Create the intermediate pair

An intermediate certificate authority (CA) is an entity that can sign certificates on behalf of the root CA. The root CA signs the intermediate certificate, forming a chain of trust.

The purpose of using an intermediate CA is primarily for security. The root key can be kept offline and used as infrequently as possible. If the intermediate key is compromised, the root CA can revoke the intermediate certificate and create a new intermediate cryptographic pair.

Prepare the directory

The root CA files are kept in `/root/ca`. Choose a different directory (`/root/ca/intermediate`) to store the intermediate CA files.

```
# mkdir /root/ca/intermediate
```

Create the same directory structure used for the root CA files. It's convenient to also create a `csr` directory to hold certificate signing requests.

```
# cd /root/ca/intermediate
# mkdir certs crl csr newcerts private
# chmod 700 private
# touch index.txt
# echo 1000 > serial
```

Add a `crlnumber` file to the intermediate CA directory tree. `crlnumber` is used to keep track of *certificate revocation lists*.

```
# echo 1000 > /root/ca/intermediate/crlnumber
```

Copy the intermediate CA configuration file from the *Appendix* to `/root/ca/intermediate/openssl.cnf`. Five options have been changed compared to the root CA configuration file:

```
[ CA_default ]
dir             = /root/ca/intermediate
private_key     = $dir/private/intermediate.key.pem
certificate     = $dir/certs/intermediate.cert.pem
crl             = $dir/crl/intermediate.crl.pem
policy          = policy_loose
```

Create the intermediate key

Create the intermediate key (`intermediate.key.pem`). Encrypt the intermediate key with AES 256-bit encryption and a strong password.

```
# cd /root/ca
# openssl genrsa -aes256 \
      -out intermediate/private/intermediate.key.pem 4096

Enter pass phrase for intermediate.key.pem: secretpassword
Verifying - Enter pass phrase for intermediate.key.pem: secretpassword

# chmod 400 intermediate/private/intermediate.key.pem
```

Create the intermediate certificate

Use the intermediate key to create a certificate signing request (CSR). The details should generally match the root CA. The **Common Name**, however, must be different.

> **⚠ Warning**
>
> Make sure you specify the intermediate CA configuration file (`intermediate/openssl.cnf`).

```
# cd /root/ca
# openssl req -config intermediate/openssl.cnf -new -sha256 \
      -key intermediate/private/intermediate.key.pem \
      -out intermediate/csr/intermediate.csr.pem

Enter pass phrase for intermediate.key.pem: secretpassword
You are about to be asked to enter information that will be incorporat
into your certificate request.
-----
Country Name (2 letter code) [XX]:GB
```

```
State or Province Name []:England
Locality Name []:
Organization Name []:Alice Ltd
Organizational Unit Name []:Alice Ltd Certificate Authority
Common Name []:Alice Ltd Intermediate CA
Email Address []:
```

To create an intermediate certificate, use the root CA with the `v3_intermediate_ca` extension to sign the intermediate CSR. The intermediate certificate should be valid for a shorter period than the root certificate. Ten years would be reasonable.

> **⚠ Warning**
>
> This time, specify the root CA configuration file (`/root/ca/openssl.cnf`).

```
# cd /root/ca
# openssl ca -config openssl.cnf -extensions v3_intermediate_ca \
      -days 3650 -notext -md sha256 \
      -in intermediate/csr/intermediate.csr.pem \
      -out intermediate/certs/intermediate.cert.pem

Enter pass phrase for ca.key.pem: secretpassword
Sign the certificate? [y/n]: y

# chmod 444 intermediate/certs/intermediate.cert.pem
```

The `index.txt` file is where the OpenSSL `ca` tool stores the certificate database. Do not delete or edit this file by hand. It should now contain a line that refers to the intermediate certificate.

```
V 250408122707Z 1000 unknown ... /CN=Alice Ltd Intermediate CA
```

Verify the intermediate certificate

As we did for the root certificate, check that the details of the intermediate certificate are correct.

```
# openssl x509 -noout -text \
      -in intermediate/certs/intermediate.cert.pem
```

Verify the intermediate certificate against the root certificate. An `OK` indicates that the chain of trust is intact.

```
# openssl verify -CAfile certs/ca.cert.pem \
      intermediate/certs/intermediate.cert.pem

intermediate.cert.pem: OK
```

Create the certificate chain file

When an application (eg, a web browser) tries to verify a certificate signed by the intermediate CA, it must also verify the intermediate certificate against the root certificate. To complete the chain of trust, create a CA certificate chain to present to the application.

To create the CA certificate chain, concatenate the intermediate and root certificates together. We will use this file later to verify certificates signed by the intermediate CA.

```
# cat intermediate/certs/intermediate.cert.pem \
      certs/ca.cert.pem > intermediate/certs/ca-chain.cert.pem
# chmod 444 intermediate/certs/ca-chain.cert.pem
```

> **Note**
>
> Our certificate chain file must include the root certificate because no client application knows about it yet. A better option, particularly if you're administrating an intranet, is to install your root certificate on every client that needs to connect. In that case, the chain file need only contain your intermediate certificate.

Comments

← Previous

Next →

Version 1.0.4 — Last updated on 2015-12-09.