PKI

Setup

Put `openssl.conf`:

openssl.conf

```
distinguished_name  = $ENV::DN
x509_extensions     = $ENV::EXTENSIONS

string_mask = nombstr         # This sets a mask for permitted string types
prompt      = no              # Don't ask questions

default_bits      = 4096      # Key length to use (due to req bug this is also specified
in cmd line)
default_md        = sha256    # The message digest to use

[ ca ]
default_ca    = CA_default    # The default ca section

[ CA_default ]
database      = index.txt     # The text database file to use. Mandatory. This file must
be present though initially it will be empty
serial        = serial        # A text file containing the next serial number to use in
hex

unique_subject  = no          # If the value no is given, several valid certificate
entries may have the exact same subject
email_in_dn     = no          # If you want the EMAIL field to be removed from the DN of
the certificate simply set this to 'no'
preserve        = no          # keep passed DN ordering
name_opt        = ca_default  # Subject name display option
cert_opt        = ca_default  # Certificate display option
```

```
policy          = ca_policy   # Policy section

[ ca_policy ]
countryName             = supplied
stateOrProvinceName     = supplied
localityName            = supplied
organizationName        = supplied
organizationalUnitName  = supplied
commonName              = supplied

[ ca_extensions ]
subjectKeyIdentifier    = hash
authorityKeyIdentifier  = keyid:always,issuer:always
basicConstraints        = CA:true
keyUsage                = cRLSign, keyCertSign
nameConstraints         = critical,permitted;DNS:.$ENV::FQDN

[ cert_extensions ]
subjectKeyIdentifier    = hash
authorityKeyIdentifier  = keyid,issuer:always
basicConstraints        = CA:FALSE
keyUsage                = digitalSignature, keyEncipherment
extendedKeyUsage        = serverAuth, 1.3.6.1.5.5.8.2.2
subjectAltName          = $ENV::ALT
nsCertType              = server
nsComment               = "HxD Internal Certificate"

[ ca_dn ]
countryName             = IE
stateOrProvinceName     = Dublin
localityName            = Dublin
0.organizationName      = Hexa Dust
organizationalUnitName  = Ops
commonName              = "HxD Internal Certification Authority"

[ cert_dn ]
countryName             = IE
stateOrProvinceName     = Dublin
localityName            = Dublin
```

```
0.organizationName      = Hexa Dust
organizationalUnitName  = Ops
commonName              = $ENV::FQDN
```

Create `certs` dir:

```
mkdir certs
```

Certificate request

Use `./mkreq <fqdn>` to create request file.

mkreq

```
#!/bin/sh
set -u

export DN=cert_dn
export EXTENSIONS=cert_extensions
export FQDN="$1"
export ALT=

openssl req \
    -new -newkey rsa \
    -keyout "certs/$FQDN.key" \
    -out "certs/$FQDN.csr" \
    -config openssl.conf
```

Self-sign certificate request

Request file can be self-signed using `./selfsign <fqdn>`.

selfsign

```
#!/bin/sh
set -u

export DN=cert_dn
export EXTENSIONS=cert_extensions
export FQDN="$1"
export ALT="DNS:$FQDN"

openssl req -x509 \
-days 7120 \
-nodes \
-new -newkey rsa \
-keyout "certs/$FQDN.key" -out "certs/$FQDN.crt" \
-config openssl.conf
openssl x509 -text -in "certs/$FQDN.crt"
```

CA certificate and signing

For CA signed certificates we need CA certificate and key and then use it to sign the certificate requires.

Create CA certificate and key

Use . `/mkca <domain>` to create CA for given domain. The `<domain>.crt` file can be installed in a browser or system wide as trusted CA.

mkca

```
#!/bin/sh
set -u

export DN=cert_dn
export EXTENSIONS=ca_extensions
export FQDN="$1"
```

```
export ALT=

test -f "$FQDN.key" || openssl genrsa \
  -aes256 -out "$FQDN.key" 4096
test -f "$FQDN.crt" || openssl req \
  -x509 \
  -new -nodes -key "$FQDN.key" \
  -sha256 -days 14240 \
  -out "$FQDN.crt" \
  -config openssl.conf
```

CA-sign certificate request

Use `./casign <fqdn>` to create CA-signed certificate.

casign

```
#!/bin/sh
set -u

export DN=cert_dn
export EXTENSIONS=cert_extensions
export FQDN="$1"
export CA=`echo "$FQDN" | sed -r 's/.*\.([^\.]+\.)/\1/'`
test "$CA" = "$FQDN" && export CA=`echo "$FQDN" | sed -r 's/.*\.//'`
shift
ALT="DNS:$FQDN"
for A in "$@"; do
  ALT="$ALT,DNS:$A"
done
export ALT

openssl x509 -req \
  -in "certs/$FQDN.csr" \
  -CA "$CA.crt" -CAkey "$CA.key" -CAcreateserial \
```

```
-sha256 -days 14240 \
-extensions $EXTENSIONS \
-extfile openssl.conf \
  -out "certs/$FQDN.crt"
openssl x509 -text -in "certs/$FQDN.crt"
```

Decrypt certificate key file

Use `./dec <fqdn>` to decrypt key file to `stdout`.

dec

```
#!/bin/sh
set -u
FQDN="$1"

openssl rsa -in "certs/$FQDN.key"
```

Revision #5
Created 2023-05-05 12:04:57 IST by hxd
Updated 2024-10-24 19:07:57 IST by hxd