moonshot-webp XML Format

The basic file format

Moonshot ships with a tool, moonshot-webp, to securely and correctly provision credentials onto clients. A basic template is provided at `/usr/share/moonshot-ui/default-identity.msht`. The format for credential files is simple XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<identities>
  <identity>
    <display-name>[i.e. John Smith from Camford University]</display-name>
    <user>[i.e. johnsmith]</user>
    <password>[i.e. correct-horse-battery-staple]</password>
    <realm>[i.e. camford.ac.uk]</realm>
    <services>[optional, see Services]</services>
    <selection-rules>[optional, see Selection Rules]</selection-rules>
    <trust-anchor>
      <!-- Either ca-cert and subject, or ca-cert and subject-alt -->
      <ca-cert>[Base64-encoded representation of the APC/IdP's CA root certificate, see Trust Anchors]</ca-cert>
      <subject>[CN value of the APC/IdP's server certificate, see Trust Anchors]</subject>
      <subject-alt>[The DNS/FQDN or IP value in the X509v3 extension of the APC/IdP's server certificate, see
Trust Anchors]</subject-alt>
      <!-- Or, alternatively, server-cert only -->
      <server-cert>[SHA-256 hash of the APC/IdP's server certificate, see Trust Anchors]</server-cert>
    </trust-anchor>
  </identity>
</identities>
```

Trust Anchors

Inclusion of the trust anchor is vital - without it credentials may be exposed to malicious resource providers. This credential format is also used to secure communication between RP's, IdP's and trust routers.

You must use either the Certificate Authority (CA) root certificate or the server certificate as trust anchor.

CA Root Certificate

> **Certificate Expiry**
>
> We recommend that a CA certificate is generated with a long expiry time (in years or decades), and that it is kept safe for subsequent server and user certificate generation cycles.

To use the CA root certificate as a trust anchor, you must populate the one or more of the following tags in the `<trust-anchor>` section:

- `<ca-cert>`: The value of this tag is either a Base64-encoded version of the CA certificate in DER form, or the contents of `ca.pem`, excluding the BEGIN and END lines.
  This value is **always** required.
- `<subject>`: The value of this tag is the CN value of the DN in the text representation of the **server** certificate.
  This value is required when `<subject-alt>` is not specified.
- `<subject-alt>`: The value of this should be the DNS name, FQDN or the IP address information in the X509v3 information of the **server** certificate.
  This value is required when `<subject>` is not specified.

To retrieve either the `subject` or `subject-alt` information, dump the server certificate's text information. Use OpenSSL as follows:

```
openssl x509 -noout -text -in server.pem
```

A sample output of the command would yield:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=FR, ST=Radius, L=Somewhere, O=Example Inc./emailAddress=admin@example.org, CN=Example
Certificate Authority
        Validity
            Not Before: Nov 14 16:22:19 2017 GMT
            Not After : Jan 13 16:22:19 2018 GMT
        Subject: C=FR, ST=Radius, O=Example Inc., CN=Example Server Certificate/emailAddress=admin@example.org
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus: [trimmed]
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Alternative Name:
                DNS:example.org, DNS:radius.example.org
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
            X509v3 CRL Distribution Points:
                Full Name:
                  URI:http://www.example.com/example_ca.crl
    Signature Algorithm: sha256WithRSAEncryption
         [trimmed]
```

The `subject` value to specify would be:

```
C=FR/ST=Radius/O=Example Inc./CN=Example Server Certificate/emailAddress=admin@example.org
```

The `subject-alt` value to specify would be the value of the `X509v3 Subject Alternative Name` value in the text:

```
DNS:example.org, DNS:radius.example.org
```

Server Certificate

> **Certificate Expiry**
>
> When your server certificate expires, the credentials must be updated and re-provisioned with the new fingerprint.

To use the server certificate itself as a trust anchor, you must populate the <server-cert> tag in the <trust-anchor> section with the SHA-256 fingerprint of the server certificate.

To obtain this fingerprint, run OpenSSL as follows and strip the colon symbols from the resulting value:

```
openssl x509 -noout -fingerprint -sha256 -in server.pem
```

A sample output of the command would yield:

```
SHA256 Fingerprint=24:98:D7:16:AB:78:23:E4:17:B8:D5:C8:71:3D:F1:6C:FC:E2:15:83:62:91:0D:5A:7D:7D:DE:55:63:E4:CF:
2D
```

The `server-cert` value to specify would be:

```
2498D716AB7823E417B8D5C8713DF16CFCE2158362910D5A7D7DDE5563E4CF2D
```

Services

The optional `services` section is used to determine which services the credential will be automatically used for - each service will be contained in its own tag. For use with a trust router, it is better to use the `selection-rules` section instead.

services

```
<services>
  <service>xmpp/jabber.project-moonshot.org</service>
  <service>email/project-moonshot.org</service>
  <service>host/ssh.project-moonshot.org</service>
</services>
```

Selection Rules

The optional `selection-rules` section is used to restrict which services the credential will be automatically used for - for use with a trust router identity, the service type is "trustidentity" for all services. Wildcards are acceptable.

selection-rules

```
<selection-rules>
   <rule>
     <pattern>trustidentity/*</pattern>
     <always-confirm>false</always-confirm>
   </rule>
</selection-rules>
```