証明書とキーに関するテクニカルノート

自己署名ルート CA の生成

自己署名ルート認証局（CA）を生成します。

```
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"
```

このルートCAは、信頼できるルートCAとしてユーザー（クライアント）マシンにインストールする必要があります。

（注） **MacOS** を使用して自己署名証明書を生成しても、転送およびリバースプロキシシナリオに使用できる適切な証明書は生成されません。証明書では [CAである（Is CA）] オプションが [True] に設定されている必要がありますが、MacOS を使用して生成された証明書ではこのオプションがありません。自己署名証明書は、Multicloud Defense の UI（[証明書（Certificates）] > [作成（Create）] > [生成（Generate）]）から生成するか、または **Linux** を使用して生成することをお勧めします。

自己署名ルート CA によって署名された証明書の生成

上記のルート認証局（CA）によって署名された証明書を生成します。この証明書は、アプリケーションで使用できます。

```
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")
```

ルート CA によって署名された中間 CA の生成

ルート認証局（CA）をアプリケーション証明書への署名に使用しない場合は、ルート CA によって署名された中間 CA を作成し、中間 CA を使用してアプリケーション証明書に署名します。アプリケーション証明書に中間証明書を追加します。この時点で、アプリケーション証明書には 2 つの証明書（チェーンとして）があります。

```
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")
```

中間 CA を使用して署名されたアプリケーション証明書

```
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")
```

app1.crt および interca.crt ファイルを追加して結合証明書を作成し、アプリケーションで結合証明書を使用します。ルート CAは、信頼できるルート CA としてクライアントマシンにインストールする必要があります。

ホストへの信頼できる CA としてのルート CA のインストール

OS	コマンド
Ubuntu	crt ファイルを `/usr/local/share/ca-certificates` にコピーし、`sudo update-ca- certificates` コマンドを実行します。
CentOS	crt ファイルを `/etc/pki/ca-trust/source/anchors` にコピーし、`sudo update-ca- trust extract` コマンドを実行します。
Windows	ファイルをダブルクリックして信頼されたルートに証明書を追加するか、コマンド `certutil -addstore "Root" <crt-file>` を実行します。

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。