Certificate and Keys Tech Notes

Generate a Self-Signed Root CA

Generate a self-signed root certificate authority (CA).

```
openssl genrsa -out myca.key 2048
# password protect key: openssl genrsa -out myca.key -des3 2048
openssl req -x509 -new -key myca.key -sha384 -days 1825 -out myca.crt \
  -subj "/C=US/ST=CA/L=Santa
Clara/O=MyOrg/OU=SecurityOU/CN=rootca.myorg.com/emailAddress=rootca@myorg.com"
```

This root CA must be installed as a trusted root CA on the users (client) machines

Note Generating a self-signed certificate using **macOS** will not produce a certificate suitable for forward and reverse proxy scenarios. The certificate must have the *Is CA* option set to *True* and the certificate generated using macOS does not. It is recommended that the self-signed certificate be generated from within the Multicloud Defense UI (Certificates > Create > Generate) or using **Linux**.

Generate a Certificate Signed by your Self-Signed Root CA

Generate a certificate signed by the above root certificate authority (CA). You can use this certificate in applications.

```
openssl genrsa -out app1.key 2048
# password protect key: openssl genrsa -out -des3 app1.key 2048
openssl req -new -key app1.key -out app1.csr \
  -subj "/C=US/ST=CA/L=Santa
Clara/O=MyOrg/OU=AppOU/CN=app1.myorg.com/emailAddress=app1@myorg.com"
openssl x509 -req -in app1.csr -CA myca.crt -CAkey myca.key -out app1.crt -sha384\
  -days 365 -CAcreateserial -extensions SAN \
  -extfile <(printf "
[SAN]\nbasicConstraints=CA:false\nsubjectAltName=DNS:app1.myorg.com,DNS:app1-
1.myorg.com,IP:192.168.10.21,IP:192.168.10.22")
```

Generate an Intermediate CA Signed by Your Root CA

If you don't want to use the root certificate authority (CA) to sign app certs, then create an intermediate CA signed by the root CA, then sign the app certs using the intermediate CA. Append the intermediate cert to the app cert. At this point the app crt has 2 certs (as a chain).

```
openssl genrsa -out interca.key 2048
# password protect key: openssl genrsa -out -des3 interca.key 2048
openssl req -new -key interca.key -out interca.csr \
  -subj "/C=US/ST=CA/L=Santa
Clara/O=MyOrg/OU=InterSecurityOU/CN=interca1.myorg.com/emailAddress=interca1@myorg.com"
openssl x509 -req -in interca.csr -CA myca.crt -CAkey myca.key -out interca.crt - sha384 \
  -days 365 -CAcreateserial -extensions SAN \
  -extfile <(printf "[SAN]\nbasicConstraints=CA:true")
```

App Certificate signed using the Intermediate CA

```
openssl genrsa -out app1.key 2048
# password protect key: openssl genrsa -out -des3 app1.key 2048
openssl req -new -key app1.key -out app1.csr \
  -subj "/C=US/ST=CA/L=Santa
Clara/O=MyOrg/OU=AppOU/CN=app1.myorg.com/emailAddress=app1@myorg.com"
openssl x509 -req -in app1.csr -CA interca.crt -CAkey interca.key -out app1.crt - sha384 \
  -days 365 -CAcreateserial -extensions SAN \
  -extfile <(printf "
[SAN]\nbasicConstraints=CA:false\nsubjectAltName=DNS:app1.myorg.com,DNS:app1-
1.myorg.com,IP:192.168.10.21,IP:192.168.10.22")
```

Append the files app1.crt and interca.crt to create a combined certificate, and use it in your application. The root CA must be installed as a trusted root CA on your client machines.

Install Root CA as Trusted CA on the Hosts

OS	**Command**
Ubuntu	Copy crt file to `/usr/local/share/ca-certificates`. Run the command `sudo update-ca- certificates`.
CentOS	Copy crt file to `/etc/pki/ca-trust/source/anchors`. Run the command `sudo update-ca- trust extract`.
Windows	Double click the file and add the cert to Trusted Root, or run the command `certutil -addstore "Root" <crt-file>`.