証明書の生成と署名

この章は、次のセクションで構成されています。

Cisco Advanced Web Security Reporting アプリケーションの自己署名証明書

このトピックでは、Cisco Advanced Web Security Reporting アプリケーションに含まれているバージョンの OpenSSL を使用して、コマンドラインで自己署名証明書を作成するための基本的な例を示します。

自己署名証明書は組織によって署名されるため、ブラウザの証明書ストアには含まれていません。その結果、Webブラウザは、自己署名証明書を「信頼できない」と見なします。これによってユーザへの警告ページが生成され、ユーザのアクセスが妨げられる場合もあります。

自己署名証明書は、Cisco Advanced Web Security Reporting アプリケーションに接続するすべてのブラウザストアに独自の CA を追加できる組織内または既知のエンティティ間での、ブラウザから Cisco Advanced Web Security Reporting アプリケーションへの通信に最適です。その他のシナリオでは、CA 署名付き証明書をお勧めします。詳細については、「Cisco Advanced Web Security Reporting アプリケーション用のサードパーティにより署名された証明書の取得」を参照してください。

はじめる前に

この説明では、`$AWSR_HOME` は AWSR Enterprise のインストールディレクトリを指します。この表記法に従うことをお勧めしますが、この表記法を使用しない場合は、これらの例を使うときに `$AWSR_HOME` をインストールディレクトリに置き換える必要があります。

Windows の場合、コマンドラインまたは [システムのプロパティ（System Properties）] ダイアログの [環境（Environment）] タブで、この変数の設定が必要になる場合があります。デフォルトのホームディレクトリは、プラットフォームによって異なります。

- Windows の場合、AWSR Enterprise ディレクトリはデフォルトで `C:\Program Files\Cisco\` です。
- ほとんどの Linux プラットフォームでは、デフォルトのインストールディレクトリは `/opt/` です。

認証局となる新しいルート証明書の生成

ステップ 1 証明書とキーをホストする新しいディレクトリを作成します。この例では、`$AWSR_HOME/etc/auth/mycerts` を使用します。

ステップ 2 新しい RSA 秘密キーを生成します。Cisco Advanced Web Security Reporting アプリケーションは 2048 ビットのキーをサポートしていますが、ブラウザでサポートされている場合は、より大きなキーを指定できます。

Linux の場合：

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

Windows の場合：

Windows では、openssl.cnf ファイルの場所の追加が必要な場合があることに注意してください。

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

Cisco Advanced Web Security Reporting アプリケーションは 2048 ビットのキーをサポートしていますが、ブラウザでサポートされている場合は、より大きなキーを指定できます。

ステップ 3 プロンプトが表示されたら、パスワードを作成します。

秘密キー `myCAPrivateKey.key` がディレクトリに表示されます。これはルート証明書の秘密キーです。

ステップ 4 ルート証明書の秘密キー `myCAPrivateKey.key` を使用して証明書署名要求を生成します。

Linux の場合：

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

Windows の場合：

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

ステップ 5 秘密キー `myCAPrivateKey.key` にパスワードを入力します。

新しい CSR myCACertificate.csr がディレクトリに表示されます。

ステップ 6 CSR を使用して、新しいルート証明書を生成し、秘密キーで署名します。

Linux の場合：

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

Windows の場合：

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

ステップ 7 プロンプトが表示されたら、秘密キー `myCAPrivateKey.key` にパスワードを入力します。

新しい証明書 `myCACertificate.pem` がディレクトリに表示されます。これが自身のパブリック証明書です。

Cisco Advanced Web Security Reporting アプリケーションの新しい秘密キーの作成

ステップ 1 新しい秘密キーを生成します。

Linux の場合：

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

Windows の場合：

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

ステップ 2 プロンプトが表示されたら、パスワードを作成します。

新しいキー `myAWSRWebPrivateKey.key` がディレクトリに表示されます。

ステップ 3 キーからパスワードを削除します（Cisco Advanced Web Security Reporting アプリケーションは、パスワードで保護された秘密キーをサポートしていません）。

Linux の場合：

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

Windows の場合：

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

次のコマンドを使用して、パスワードが削除されたことを確認できます。

Linux の場合：

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

Windows の場合：

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

パスワードを入力しなくても、証明書の内容を読み取ることができます。

サーバ証明書の作成と署名

ステップ 1 秘密キー `myAWSRWebPrivateKey.key` を使用して新しい証明書署名要求を作成します。

Linux の場合：

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

Windows の場合：

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

CSR `myAWSRWebCert.csr` がディレクトリに表示されます。

ステップ 2 ルート証明書の秘密キー `myCAPrivateKey.key` を使用して CSR に自己署名します。

Linux の場合：

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

Windows の場合：

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

ステップ 3 プロンプトが表示されたら、ルート証明書の秘密キー `myCAPrivateKey.key` にパスワードを入力します。

証明書 `myAWSRWebCert.pem` がディレクトリに追加されます。これはサーバ証明書です。

単一の PEM ファイルの作成

サーバ証明書とパブリック証明書をこの順序で 1 つの PEM ファイルに結合します。

Linux でこれを行う例を次に示します。

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

Windows の例を次に示します。

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

証明書チェーンの設定

複数の証明書を使用するには、次の順序でサーバの証明書ファイルの最後に中間証明書を追加します。

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

たとえば、証明書チェーンは次のようになります。

```
-----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-----
```

Cisco Advanced Web Security Reporting アプリケーション用のサードパーティにより署名された証明書の取得

このトピックでは、SSL 認証および暗号化用に Cisco Advanced Web SecurityReporting アプリケーションを構成するために必要な第三者の署名付き証明書を作成するための基本的な例を示します。

Cisco Advanced Web Security Reporting アプリケーションの新しい秘密キーの作成

ステップ 1 独自の証明書とキーをホストするための新しいディレクトリを作成します。この例では、`$AWSR_HOME/etc/auth/mycerts` を使用します。

既存の証明書を上書きしないように、新しい証明書を `$AWSR_HOME/etc/auth/splunkweb` とは別のディレクトリに保存することをお勧めします。これにより、必要に応じて AWSR に付属の証明書を他の AWSR コンポーネントに使用できます。

ステップ 2 新しい秘密キーを生成します。Cisco Advanced Web Security Reporting アプリケーションは、2048 ビット以上のキーをサポートします。

Linux の場合：

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

Windows の場合：

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

ステップ 3 元のキーのパスフレーズを入力するように求められたら、パスワードを作成します。

新しい秘密キー `myAWSRWebPrivateKey.key` がディレクトリに追加されます。このキーを使用して、CSR に署名できます。

ステップ 4 秘密キーからパスワードを削除します。Cisco Advanced Web Security Reporting アプリケーションは、秘密キーのパスワードをサポートしていません。

Linux の場合：

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

Windows の場合：

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

次のコマンドを使用して、パスワードが正常に削除されたことを確認できます。

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

パスワードが正常に削除された場合は、パスワードを入力しなくても証明書の内容を表示できます。

認証局（CA）要求の作成とサーバ証明書の取得

ステップ 1 秘密キー `myAWSRWebPrivateKey.key` を使用して新しい証明書署名要求を作成します。

Linux の場合：

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

Windows の場合：

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

Windows プラットフォームに関する注意：次のようなエラーが表示された場合

```
Unable to load config info from

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

コマンドプロンプトに次のように入力してから、openssl コマンドを再度実行してください。

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

ステップ 2 CSR `myAWSRWebCert.csr` を使用して、認証局（CA）に新しい署名証明書を要求します。署名証明書を要求するプロセスは、認証局が証明書署名要求を処理する方法によって異なります。詳細については、CA に問い合わせてください。

ステップ 3 認証局から返されたサーバ証明書をダウンロードします。この例では、この証明書を「`myAWSRWebCert.pem`」とします。

ステップ 4 認証局の公開 CA 証明書をダウンロードします。この例では、この証明書を「`myCAcert.pem`」とします。

ステップ 5 サーバ証明書とパブリック CA 証明書の両方が PEM 形式であることを確認してください。証明書が PEM 形式でない場合は、既存のファイルタイプに適した `openssl` コマンドを使用して証明書を変換します。DER 形式に使用できるコマンドの例を次に示します。

```
$ 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 myAWSRWebCert.crt -inform DER -out myAWSRWebCert.pem
-outform PEM
```

ステップ 6 両方の証明書をチェックして、必要な情報が記載されていて、パスワードで保護されていないことを確認してください。

Linux の場合：

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

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

Windows の場合：

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

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

`myAWSRWebCert.pem` の発行者情報は、`myCACert.pem` のサブジェクト情報である必要があります（中間証明書を使用している場合を除く）。

証明書とキーを 1 つのファイルに結合

サーバ証明書とパブリック証明書をこの順序で 1 つの PEM ファイルに結合します。

証明書チェーンの設定

複数の証明書を使用するには、次の順序でサーバの証明書ファイルの最後に中間証明書を追加します。

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

たとえば、証明書チェーンは次のようになります。

```
-----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-----
```

中間証明書に署名したルート CA とすべての中間証明書はブラウザの証明書ストアに存在する必要があることに注意してください。

Cisco Advanced Web Security Reporting 認証用に署名証明書を準備する方法

証明書を取得したら、サーバ証明書とキーを 1 つのファイルにまとめて、Cisco Advanced Web Security Reporting ソフトウェアで使用できるようにする必要があります。

（注） 証明書と公開キーが x509 形式であること、および秘密キーが RSA 形式であることを確認します。

単一の PEM ファイルの作成

サーバ証明書とパブリック証明書をこの順序で 1 つの PEM ファイルに結合します。この例では、「Cisco Advanced Web Security Reporting アプリケーションの自己署名証明書」および「Cisco Advanced Web Security Reporting アプリケーション用のサードパーティにより署名された証明書の取得」で説明されているファイル名を使用しています。

以下は Linux の例です。

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

myNewServerCertificate.pem
```

以下は Windows の例です。

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

myNewServerCertificate.pem
```

作成したら、ファイル `myNewServerCertificate.pem` の内容に以下をこの順序で含める必要があります。

- サーバー証明書（`myServerCertificate.pem`）
- 秘密キー（`myServerPrivateKey.key`）
- 認証局の公開キー（`myCACertificate.pem`）

適切に連結された証明書の例を次に示します。

```
-----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-----
```

証明書チェーンの設定方法

複数の証明書を使用するには、サーバの証明書ファイルの最後に中間証明書を追加します。ルートまで、階層の降順で必要な数の証明書を追加できます。

証明書は次の順序で連結されます。

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

たとえば、証明書チェーンは次のようになります。

```
-----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-----
```

もう 1 つの例では、秘密キーを含む AWSR フォワーダからインデクサへの証明書を使用すると、完成した証明書ファイルは次のようになります。

```
-----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-----
```

証明書認証を使用した導入サーバとクライアントの保護

導入サーバからクライアントにプッシュされた構成データは不正利用可能な情報を提供しないため、導入サーバとクライアント間で署名付き証明書を使用した認証はお勧めしません。導入サーバとクライアントの証明書認証を構成すると、他の構成に次のような影響があります。

- Cisco Advanced Web Security Reporting アプリケーションは、証明書を使用するように設定しない限り、認証に失敗します。
- CLI は導入サーバと通信できなくなります。

非常に機密性の高いサーバ構成データがファイアウォールの外側のさまざまな場所に送信される可能性のある特定の分散構成では、証明書認証が必要になる場合があります。導入サーバと通信するように各インデクサを手動で設定できます。

（注） 導入サーバは、証明書をピアに正しくプッシュできません。各メンバーを個別に設定する必要があります。

ステップ 1 同じルート CA を使用して 1 つ以上の証明書を作成します。

ステップ 2 証明書を導入サーバとクライアントに配布します。

ステップ 3 **server.conf** を編集して、証明書の場所を指定します。

```
[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.
```

ステップ 4 前の手順の [sslConfig] スタンザに次の属性を追加して、server.conf を編集して証明書に対して認証します。

```
requireClientCert = true
```

（注） この `requireClientCert` は、デフォルトで「false」に設定されています。これを true に変更して Splunk にクライアントの証明書を強制的にチェックさせると、Cisco Advanced Web Security Reporting アプリケーションと CLI でも証明書がチェックされます。CLI では証明書をクライアントとして提示できないため、CLI 接続は機能しなくなります。

ステップ 5 web.conf を編集して同じルート CA で署名された証明書を提示して、Cisco Advanced Web Security Reporting アプリケーションがサーバに接続できるようにします。

以下は編集された設定スタンザの例です。

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

（注） Cisco Advanced Web Security Reporting アプリケーションはパスワードをサポートしていないため、秘密キーからパスワードを削除する必要があります。

Cisco Advanced Web Security Reporting 認証のトラブルシューティング

証明書の構成を確認できない場合は、`$AWSR_HOME/var/log/splunk` の **web_service.log** を使用して、再起動時に発生するエラーを表示およびトラブルシューティングできます。

SSL構成の警告を探します。たとえば、`serverCert` で宣言されたサーバ証明書への誤ったパスを指定すると、Cisco Advanced Web Security Reporting アプリケーションの起動に失敗し、次のエラーが表示されます。

```
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'
```

（注） `privKeyPath` で提供されている秘密キーがパスワードで保護されている場合、エラーは発生しませんが、ブラウザでは Cisco Advanced Web SecurityReporting アプリケーションがロードされません。