証明書

デジタル証明書は、認証に使用されるデジタル ID を提供します。証明書は、SSL（セキュア ソケット レイヤ）、TLS（Transport Layer Security）、および DTLS（データグラム TLS）接続（HTTPS や LDAPS など）に使用されます。次のトピックでは、証明書の作成と管理の方法について説明します。

証明書について

デジタル証明書は、認証に使用されるデジタル ID を保持しています。デジタル証明書には、名前、シリアル番号、会社、部門、または IP アドレスなど、ユーザまたはデバイスを識別する情報が含まれます。デジタル証明書には、ユーザまたはデバイスの公開キーのコピーも含まれています。証明書は、SSL（セキュア ソケット レイヤ）、TLS（Transport Layer Security）、および DTLS（データグラム TLS）接続（HTTPS や LDAPS など）に使用されます。

次のタイプの証明書を作成できます。

- 内部証明書：内部アイデンティティ証明書は、特定のシステムまたはホストの証明書です。これらは OpenSSL ツールキットを使用して自分で生成することも、認証局から取得することもできます。自己署名証明書を生成することもできます。
- 内部証明書認証局（CA）証明書：内部 CA 証明書は、他の証明書の署名にシステムが使用できる証明書です。これらの証明書は、基本制約拡張と CA フラグに関して内部アイデンティティ証明書と異なります。これらは CA 証明書では有効ですが、アイデンティティ証明書では無効です。これらは OpenSSL ツールキットを使用して自分で生成することも、認証局から取得することもできます。自己署名内部 CA 証明書を生成することもできます。自己署名内部 CA 証明書を設定する場合は、CA はデバイス自体で稼働します。
- 信頼できる認証局（CA）証明書：信頼できる CA 証明書は、他の証明書に署名するために使用されます。これは自己署名され、ルート証明書と呼ばれます。別の CA 証明書により発行される証明書は、下位証明書と呼ばれます。

認証局（CA） は、証明書に「署名」してその認証を確認することで、デバイスまたはユーザのアイデンティティを保証する、信頼できる機関です。CA は、公開キーまたは秘密キーの暗号化を使用してセキュリティを保証する PKI コンテキストで、デジタル証明書を発行します。CA は、信頼できるサードパーティ（VeriSign など）の場合もあれば、組織内に設置したプライベート CA（インハウス CA）の場合もあります。CA は、証明書要求の管理とデジタル証明書の発行を行います。詳細については、公開キー暗号化 （2 ページ）を参照してください。

公開キー暗号化

RSA 暗号化システムなどの公開キー暗号化では、各ユーザは、公開キーと秘密キーの両方を含むキーペアを使用します。これらのキーは、補足として機能し、一方で暗号化されたものは、もう一方で復号できます。

簡単に言えば、データが秘密キーで暗号化されたとき、署名が形成されます。署名はデータに付加されて受信者に送信されます。受信者は送信者の公開キーをデータに適用します。データとともに送信された署名が、公開キーをデータに適用した結果と一致した場合、メッセージの有効性が確立されます。

このプロセスは、受信者が送信者の公開キーのコピーを持っていること、およびその公開キーが送信者になりすました別人のものではなく、送信者本人のものであることを受信者が強く確信していることに依存しています。

通常、送信者の公開キーは外部で取得するか、インストール時の操作によって取得します。たとえば、ほとんどの Web ブラウザでは、いくつかの CA のルート証明書がデフォルトで設定されています。

デジタル証明書および公開キー暗号化の詳細については、openssl.org、Wikipedia、またはその他のソースを参照してください。SSL/TLS 暗号化をしっかりと理解することで、デバイスへのセキュアな接続を確立できます。

各機能で使用される証明書タイプ

各機能に適したタイプの証明書を作成する必要があります。次の機能は、証明書が必要です。

アイデンティティ ポリシー（キャプティブ ポータル）：内部証明書

（オプション）キャプティブ ポータルはアイデンティティ ポリシーで使用されます。この証明書は、ユーザが自身を特定し、自分のユーザ名にデバイスの IP アドレスを関連付けることを目的としてデバイスを認証するときに承認する必要があります。証明書を提示しないと、デバイスは自動生成された証明書を使用します。

アイデンティティ レルム（アイデンティティ ポリシーおよびリモート アクセス VPN）：信頼できる CA 証明書

（オプション）ディレクトリ サーバに暗号化接続を使用する場合、ディレクトリ サーバの認証を行うためにこの証明書を承認する必要があります。ユーザは、アイデンティティ ポリシーおよびリモート アクセス VPN ポリシーから求められたときに認証する必要があります。ディレクトリ サーバに暗号化を使用しない場合、証明書は必要ありません。

管理 Web サーバ（管理アクセス システム設定）：内部証明書

（オプション）Firepower Device Manager は Web ベースのアプリケーションであり、Web サーバで動作します。お使いのブラウザで有効として受け入れられる証明書をアップロードすると、Untrusted Authority の警告を受けるのを回避できます。

リモート アクセス VPN：**内部証明書**

（必須）内部証明書は、AnyConnect クライアントがデバイスへの接続を行うときにデバイス ID を確立する外部インターフェイスに使用します。クライアントはこの証明書を承認する必要があります。

サイト間 VPN：**内部および信頼できる CA 証明書**

サイト間 VPN 接続に証明書認証を使用する場合は、接続内のローカルピア認証に使用される内部アイデンティティ証明書を選択する必要があります。これは VPN 接続の定義の一部ではありませんが、システムがピアを認証できるように、ローカルおよびリモートピアのアイデンティティ証明書に署名するために使用した信頼できる CA 証明書をアップロードする必要があります。

SSL 復号ポリシー：**内部、内部 CA、および信用できる CA 証明書**

（必須）SSL 復号ポリシーは、以下の目的のため証明書を使用します。

- 内部証明書は既知のキー復号ルールに使用されます。
- 内部CA証明書は、クライアントとFTDデバイス間にセッションを作成するときに、再署名の復号ルールに使用されます。
- 信頼できるCA証明書は、FTDデバイスとサーバ間にセッションを作成するときに、再署名の復号ルールに間接的に使用されます。その他の証明書とは異なり、これらの証明書はSSL復号ポリシーで直接設定しません。これらは単にシステムにアップロードする必要があります。システムには多数の信用できるCA証明書が含まれるため、追加の証明書をアップロードする必要はないことがあります。

例：OpenSSL を使用した内部証明書の生成

次の例では、OpenSSL コマンドを使用して内部サーバの証明書を生成します。OpenSSL は openssl.org から取得できます。具体的な情報については、OpenSSL のマニュアルを参照してください。この例で使用するコマンドは変更される場合があり、この他にも利用できるオプションがある可能性もあります。

この手順は、FTD にアップロードする証明書の取得方法について、1 つの考え方を示すものです。

（注） 次に示す OpenSSL コマンドは一例にすぎません。セキュリティ要件に合わせてパラメータを調整してください。

手順

ステップ 1 キーを生成します。

```
openssl genrsa -out server.key 4096
```

ステップ 2 証明書署名要求（CSR）を生成します。

```
openssl req -new -key server.key -out server.csr
```

ステップ 3 キーと CSR を持つ自己署名証明書を生成します。

```
openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
```

Firepower Device Manager は暗号化キーをサポートしないため、自己署名証明書を生成するときはリターン キーを押してチャレンジ パスワードをスキップしてください。

ステップ 4 内部証明書のオブジェクトを Firepower Device Manager で作成するときは、正しいフィールドにファイルをアップロードします。

ファイルの内容をコピーして貼り付けることもできます。サンプル コマンドは、次のファイルを作成します。

- server.crt：[サーバ証明書（Server Certificate）] フィールドにコンテンツをアップロードするか、貼り付けます。
- server.key：[証明書キー（Certificate Key）] フィールドにコンテンツをアップロードするか、貼り付けます。キーの生成時にパスワードを入力すると、次のコマンドを使用してそれを復号できます。出力は stdout に送信され、コピーできます。

  ```
  openssl rsa -in server.key -check
  ```

証明書の設定

FTDPEM または DER 形式の X509 証明書をサポートします。OpenSSL を使用して必要に応じて証明書を生成、信頼できる認証局から取得、または自己署名証明書を作成します。

証明書の詳細については、証明書について （1 ページ）を参照してください。

各機能にどのタイプが使用されているかについては、各機能で使用される証明書タイプ （2 ページ）を参照してください。

次に、[オブジェクト（Objects）]ページで直接オブジェクトを作成および編集する方法について説明します。オブジェクト リストに表示されている [新規証明書の作成（Create New

Certificate）] リンクをクリックし、証明書プロパティを編集しながら、証明書オブジェクトを作成することもできます。

手順

ステップ 1 [オブジェクト（Objects）] を選択し、目次から [証明書（Certificates）] を選択します。

システムには、そのまま、または置き換えて使用できる次の事前定義された証明書が付属します。

- DefaultInternalCertificate
- DefaultWebserverCertificate
- NGFW-Default-InternalCA

システムには、サードパーティ証明機関からの多数の信頼された CA の証明書も含まれています。これらは再署名の復号アクションのために SSL 復号化ポリシーが使用します。

ステップ 2 次のいずれかを実行します。

- 新しい証明書オブジェクトを作成するには、[+] メニューから証明書のタイプに適したコマンドを使用します。
- 証明書を表示または編集するには、証明書の [編集（edit）] アイコン（）または [表示（view）]アイコン（）をクリックします。
- 証明書を削除するには、その証明書のごみ箱アイコン（）をクリックします。

証明書の作成と編集の詳細については、次のトピックを参照してください。

- 内部および内部 CA 証明書のアップロード （5 ページ）
- 自己署名内部および内部 CA 証明書の生成 （7 ページ）
- 信頼できる CA 証明書のアップロード （8 ページ）

内部および内部 CA 証明書のアップロード

内部アイデンティティ証明書は、特定のシステムまたはホストの証明書です。

内部 CA 証明書は、他の証明書の署名に使用できる証明書です。これらの証明書は、基本制約拡張と CA フラグに関して内部アイデンティティ証明書と異なります。これらは CA 証明書では有効ですが、アイデンティティ証明書では無効です。

この証明書は、OpenSSL ツールキットを使用して自分で生成するか、認証局から取得できます。その後、次の手順を使用してアップロードします。キー生成の例については、例：OpenSSL を使用した内部証明書の生成 （3 ページ）を参照してください。

自己署名内部アイデンティティ証明書および内部 CA 証明書を生成することもできます。自己署名内部 CA 証明書を設定する場合は、CA はデバイス自体で稼働します。自己署名証明書の作成の詳細については、自己署名内部および内部 CA 証明書の生成 （7 ページ）を参照してください。

これらの証明書を使用する機能の詳細については、各機能で使用される証明書タイプ （2 ページ）を参照してください。

手順

ステップ 1 [オブジェクト（Objects）] を選択し、目次から [証明書（Certificates）] を選択します。

ステップ 2 次のいずれかを実行します。

- **[+] > [内部証明書の追加（Add Internal Certificate）]** をクリックし、次に [証明書とキーのアップロード（Upload Certificate and Key）] をクリックします。
- **[+] > [内部CA証明書の追加（Add Internal CA Certificate）]** をクリックし、次に [証明書とキーのアップロード（Upload Certificate and Key）] をクリックします。
- 証明書を編集または表示するには、情報アイコン（）をクリックします。ダイアログボックスには、証明書の件名、発行者、および有効な時間範囲が表示されます。[証明書の置換（Replace Certificate）] をクリックして、新しい証明書とキーをアップロードします。ダイアログ ボックスで証明書とキーを貼り付けることもできます。

ステップ 3 証明書の名前を入力します。

名前は、設定時にオブジェクト名としてのみ使用され、証明書自体には含まれません。

ステップ 4 [証明書のアップロード（Upload Certificate）]（編集する場合は、[証明書の置換（Replace Certificate）]）をクリックし、証明書ファイル（例：*.crt）を選択します。許可されるファイル拡張子は、.pem、.cert、.cer、.crt、および .der です。または、証明書に貼り付けます。

証明書は PEM または DER 形式の X509 証明書である必要があります。

貼り付ける証明書は、BEGIN CERTIFICATE と END CERTIFICATE の行を含める必要があります。次に例を示します。

```
-----BEGIN CERTIFICATE-----
MIICMTCCAZoCCQDdUV3NGK/cUjANBgkqhkiG9w0BAQsFADBdMQswCQYDVQQGEwJV
UzETMBEGA1UECAwKU29tZS1TdGF0ZTEhMB8GA1UECgwYSW50ZXJuZXQgV2lkZ2l0
(...5 lines removed...)
shGJDReRYJQqilhHZrYTWZAYTrD7NQPHutK+ZiJng67cPgnNDuXEn55UwMOQoHBp
HMUwmhiGZlzJM8BpX2Js2yQ3ms30pr8rO+gPCPMCAwEAATANBgkqhkiG9w0BAQsF
AAOBgQCB02CebA6YjJCGr2CJZrQSeUwSveRBpmOuoqm98o2Z+5gJM5CkqgfxwCUn
RV7LRfQGFYd76V/5uor4Wx2ZCjqy6+zuQEm4ZxWNSZpA9UBixFXJCs9MBO4qkG5D
vlk3WYJfcgyJ10h4E4b0W2xiixBU+xoOTLRATnbKY36EWAG5cw==
-----END CERTIFICATE-----
```

ステップ 5 [キーのアップロード（Upload Key）]（または編集時に、[キーの交換（Replace Key）]）をクリックし、証明書ファイル（例：*.key）を選択します。ファイル拡張子は .key である必要があります。または、証明書のキーに貼り付けます。

キーは暗号化できません。

次に例を示します。

```
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQClSu1BknrMjzw/5FZ9YgdMLDUGJlbYgkjN7mVrkjyLQx2TYsem
r8iTiKB6iyTKbuS4iPeyEYkNF5FglCqKWEdmthNZkBhOsPs1A8e60r5mImeDrtw+
Cc0O5cSfnlTAw5CgcGkcxTCaGIZmXMkzwGlfYmzbJDeazfSmvys76A8I8wIDAQAB
AoGAUVDgEX8vXE0m9cOubPZ54pZo64KW/OJzUKP0TwxdLqGw/h39XFpkEXiIgmDL
(...5 lines removed...)
DSWvzekRDH83dmP66+MIbWePhbhty+D1OxbiuVuHV0/ZhxOhCG8tig3R8QJBAJmj
fId05+1dNI4tGbWv6hHh/H/dTP2STlZ3jERMZd29fjIRuJ9jpFC2lIDjvs8YGeAe
0YHkfSOULJn8/jOCf6kCQQDIJiHfGF/31Dk/8/5MGrg+3zau6oKXiuv6db8Rh+7l
MUOx09tvbBUy9REJq1YJWTKpeKD+E0QL+FX0bqvz4tHA
-----END RSA PRIVATE KEY-----
```

ステップ 6 [OK] をクリックします。

自己署名内部および内部 CA 証明書の生成

内部アイデンティティ証明書は、特定のシステムまたはホストの証明書です。

内部 CA 証明書は、他の証明書の署名に使用できる証明書です。これらの証明書は、基本制約拡張と CA フラグに関して内部アイデンティティ証明書と異なります。これらは CA 証明書では有効ですが、アイデンティティ証明書では無効です。

ユーザは、自己署名内部アイデンティティと内部 CA 証明書を生成できます。つまり、証明書はデバイス自体によって署名されます。自己署名内部 CA 証明書を設定すると、CA がデバイス上で有効になります。システムは、証明書とキーの両方を生成します。

また、これらの証明書は、OpenSSL を使用して作成することも、信頼できる CA から取得してアップロードすることもできます。詳細については、内部および内部 CA 証明書のアップロード （5 ページ）を参照してください。

これらの証明書を使用する機能の詳細については、各機能で使用される証明書タイプ （2 ページ）を参照してください。

（注） 新しい自己署名証明書は 5 年の有効期間で生成されます。期限が切れる前に必ず証明書を交換してください。

手順

ステップ 1 [オブジェクト（Objects）] を選択し、目次から [証明書（Certificates）] を選択します。

ステップ 2 次のいずれかを実行します。

- **[+] > [内部証明書の追加（Add Internal Certificate）]** をクリックし、次に [自己署名証明書（Self-Signed Certificate）] をクリックする。
- **[+] > [内部CA証明書の追加（Add Internal CA Certificate）]** をクリックし、次に [自己署名証明書（Self-Signed Certificate）] をクリックする。

（注） 証明書を編集または表示するには、情報アイコン（ⓘ）をクリックします。ダイアログ ボックスには、証明書の件名、発行者、および有効な時間範囲が表示されます。[証明書の置換（Replace Certificate）] をクリックして、新しい証明書とキーをアップロードします。証明書を交換する際は、次の手順で説明されている自己署名の特性を設定し直すことはできません。代わりに、内部および内部 CA 証明書のアップロード（5 ページ）の説明に従って、新しい証明書を貼り付けるかアップロードする必要があります。残りの手順は、新しい自己署名証明書のみに適用されます。

ステップ 3 証明書の名前を入力します。

名前は、設定時にオブジェクト名としてのみ使用され、証明書自体には含まれません。

ステップ 4 証明書の件名および発行者の情報については、次の少なくとも 1 つを設定します。

- **Country**（C）：証明書に含める 2 文字の ISO 3166 国コード。たとえば、米国の国コードは US です。ドロップダウン リストから国コードを選択します。
- **State or Province**（ST）：証明書に含める都道府県または州。
- **Locality or City**（L）：都市の名前など、証明書に含める地域。
- **Organization**（O）：証明書に含める組織または会社の名前。
- **Organizational Unit (Department)**（OU）：証明書に含める組織単位の名前（部門名など）。
- **Common Name**（CN）：証明書に含める X.500 共通名。これは、デバイスの名前、Web サイト、または他の文字列にできます。この要素は、通常は正常な接続のために必要です。たとえば、リモート アクセス VPN で使用する内部証明書に CN を含める必要があります。

ステップ 5 [保存（Save）] をクリックします。

信頼できる CA 証明書のアップロード

信頼できる認証局（CA）の証明書は、他の証明書に署名するために使用されます。これは自己署名され、ルート証明書と呼ばれます。別の CA 証明書により発行される証明書は、下位証明書と呼ばれます。

これらの証明書を使用する機能の詳細については、各機能で使用される証明書タイプ（2 ページ）を参照してください。

外部の認証局から信頼できる CA 証明書を取得するか、自身の内部 CA を使用して（OpenSSL ツールを使用するなど）CA 証明書を作成します。その後、次の手順を使用して証明書をアップロードします。

手順

ステップ 1 [オブジェクト（Objects）] を選択し、目次から [証明書（Certificates）] を選択します。

ステップ 2 次のいずれかを実行します。

- **[+] > [信頼済みCAの証明書の追加（Add Trusted CA Certificate）]** をクリックします。
- 証明書を編集するには、その証明書の編集アイコン（）をクリックします。

ステップ 3 証明書の名前を入力します。

名前は、設定時にオブジェクト名としてのみ使用され、証明書自体には含まれません。

ステップ 4 [証明書のアップロード（Upload Certificate）]（または、編集時は [証明書の置換（Replace Certificate）]）をクリックして、信頼できる CA 証明書ファイル（*.pem など）を選択します。許可されるファイル拡張子は、.pem、.cert、.cer、.crt、および .der です。または、信頼できる CA 証明書に貼り付けます。

証明書内のサーバ名は、サーバのホスト名または IP アドレスと一致している必要があります。たとえば、IP アドレスとして 10.10.10.250 を使用しているのに、証明書で ad.example.com を使用すると接続が失敗します。

証明書は PEM または DER 形式の X509 証明書である必要があります。

貼り付ける証明書は、BEGIN CERTIFICATE と END CERTIFICATE の行を含める必要があります。次に例を示します。

```
-----BEGIN CERTIFICATE-----
MIIFgTCCA2mgAwIBAgIJANvdcLnabFGYMA0GCSqGSIb3DQEBCwUAMFcxCzAJBgNV
BAYTAlVTMQswCQYDVQQIDAJUWDEPMA0GA1UEBwwGYXVzdGluMRQwEgYDVQQKDAsx
OTIuMTY4LjEuMTEUMBIGA1UEAwwLMTkyLjE2OC4xLjEwHhcNMTYxMDI3MjIzNDE3
WhcNMTcxMDI3MjIzNDE3WjBXMQswCQYDVQQGEwJVUzELMAkGA1UECAwCVFgxDzAN
BgNVBAcMBmF1c3RpbjEUMBIGA1UECgwLMTkyLjE2OC4xLjExFDASBgNVBAMMCzE5
Mi4xNjguMS4xMIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEA5NceYwtP
ES6Ve+S9z7WLKGX5JlF58AvH82GPkOQdrixn3FZeWLQapTpJZt/vgtAI2FZIK31h
(...20 lines removed...)
hbr6HOgKlOwXbRvOdksTzTEzVUqbgxt5Lwupg3b2ebQhWJz4BZvMsZX9etveEXDh
PY184V3yeSeYjbSCF5rP71fObG9Iu6+u4EfHp/NQv9s9dN5PMffXKieqpuN20Ojv
2b1sfOydf4GMUKLBUMkhQnip6+3W
-----END CERTIFICATE-----
```

ステップ 5 [OK] をクリックします。