Generating a CA-signed certificate

Snapdrive for Unix

Ivana Devine
January 25, 2021

Table of Contents

Generating a CA-signed certificate

The SnapDrive for UNIX daemon service requires that you generate a CA-signed certificate for successful daemon communication. You must provide the CA-signed certificate at the path specified in the `snapdrive.conf` file.

- You must be logged in as a root user.
- You must have set the following parameters in the `snapdrive.conf` file to use HTTPS for communication:
  - use-https-to-sdu-daemon=on
  - contact-https-port-sdu-daemon=4095
  - sdu-daemon-certificate-path=`/opt/NetApp/snapdrive/snapdrive.pem`

Steps

1. Generate a new unencrypted RSA private key in a pem format:

   `$ openssl genrsa -out privkey.pem 1024`

   ```
   Generating RSA private key, 1024 bit long modulus
    .....................++++++ ....................................++++++
   e is 65537 (0x10001)
   ```

2. Configure `/etc/ssl/openssl.cnf` to create the CA private key and the certificate vi `/etc/ssl/openssl.cnf`.
3. Create an unsigned certificate using your RSA private key:

   `$ openssl req -new -x509 -key privkey.pem -out cert.pem`

   ```
   You are about to be asked to enter information that will be
   incorporated into your certificate request.
   What you are about to enter is what is called a Distinguished Name or
   a DN.
   There are quite a few fields but you can leave some blank For some
   fields there will be a default value, If you enter '.', the field
   will be left blank.
   -----
   Country Name (2 letter code) [XX]:NY
   State or Province Name (full name) []:Nebraska Locality Name (eg,
   city) [Default City]:Omaha Organization Name (eg, company) [Default
   Company Ltd]:abc.com Organizational Unit Name (eg, section) []:
   Common Name (eg, your name or your server's hostname) []:localhost
   Email Address []:abc@example.org
   ```

4. Use your private key and your certificate to create a CSR:

   cat cert.pem privkey.pem | openssl x509 -x509toreq -signkey privkey.pem -out certreq.csr

   ```
   Getting request Private Key Generating certificate request
   ```

5. Sign the certificate with the CA private key by using the CSR that you have just created:

   $ openssl ca -in certreq.csr -out newcert.pem

   ```
   Using configuration from /etc/pki/tls/openssl.cnf Check that the
   request matches the signature Signature ok Certificate Details:
            Serial Number: 4096 (0x1000)
            Validity
               Not Before: May 17 06:02:51 2015 GMT
                Not After : May 16 06:02:51 2016 GMT
                Subject:
                countryName               = NY
                stateOrProvinceName       = Nebraska
                organizationName          = abc.com
                commonName                = localhost
                emailAddress              = abc@example.org
                X509v3 extensions:
                X509v3 Basic Constraints:
                    CA:FALSE
                X509v3 Key Usage:
                    Digital Signature, Non Repudiation, Key Encipherment
                Netscape Comment:
                    OpenSSL Generated Certificate
                X509v3 Subject Key Identifier:

   FB:B0:F6:A0:9B:F2:C2:BC:50:BF:45:B2:9D:DB:AA:3B:C5:07:5B:7F
                X509v3 Authority Key Identifier:

    keyid:FB:B0:F6:A0:9B:F2:C2:BC:50:BF:45:B2:9D:DB:AA:3B:C5:07:5B:7F

    Certificate is to be certified until May 16 06:02:51 2016 GMT (365
    days) Sign the certificate? [y/n]:y


    1 out of 1 certificate requests certified, commit? [y/n]y Write out
    database with 1 new entries Data Base Updated
   ```

6. Install the signed certificate and the private key to be used by an SSL server.

```
The newcert.pem is the certificate signed by your local CA that you can
then use in an
ssl server:
( openssl x509 -in newcert.pem; cat privkey.pem ) > server.pem
ln -s server.pem `openssl x509 -hash -noout -in server.pem`.0 # dot-zero
( server.pem refers to location of https server certificate)
```