Front-End Web & Mobile

Use Your Own Certificate with AWS IoT

by Olawale Oladehin | on 15 APR 2016 | Permalink | Comments | Share

Introduction

Earlier this week, AWS IoT released support for customers who need to use their own device certificates signed by their preferred Certificate Authority (CA). This is in addition to the support for AWS IoT generated certificates. The CA certificate is used to sign and issue device certificates, while the device certificates are used to connect a client to AWS IoT. Certificates provide strong client side authentication for constrained IoT devices. During TLS handshake, the server authenticates the client using the X.509 certificate presented by the client.

With this feature, customers with existing devices in the field or new devices with certificates signed by a CA other than AWS IoT can seamlessly authenticate with AWS IoT. It also provides manufacturers the ability to provision device certificates using their current processes and then register those device certificates to AWS IoT. For example, if a customer's manufacturing lines lack internet connectivity, they can provision their devices offline with their own CA issued certificates and later register them with AWS IoT.

This blog will walk you through an end-to-end process of setting up a client that uses a device certificate signed by your own CA. First, you will generate a CA certificate that will be used to sign your device certificate. Next, you will register the CA certificate and then register the device certificates. After these steps, your device certificate will be ready to connect AWS IoT service. This blog will also walk you through additional scenarios such as using multiple CA certificates and revoking CA certificates.

This blog assumes that you are familiar with AWS IoT and the process of creating an AWS IoT certificate. Refer to the AWS IoT developer documentation for more information on how to use AWS IoT generated certificates or to learn more about authentication in AWS IoT.

Registering Your First CA Certificate

If you are a manufacturer, you would have purchased CA certificates from vendors such as Symantec, Verisign, etc., In order to use your own X.509 certificates that have been signed by your CA certificate, AWS IoT first needs to verify that you not only own the CA certificate but that you also have access to the private key for that certificate. The process of validating ownership of a CA certificate is done through a challenge and response workflow with AWS IoT.

Let's begin by creating your first sample CA certificate using openssl in a terminal. In reality, you would have the signing certificates issued by your CA vendor in the place of this sample CA. This sample CA certificate is used later in the walkthrough to sign a device certificate that you register with AWS IoT:

```
$ openssl genrsa -out sampleCACertificateOne.key 2048 $ openssl req -x509 –new –nodes
```

Now that you've created a sample CA certificate, you will register it with AWS IoT. When registering a CA certificate with AWS IoT, you follow a workflow to verify that you have access to both the CA certificate and the private key associated with the CA certificate. To verify ownership of the private key, you generate a verification certificate using the CA certificate, the private key, and a registration code that you generate from AWS IoT.

The registration workflow first requires retrieving a registration code from AWS IoT. You retrieve the registratic code using the AWS CLI or from the AWS IoT Console in the "Register Certificate" section. Use the AWS CLI to generate a registration code with the following command:

```
$ aws iot get-registration-code
```

The AWS CLI command, get-registration-code, will return a randomly generated unique registration code that is bound to your AWS account. This registration code is long-lived and does not expire until you delete it. To illustrate using the registration code, create a new CSR:

```
$ openssl genrsa –out privateKeyVerificationOne.key 2048 $ openssl req –new –key priva
```

During the CSR process, you will be prompted for information. Enter the registration code from AWS IoT into the Common Name field of the verification certificate:

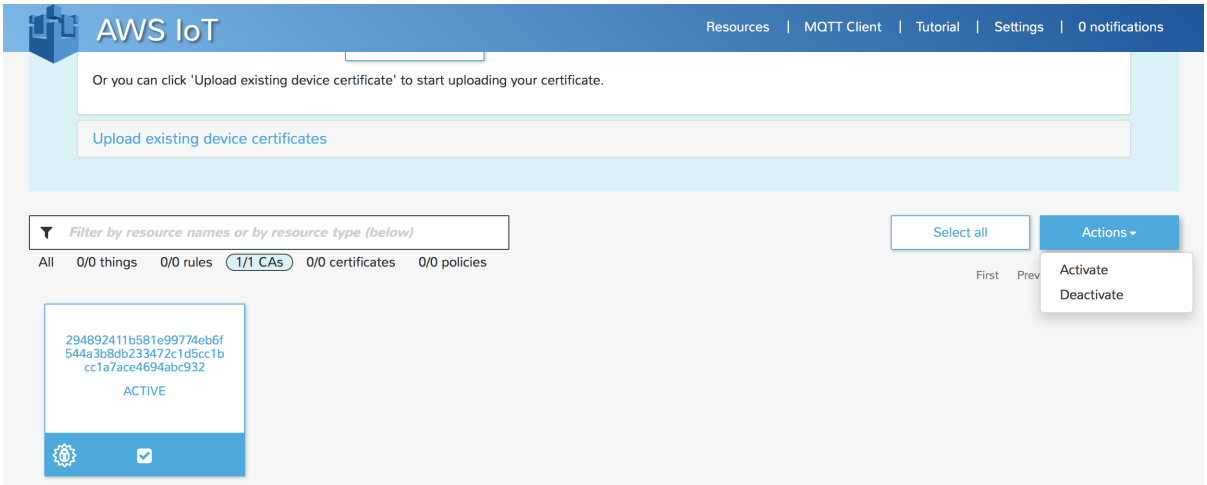
```
... Organization Name (eg, company) []: Organizational Unit Name (eg, section) Common
```

The registration code validates that the generated verification certificate was created specifically for registering the CA certificate with AWS IoT, and that the verification certificate is not a previously issued certificate. Now that you have a CSR that includes the registration code from AWS IoT, use your first sample CA certificate and the CSR to create a new certificate:

```
$ openssl x509 –req –in privateKeyVerificationOne.csr –CA sampleCACertificateOne.pem -
```

When you register your CA certificate to AWS IoT, the combination of the registration code, verification certificate signed with the CA private key, and the CA certificate are used to verify ownership of the CA private key. Next, log into the AWS IoT console and select "Use my certificate", then select "Register your CA certificate", and upload your sample CA certificate and verification certificate:

Once your CA certificate has been uploaded to AWS IoT, select the CA certificate from the console and select the "Actions" option to activate the CA certificate:

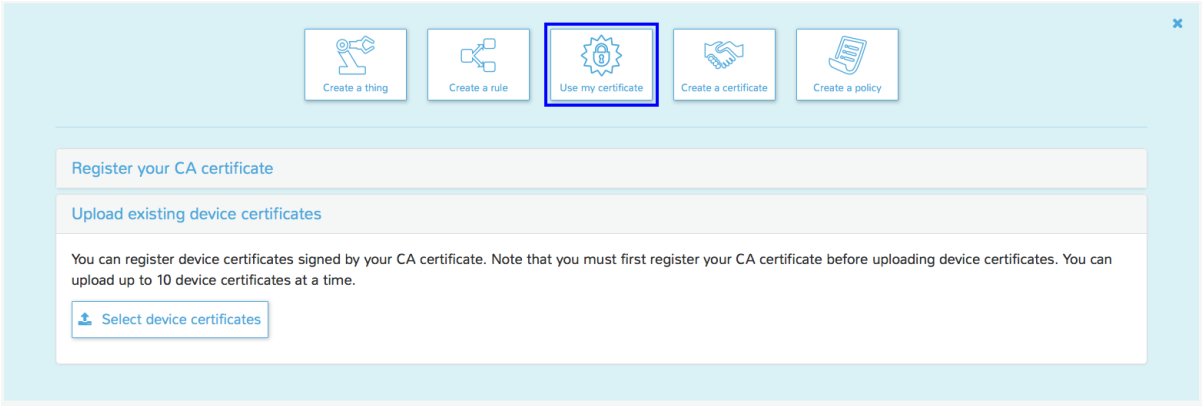


Registering A Device Certificate Signed by Your CA Certificate

Now that you've created, registered, and activated a sample CA certificate, let's use the CA certificate to create a new device certificate and upload the device certificate into AWS IoT. Enter the following commands in your terminal to create a device certificate:

```
$ openssl genrsa –out deviceCertOne.key 2048 $ openssl req –new –key deviceCertOne.key
```

You can upload the device certificate through the AWS CLI or the AWS Console. In the AWS Console, navigate to AWS IoT, and select "Use my certificate" then "Upload existing device certificates" in order to upload your device certificate into AWS IoT.



After uploading the device certificate, activate the device certificate in the AWS console so you can begin using the device certificate to communicate over MQTT with AWS IoT:

Once a device certificate is registered with AWS IoT, you can interact with it similarly to a device certificate generated by AWS IoT. You can attach policies and associate things to your device certificate. In addition, you can manage the lifecycle of that device certificate in AWS IoT such as deactivating, revoking or activating the certificate in your AWS account.

Scenario 1: Working with Multiple CA Certificates

Using AWS IoT, you can register one or more CA certificates to your AWS account. You can register, activate, deactivate, or delete CA certificates associated to your AWS account without affecting the device certificates that you have previously registered with AWS IoT. Now that you have completed the workflow of registering your first CA certificate and device certificate, let's create a second CA certificate and go into more detail about how you can manage multiple CA certificates in your AWS account. For the second CA Certificate, use openssl to generate the certificate:

```
$ openssl genrsa -out sampleCACertificateTwo.key 2048 $ openssl req -x509 –new –nodes
```

For this CA Certificate, you will use the same AWS registration code that you generated for the first CA Certificate:

```
$ aws iot get-registration-code
```

Similar to the first CA Certificate, use the registration code as the Common Name field of a second verification CSR:

```
$ openssl genrsa –out privateKeyVerificationTwo.key 2048 $ openssl req –new –key priva
```

You use the second sample CA certificate and the second verification CSR to create a new verification certificate:

```
$ openssl x509 –req –in privateKeyVerificationTwo.csr –CA sampleCACertificateTwo.pem -
```

During the provisioning process, customers may choose to preregister several CA certificates with AWS IoT at the same time. For example, a customer may want to provision multiple batches of devices using intermediate CA certificates and actively rotate their intermediate CA certificate every few months. In this use case, a customer may already have a number of intermediate CA certificates but not all of them will be active immediately. With AWS IoT, customers can register their CA certificates and choose to set the CA certificate as inactive or active when needed.

For your second CA certificate, register the CA certificate with an inactive state using the AWS CLI:

```
$ aws iot register-ca-certificate --ca-certificate file://sampleCACertificateTwo.pem -
```

To illustrate the behavior of attempting to register a device certificate using an inactive CA certificate, create a new device certificate using the inactive CA certificate and attempt to register the device certificate with AWS IoT:

```
$ openssl genrsa –out deviceCertTwo.key 2048 $ openssl req –new –key deviceCertTwo.key
```

When you attempt to register a certificate that has an inactive CA certificate, AWS IoT returns an exception informing you that the CA certificate associated to the device certificate is not active.

Scenario 2: Deactivating CA Certificates

During device certificate registration, AWS will check if the associated CA certificate is active for a device certificate. If the associated CA certificate is inactive, AWS IoT does not allow the device certificate to be registered. This feature provides customers the flexibility to manage the lifecycle of device certificates that have yet to be registered with AWS IoT through the CA certificate.

To demonstrate this scenario, use the AWS CLI to list all of the CA certificates:

```
$ aws iot list-ca-certificates
```

Then, copy the certificate ID for the first sample CA certificate and use the certificate ID as input to the AWS CL command, update-ca-certificate, in order to change the first sample CA certificate to inactive:

```
$ aws iot update-ca-certificate --certificate-id [CERTIFICATE_ID] --new-status INACTIV
```

After making this change, you will have two inactive CA certificates linked to your AWS account. You do not have any device certificates associated with the second CA certificate. But with the first CA certificate, you do have a previously registered and activated device certificate. Let's view the state of the previously registered device certificate. For this step, use the AWS CLI to list all device certificates that are associated with the deactivated CA certificate:

```
$ aws iot list-certificates-by-ca --ca-certificate-id [CERTIFICATE_ID]
```

In the output of the above AWS CLI command, you will see that even though you have deactivated your CA certificate, the associated device certificate is still active. In AWS IoT, the CA certificate is used only at registration time of the device certificate. Once a device certificate has been registered with AWS IoT, the lifecycle of that device certificate is independent from the changes made to the associated CA certificate. This feature gives customers the ability to manage the workflows of device certificates and CA certificates independently, and create flexibility to decide how to operate the lifecycle of their certificates.

You just completed a walkthrough of using your own device certificate with AWS IoT, working with multiple CA certificates and revoking CA certificates. We hope these steps were useful and enables you to provision and register your device certificates with AWS IoT. Try it out and let us know your feedback.