help**

DESCRIPTION

pki is a suite of commands that allow you to manage a simple public key infrastructure (PKI).

Generate RSA and ECDSA key pairs, create PKCS#10 certificate requests containing subjectAltNames, create X.509 self-signed end-entity and root CA certificates, issue end-entity and intermediate CA certificates signed by the private key of a CA and containing subjectAltNames, CRL distribution points and URIs of OCSP servers. You can also extract raw public keys from private keys, certificate requests and certificates and compute two kinds of SHA-1-based key IDs.

The **pki** command also supports certificate enrollment via the **Simple Certificate Enrollment Protocol** (SCEP) as defined by RFC 8894, replacing the obsoleted **ipsec scepclient** tool. Additionally the **Enrollment over Secure Transport** (EST) protocol (RFC 7030) is supported, too.

The latest feature is an **Online Certificate Status Protocol** (OCSP) responder as defined by RFC 6960, interoperating with an **OpenXPKI** server by directly accessing its internal certificate datebase.

COMMANDS

-h, --help

Prints usage information and a short summary of the available commands.

-g, --gen

Generate a new private key.

-s, --self

Create a self-signed certificate.

-i, --issue

Issue a certificate using a CA certificate and key.

-c, --signcrl

Issue a CRL using a CA certificate and key.

-z, --acert
Issue an attribute certificate.

-r, --req
Create a PKCS#10 certificate request.

-7, --pkcs7
Provides PKCS#7 wrap/unwrap functions.

-k, --keyid
Calculate key identifiers of a key or certificate.

-a, --print
Print a credential (key, certificate etc.) in human readable form.

-d, --dn
Extract the subject DN of an X.509 certificate.

-p, --pub
Extract a public key from a private key or certificate.

-v, --verify
Verify a certificate using a CA certificate.

-S, --scep
Enroll an X.509 certificate with a SCEP server.

-C, --scepca
Get CA [and RA] certificate[s] from a SCEP server.

-E, --est
Enroll an X.509 certificate with an EST server.

-e, --estca
Get CA certificate[s] from an EST server.

-o, --ocsp
OCSP request parser and OCSP responder.

EXAMPLES

Generating a CA Certificate

The first step is to generate a private key using the **--gen** command. By default this generates a 2048-bit RSA key.

```
pki --gen > ca_key.der
```

This key is used to create the self-signed CA certificate, using the **--self** command. The distinguished name should be adjusted to your needs.

```
pki --self --ca --in ca_key.der \
  --dn "C=CH, O=strongSwan, CN=strongSwan CA" > ca_cert.der
```

Generating End-Entity Certificates

With the root CA certificate and key at hand end-entity certificates for clients and servers can be issued. Similarly intermediate CA certificates can be issued, which in turn can issue other certificates. To generate a certificate for a server, we start by generating a private key.

```
pki --gen > server_key.der
```

The public key will be included in the certificate so lets extract that from the private key.

```
pki --pub --in server_key.der > server_pub.der
```

The following command will use the CA certificate and private key to issue the certificate for this server. Adjust the distinguished name, subjectAltName(s) and flags as needed (check **pki --issue**(8) for more options).

```
pki --issue --in server_pub.der --cacert ca_cert.der \
  --cakey ca_key.der --dn "C=CH, O=strongSwan, CN=VPN Server" \
  --san vpn.strongswan.org --flag serverAuth > server_cert.der
```

Instead of storing the public key in a separate file, the output of **--pub** may also be piped directly into the above command.

Generating Certificate Revocation Lists (CRL)

If end-entity certificates have to be revoked, CRLs may be generated using the **--signcrl** command.

```
pki --signcrl --cacert ca_cert.der --cakey ca_key.der \
  --reason superseded --cert server_cert.der > crl.der
```

The certificate given with --cacert must be either a CA certificate or a certificate with the *crlSign* extended key usage (--flag crlSign). URIs to CRLs may be included in issued certificates with the --crl option.

SEE ALSO

pki --gen(1), **pki --self**(1), **pki --issue**(1), **pki --signcrl**(1), **pki --acert**(1), **pki --req**(1), **pki --pkcs7**(1), **pki --keyid**(1), **pki --print**(1), **pki --dn**(1), **pki --pub**(1), **pki --verify**(1), **pki --scep**(1) **pki --scepca**(1) **pki --est**(1) **pki --estca**(1) **pki --ocsp**(1)