Generate and Sign Certificates

This chapter covers the following sections:

Self-sign certificates for Cisco Advanced Web Security Reporting application

This topic provides basic examples for creating the self-signed certificates in the command line using the version of OpenSSL included with Cisco Advanced Web Security Reporting application.

Since self-signed certificates are signed by your organization, they are not contained in browser certificate stores. As a result, web browsers consider self-signed certificates “untrusted”. This produces a warning page to users and may even prevent access for the user.

Self-signed certificates are best for browser to Cisco Advanced Web Security Reporting application communication that happens within an organization or between known entities where you can add your own CA to all browser stores that will contact Cisco Advanced Web Security Reporting application. For any other scenario, CA-signed certificates are recommended. See Get certificates signed by a third-party for Cisco Advanced Web Security Reporting application for more information.

Before You Begin

In this discussion, `$AWSR_HOME` refers to the AWSR Enterprise installation directory. We recommend that you follow this convention, but if you do not, you should replace `$AWSR_HOME` with your installation directory when using these examples.

For Windows, you might need to set this variable at the command line or in the Environment tab in the System Properties dialog. Default home directories depend on your platform:

- For Windows, the AWSR Enterprise directory is at `C:\Program Files\Cisco\` by default.
- For most Linux platforms, the default installation directory is at `/opt/`.

Generate a new root certificate to be your Certificate Authority

Step 1 Create a new directory to host your certificates and keys. For this example we will use `$AWSR_HOME/etc/auth/mycerts`.

Step 2 Generate a new RSA private key. Cisco Advanced Web Security Reporting application supports 2048 bit keys, but you can specify larger keys if they are supported by your browser.

On Linux:

```
$AWSR_HOME/bin/splunk cmd openssl genrsa -des3 -out myCAPrivateKey.key 2048
```

On Windows:

Note that in Windows you may need to append the location of the openssl.cnf file:

```
$AWSR_HOME\bin\splunk cmd openssl genrsa -des3 -out myCAPrivateKey.key 2048
```

Cisco Advanced Web Security Reporting application supports 2048 bit keys, but you can specify larger keys if they are supported by your browser.

Step 3 When prompted, create a password.

The private key `myCAPrivateKey.key` appears in your directory. This is your root certificate private key.

Step 4 Generate a certificate signing request using the root certificate private key `myCAPrivateKey.key`:

On Linux:

```
$AWSR_HOME/bin/splunk cmd openssl req -new -key myCAPrivateKey.key -out myCACertificate.csr
```

On Windows:

```
$AWSR_HOME\bin\splunk cmd openssl req -new -key myCAPrivateKey.key -out myCACertificate.csr
```

Step 5 Provide the password to the private key `myCAPrivateKey.key`.

A new CSR myCACertificate.csr appears in your directory.

Step 6 Use the CSR to generate a new root certificate and sign it with your private key:

On Linux:

```
$AWSR_HOME/bin/splunk cmd openssl x509 -req -in myCACertificate.csr -signkey myCAPrivateKey.key -out
myCACertificate.pem -days 3650
```

On Windows:

```
$AWSR_HOME\bin\splunk cmd openssl x509 -req -in myCACertificate.csr -signkey myCAPrivateKey.key -out
myCACertificate.pem -days 3650
```

Step 7 When prompted, provide for the password to the private key `myCAPrivateKey.key`.

A new certificate `myCACertificate.pem` appears in your directory. This is your public certificate.

Create a new private key for Cisco Advanced Web Security Reporting application

Step 1 Generate a new private key:

On Linux:

```
$AWSR_HOME/bin/splunk cmd openssl genrsa -des3 -out myAWSRWebPrivateKey.key 2048
```

On Windows:

```
$AWSR_HOME\bin\splunk cmd openssl genrsa -des3 -out myAWSRWebPrivateKey.key 2048 -config
```

Step 2 When prompted, create a password.

A new key, `myAWSRWebPrivateKey.key` appears in your directory.

Step 3 Remove the password from your key. (Cisco Advanced Web Security Reporting application does not support password-protected private keys.)

On Linux:

```
$AWSR_HOME/bin/splunk cmd openssl rsa -in myAWSRWebPrivateKey.key -out myAWSRWebPrivateKey.key
```

On Windows:

```
$AWSR_HOME\bin\splunk cmd openssl rsa -in myAWSRWebPrivateKey.key -out myAWSRWebPrivateKey.key
```

You can verify that your password was removed with the following command:

On Linux:

```
$AWSR_HOME/bin/splunk cmd openssl rsa -in myAWSRWebPrivateKey.key -text
```

On Windows:

```
$AWSR_HOME\bin\splunk cmd openssl rsa -in myAWSRWebPrivateKey.key -text
```

You should be able to read the contents of your certificate without providing a password.

Create and sign a server certificate

Step 1 Create a new certificate signature request using your private key `myAWSRWebPrivateKey.key`:

On Linux:

```
$AWSR_HOME/bin/splunk cmd openssl req -new -key myAWSRWebPrivateKey.key -out myAWSRWebCert.csr
```

On Windows:

```
$AWSR_HOME\bin\splunk cmd openssl req -new -key myAWSRWebPrivateKey.key -out myAWSRWebCert.csr
```

The CSR `myAWSRWebCert.csr` appears in your directory.

Step 2 Self-sign the CSR with the root certificate private key `myCAPrivateKey.key`:

On Linux:

```
$AWSR_HOME/bin/splunk cmd openssl x509 -req -in myAWSRWebCert.csr -CA myCACertificate.pem -CAkey
myCAPrivateKey.key -CAcreateserial -out myAWSRWebCert.pem -days 1095
```

On Windows:

```
$AWSR_HOME\bin\splunk cmd openssl x509 -req -in myAWSRWebCert.csr -CA myCACertificate.pem -CAkey
myCAPrivateKey.key -CAcreateserial -out myAWSRWebCert.pem -days 1095
```

Step 3 When prompted, provide the password to the root certificate private key `myCAPrivateKey.key`.

The certificate `myAWSRWebCert.pem` is added to your directory. This is your server certificate.

Create a single PEM file

Combine your server certificate and public certificates, in that order, into a single PEM file.

Here's an example of how to do this in Linux:

```
# cat myAWSRWebCert.pem myCACertificate.pem > myAWSRWebCertificate.pem
```

Here's an example in Windows:

```
# type myAWSRWebCert.pem myCACertificate.pem > myAWSRWebCertificate.pem
```

Set up certificate chains

To use multiple certificates, append the intermediate certificate to the end of the server's certificate file in the following order:

```
<div class=samplecode
[ server certificate]
[ intermediate certificate]
[ root certificate (if required) ]
</div>
```

So for example, a certificate chain might look like this:

```
-----BEGIN CERTIFICATE-----
... (certificate for your server)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the intermediate certificate)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the root certificate for the CA)...
-----END CERTIFICATE-----
```

Get certificates signed by a third-party for Cisco Advanced Web Security Reporting application

This topic provides basic examples for creating the third-party signed certificates necessary to configure Cisco Advanced Web Security Reporting application for SSL authentication and encryption.

Create a new private key for Cisco Advanced Web Security Reporting application

Step 1 Create a new directory to host your own certificates and keys. In this example we will use `$AWSR_HOME/etc/auth/mycerts`.

We recommend that you place your new certificates in a different directory than `$AWSR_HOME/etc/auth/splunkweb` so that you don't overwrite the existing certificates. This ensures that you can use the certificates that ship with AWSR for other AWSR components as necessary.

Step 2 Generate a new private key. CISCO Advanced Web Security Reporting application supports 2048-bit keys or larger.

On Linux:

```
$AWSR_HOME/bin/splunk cmd openssl genrsa -des3 -out myAWSRWebPrivateKey.key 2048
```

On Windows:

```
$AWSR_HOME\bin\splunk cmd openssl genrsa -des3 -out myAWSRWebPrivateKey.key 2048
```

Step 3 Create a password when prompted to enter the passphrase for the original key.

A new private key `myAWSRWebPrivateKey.key` is added to your directory. You can use this key to sign your CSR.

Step 4 Remove the password from the private key. CISCO Advanced Web Security Reporting application does not support private key passwords.

On Linux:

```
$AWSR_HOME/bin/splunk cmd openssl rsa -in myAWSRWebPrivateKey.key -out myAWSRWebPrivateKey.key
```

On Windows:

```
$AWSR_HOME\bin\splunk cmd openssl rsa -in myAWSRWebPrivateKey.key -out myAWSRWebPrivateKey.key -config
$AWSR_HOME\openssl.cnf
```

You can use to following command to make sure that your password was successfully removed:

```
# openssl rsa -in myAWSRWebPrivateKey.key -text
```

If the password was successfully removed, you can view the certificate contents without providing a password.

Create a Certificate Authority (CA) request and obtain your server certificate

Step 1 Create a new certificate signature request using your private key `myAWSRWebPrivateKey.key`:

On Linux:

```
$AWSR_HOME/bin/splunk cmd openssl req -new -key myAWSRWebPrivateKey.key -out myAWSRWebCert.csr
```

On Windows:

```
$AWSR_HOME\bin\splunk cmd openssl req -new -key myAWSRWebPrivateKey.key -out myAWSRWebCert.csr
```

Note for Windows platforms: If you see an error similar to this:

```
Unable to load config info from

c:\\build-amd64-5.0.2-20130120-1800\\AWSR/ssl/openssl.cnf
```

Try typing the following in your command prompt then run the openssl command again:

```
set OPENSSL_CONF=c:/Program Files/AWSR/openssl.cnf
```

Step 2 Use the CSR `myAWSRWebCert.csr` to request a new signed certificate from your Certificate Authority (CA). The process for requesting a signed certificate varies depending on how your Certificate Authority handles a certificate signature request. Contact your CA for more information.

Step 3 Download the server certificate returned by your Certificate Authority. For this example, let's call it "`myAWSRWebCert.pem`".

Step 4 Download your Certificate Authority's public CA certificate. For this example, let's call it "`myCAcert.pem`".

Step 5 Make sure that both the server certificate and the public CA certificate are both in PEM format. If the certificates are not in PEM format, convert them using the `openssl` command appropriate to your existing file type. Here's an example of a command that you can use for DER formats:

```
$AWSR_HOME/bin/splunk cmd openssl x509 -in myAWSRWebCert.crt -inform DER -out myAWSRWebCert.pem
-outform PEM

$AWSR_HOME\bin\splunk cmd openssl x509 -in myCACert.crt -inform DER -out myCACert.pem -outform PEM
```

Step 6 Check both certificates to make sure they have the necessary information and are not password protected.

On Linux:

```
$AWSR_HOME/bin/splunk cmd openssl x509 -in myCACert.pem -text

$AWSR_HOME/bin/splunk cmd openssl x509 -in myAWSRWebCert.pem -text
```

On Windows:

```
$AWSR_HOME\bin\splunk cmd openssl x509 -in myCACert.pem -text

$AWSR_HOME\bin\splunk cmd openssl x509 -in myAWSRWebCert.pem -text
```

The issuer information for `myAWSRWebCert.pem` should be the subject information for `myCACert.pem` (unless you are using intermediary certificates).

Combine your certificate and keys into a single file

Combine your server certificate and public certificate, in that order, into a single PEM file.

Set up certificate chains

To use multiple certificates, append the intermediate certificate to the end of the server's certificate file in the following order:

```
[ server certificate]
[ intermediate certificate]
[ root certificate (if required) ]
```

So for example, a certificate chain might look like this:

```
-----BEGIN CERTIFICATE-----
... (certificate for your server)...
-----END CERTIFICATE-----
```

```
-----BEGIN CERTIFICATE-----
... (the intermediate certificate)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the root certificate for the CA)...
-----END CERTIFICATE-----
```

Note that the root CA that signed the intermediate certificate and all intermediary certificates must be in the browser certificate stores.

How to prepare your signed certificates for Cisco Advanced Web Security Reporting authentication

Once you have your certificates, you must combine the server certificate and your keys into a single file that Cisco Advanced Web Security Reporting software can use.

Note Make sure your certificates and public key are in x509 format and that your private key is in RSA format.

Create a single PEM file

Combine your server certificate and public certificate, in that order, into a single PEM file. For the examples here, we are using the file names described in Self-sign certificates for Cisco Advanced Web Security Reporting application and Get certificates signed by a third-party for Cisco Advanced Web Security Reporting application.

The following is an example for Linux:

```
cat myServerCertificate.pem myServerPrivateKey.key myCACertificate.pem >

myNewServerCertificate.pem
```

The following is an example for Windows:

```
type myServerCertificate.pem myServerPrivateKey.key myCACertificate.pem >

myNewServerCertificate.pem
```

Once created, the contents of the file `myNewServerCertificate.pem` should contain, in the following order:

- The server certificate (`myServerCertificate.pem`)
- The private key (`myServerPrivateKey.key`)
- The certificate authority public key (`myCACertificate.pem`)

Here's an example of a properly concatenated certificate:

```
-----BEGIN CERTIFICATE-----
MIICUTCCAboCCQCscBkn/xey1TANBgkqhkiG9w0BAQUFADBtMQswCQYDVQQGEwJV
...
<Server Certificate>
...
8/PZr3EuXYk1c+N5hgIQys5a/HIn
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
```

```
DEK-Info: DES-EDE3-CBC,CFCECC7976725DE5
S+DPcQ0l2Z1bk71N3cBqr/nwEXPNDQ4uqtecCd3iGMV3B/WSOWAQxcWzhe9JnIsl
...
<Server Private Key – Passphrase protected>
...
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIICUTCCAboCCQCscBkn/xey1TANBgkqhkiG9w0BAQUFADBtMQswCQYDVQQGEwJV
...
<Certificate Authority Public Key>
...
8/PZr3EuXYk1c+N5hgIQys5a/HIn
-----END CERTIFICATE-----
```

How to configure certificate chains

To use multiple certificates, append the intermediate certificate to the end of the server's certificate file. You can add as many certificates you need to in decreasing order of hierarchy, up to the root.

The certificates should be concatenated in the following order:

```
[ server certificate]
[ intermediate certificate]
[ root certificate (if required) ]
```

So for example, a certificate chain might look like this:

```
-----BEGIN CERTIFICATE-----
... (certificate for your server)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the intermediate certificate)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the root certificate for the CA)...
-----END CERTIFICATE-----
```

In another example, when using AWSR Forwarder to Indexer Certificates that contain a Private Key,the completed certificate file might look like this :

```
-----BEGIN CERTIFICATE-----
... (certificate for your server)...
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
...<Server Private Key – Passphrase protected>
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
... (certificate for your server)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the intermediate certificate)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the root certificate for the CA)...
-----END CERTIFICATE-----
```

Secure your deployment server and clients using certificate authentication

Authentication using signed certificates between deployment servers and clients is not recommended, because the configuration data pushed from the deployment server to client does not generally provide exploitable information. Configuring certificate authentication for a deployment server and clients impacts the rest of your configuration as follows:

- Cisco Advanced Web Security Reporting application will fail to authenticate unless you also configure it to use the certificate.
- The CLI will be not be able to communicate with the deployment server.

You may find certificate authentication necessary in certain distributed configurations, perhaps where extremely sensitive server configuration data is sent to a variety of locations outside your firewall. You can manually configure each indexer to communicate with your Deployment Server:

Note The deployment server cannot properly push certificates to peers. You must configure each member separately.

Step 1 Create one or more certificates using the same root CA.

Step 2 Distribute the certificates to your deployment server and clients.

Step 3 Edit **server.conf** to provide the location of your certificates:

```
[sslConfig]

enableSplunkdSSL = true

sslVersions = Defaults to "*,-ssl2" (anything newer than SSLv2). This is the recommended
setting.

serverCert = The full path to the PEM format server certificate file. Default
certificates

($SPLUNK_HOME/etc/auth/server.pem) are generated by Splunk at start. To secure Splunk,
you should replace the default cert with your own PEM file.

sslPassword = password

sslRootCAPath = absolute path to the operating system's root CA (Certificate Authority)
PEM

format file containing one or more root CA. Do not configure this attribute on Windows.
```

Step 4 Edit server.conf to authenticate against your certificates by adding the following attribute to the [sslConfig] stanza in previous step:

```
requireClientCert = true
```

Note This `requireClientCert` is set to “false” by default. If you change it to true to force Splunk to check your client's certificates, Cisco Advanced Web Security Reporting application and the CLI will also be checked for certificates. Your CLI connection will no longer work because your CLI is unable to present a certificate as a client.

Step 5 Edit web.conf to present a certificate signed by the same root CA so that Cisco Advanced Web Security Reporting application can connect to the server.

The following is an example of an edited settings stanza:

```
[settings]
enableSplunkWebSSL = true
privKeyPath = etc/auth/splunkweb/mySplunkWebPrivateKey.key
serverCert = etc/auth/splunkweb/mySplunkWebCertificate.pem
cipherSuite = <your chosen cipher suite (optional)>
```

Note Cisco Advanced Web Security Reporting application does not support passwords, so you must remove the password from the private key.

Troubleshoot your Cisco Advanced Web Security Reporting authentication

If you are unable to verify your certificate configuration, you can use the **web_service.log** in `$AWSR_HOME/var/log/splunk` to view and troubleshoot any errors that occur upon restart.

Look for SSL configuration warnings. For example, if you provide an incorrect path to the server certificate declared in `serverCert`, Cisco Advanced Web Security Reporting application fails to start and the following error appears:

```
2010-12-21 16:25:02,804 ERROR [4d11455df3182e6710] root:442 - [Errno 2] No such file or
directory: '/opt/splunk/share/splunk/mycerts/mySplunkWebCertificate.pem'
```

Note If the private key is provided in `privKeyPath` is password protected, no error is provided but your browser won't load Cisco Advanced Web Security Reporting application.