CHAPTER 12

証明書の設定

この章では、証明書の設定方法について説明します。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。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。CA は、信頼できる第三者（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。

この章では、次の内容について説明します。

公開キー暗号化

ここでは、次の内容について説明します。

公開キー暗号化について

デジタル署名は、公開キー暗号化によってイネーブルになり、デバイスおよびユーザを認証する手段です。RSA 暗号化システムをはじめとする公開キー暗号化では、各ユーザが公開キーと秘密キーの両方を含むキー ペアを持ちます。両キーは補完的に動作し、片方のキーで暗号化されたものはすべて他方のキーで解読できます。

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、アウトオブバンドで取得するか、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。VPN の場合、IKE プロトコルは IPSec のコンポーネントであり、デジタル署名を使用してピア デバイスを認証した後で、セキュリティ アソシエーションをセットアップできます。

証明書のスケーラビリティ

デジタル証明書がない場合は、通信相手のピアごとに各 IPSec ピアを手動で設定する必要があります。また、ネットワークに新しいピアを追加するたびに、セキュアに通信する必要のあるピアごとの設定変更が必要になります。

デジタル証明書を使用している場合、各ピアは CA に登録されます。2 つのピアは、通信を試みるときに、証明書とデジタル署名されたデータを交換して、相互の認証を行います。新しいピアがネットワークに追加された場合は、そのピアを CA に登録するだけで済みます。他のピアを修正する必要はありません。新しいピアが IPSec 接続を試みると、証明書が自動的に交換され、そのピアの認証ができます。

CA を使用した場合、ピアはリモート ピアに証明書を送り、公開キー暗号化を実行することによって、そのリモート ピアに対して自分自身を認証します。各ピアは、CA によって発行および検証された一意の証明書を送ります。このプロセスが機能を果たすのは、関連付けられているピアの公開キーが各証明書にカプセル化され、各証明書が CA によって認証され、参加しているすべてのピアによって CA が認証権限者として認識されるためです。このプロセスは、RSA シグニチャ付きの IKE と呼ばれます。

ピアは、証明書が期限満了になるまで、複数の IPSec セッションに対して、および複数の IPSec ピア宛てに証明書を送り続けることができます。証明書が期限満了になったときは、ピアの管理者は新しい証明書を CA から入手する必要があります。

CA は、IPSec に参加しなくなったピアの証明書を無効にすることもできます。無効にされた証明書は、他のピアからは有効な証明書とは認識されなくなります。無効にされた証明書は CRL に記載され、各ピアは別のピアの証明書を受け取る前に、CRL をチェックします。

CA の中には、実装の一部として RA を持つものもあります。RA は CA のプロキシの役割を果たすサーバであるため、CA が使用できないときも CA 機能は継続しています。

キー ペアについて

キー ペアは SSH または SSL 接続に使用できる RSA キーであり、以下の特徴があります。

- キー生成では、RSA キーの最大キー係数は 2048 ビットです。デフォルトのサイズは 1024 ビットです。1024 ビットを超える RSA キー ペアによるアイデンティティ証明書を使用する多数の SSL 接続では、FWSM での CPU 使用率が高くなり、クライアントレス ログインが拒否される可能性があります。
- 署名操作でサポートされているキーの最大サイズは 4096 ビットです。
- 署名にも暗号化にも使用できる汎用 RSA キー ペアを生成することも、署名用と暗号化用に別々の RSA キー ペアを生成することもできます。

  署名用と暗号化用にキーを分けると、キーが公開される頻度を少なくすることができます。これは、SSL は署名用ではなく暗号化用にキーを使用しますが、IKE は暗号化用ではなく署名用にキーを使用するためです。

トラストポイントについて

トラストポイントを使用すると、CA と証明書の管理とトレースができます。トラストポイントとは、CA または ID ペアを表現したものです。トラストポイントには、CA の ID、CA 固有のコンフィギュレーション パラメータ、登録されている ID 証明書とのアソシエーションが含まれています。

トラストポイントの定義が完了したら、CA の指定を必要とするコマンドで、名前によってトラストポイントを参照できます。トラストポイントは複数設定できます。

（注）　FWSM に同じ CA を共有するトラストポイントが複数ある場合、CA を共有するトラストポイントのうち、ユーザ証明書の検証に使用できるのは 1 つだけです。CA を共有するどのトラストポイントを使用して、その CA が発行したユーザ証明書を検証するかを制御するには、**support-user-cert-validation** コマンドを入力します。

トラストポイントに関連するキーペアおよび発行済み証明書を PKCS12 形式でエクスポートおよびインポートできます。これは、別の FWSM でトラストポイント コンフィギュレーションを手動で複製する場合に役に立ちます。

失効チェックについて

証明書は発行されると、一定期間有効です。CA は、安全上の問題や名前またはアソシエーションの変更などの理由で、期限が切れる前に証明書を無効にすることがあります。CA は、無効になった証明書の署名付きリストを定期的に発行します。CA が認証にその証明書を使用するたびに、その証明書が無効にされていないかどうか、FWSM がチェックします。

PKI 証明書検証プロセス中に失効チェックをイネーブルにすると、FWSM は、CRL チェックまたは OCSP あるいはこれらの両方を使用して証明書失効ステータスをチェックします。設定する 2 つめの方式は、1 つめの方式によりエラーが返された場合（たとえば、サーバが使用不可能になった場合など）のみ使用されます。

CRL チェックを使用すると、FWSM によって、無効になった証明書がすべてリストされている CRL が取得、解析、およびキャッシュされます。OCSP は失効ステータスを確認する拡張性の高い方法であり、検証局で証明書ステータスをローカライズします。この検証局が特定の証明書のステータスを問い合わせます。

証明書の設定

この項では、FWSM における証明書の設定方法と、証明書の使用と管理に関するその他の手順について説明します。

証明書の準備

証明書を使用する FWSM を設定する前に、FWSM が証明書をサポートするように正しく設定されていることを確認してください。FWSM の設定に誤りがあると、登録に失敗したり、不正確な情報を含む証明書が要求されたりする可能性があります。

証明書を使用できるように FWSM を準備するには、次の手順を実行します。

ステップ 1　FWSM のホスト名とドメイン名が正しく設定されていることを確認します。現在設定されているホスト名およびドメイン名を表示するには、次のコマンドを入力します。

```
hostname(config)# show running-config
```

ホスト名を設定する方法の詳細については、「ホスト名の設定」（P.7-3）を参照してください。

ドメイン名を設定する方法の詳細については、「ドメイン名の設定」（P.7-4）を参照してください。

ステップ 2　CA を設定する前に、FWSM のクロックが正しく設定されていることを確認します。証明書には、有効になる、および失効する日付と時刻が設定されます。FWSM が CA に登録して証明書を取得するとき、FWSM は現在の時刻が証明書の有効期間の範囲内であるかどうかをチェックします。時刻がこの範囲外の場合、登録は失敗します。

クロックを設定する方法の詳細については、「hostname(config)# domain-name example.com」（P.7-4）を参照してください。

キー ペアの生成

「キー ペアについて」（P.12-2）で説明したように、キー ペアとは RSA キーのことです。使用する証明書のタイプに応じてキー ペアを生成する必要があります。

キー ペアを生成するには、次の手順を実行します。

ステップ 1　PKI 実装に必要なタイプのキー ペアを生成します。これには、必要に応じて次の手順を実行します。

a.　RSA キー ペアを生成するには、次のコマンドを入力します。

```
hostname/contexta(config)# crypto key generate rsa
```

追加のキーワードを使用しない場合、このコマンドは汎用 RSA キー ペアを 1 つ生成します。キー係数が指定されないため、デフォルトのキー係数である 1024 ビットが使用されます。その他の係数サイズを指定するには、**modulus** キーワードを使用します。

（注）　1024 ビットを超える RSA キー ペアを持つ ID 証明書を使用している複数の SSL 接続によって、FWSM での CPU 使用率が高くなり、クライアントレス ログインが拒否される可能性があります。

label キーワードを使用すると、各キー ペアにラベルを割り当てることもできます。このラベルは、キー ペアを使用するトラストポイントによって参照されます。ラベルを割り当てなかった場合、キー ペアには *Default-RSA-Key* というラベルが自動的に付けられます。

b.　ラベルを各キー ペアに割り当てるには、次のコマンドを入力します。

```
hostname/contexta (config)# crypto key generate rsa label key-pair-label
```

ステップ 2　（任意）キー ペアを表示するには、次のコマンドを入力します。

```
hostname/contexta(config)# show crypto key mypubkey
```

次に、**show crypto key mypubkey** コマンドの出力例を示します。

```
Key pair was generated at: 16:39:47 central Feb 10 2009
Key name: <Default-RSA-Key>
 Usage: General Purpose Key
 Modulus Size (bits): 1024
 Key Data:

  30819f30 0d06092a 864886f7 0d010101 05000381 8d003081 89028181 00ea51b7
  0781848f 78bccac2 4a1b5b8d 2f3e30b4 4cae9f86 f4485207 159108c9 f5e49103
  9eeb0f5d 45fd1811 3b4aafce 292b3b64 b4124a6f 7a777b08 75b88df1 8092a9f8
  5508e9e5 2c271245 7fd1c0c3 3aaf1e04 c7c4efa4 600f4c4a 6afe56ad c1d2c01c
  e08407dd 45d9e36e 8cc0bfef 14f9e6ac eca141e4 276d7358 f7f50d13 79020301 0001
Key pair was generated at: 16:34:54 central Feb 10 2005
```

ステップ 3　生成したキー ペアを保存します。これを行うには、次のコマンドを入力して、実行コンフィギュレーションを保存します。

```
hostname(config)# write memory
```

キー ペアの削除

キー ペアを削除するには、次のコマンドを入力します。

```
hostname(config)# crypto key zeroize rsa
```

次に、**crypto key zeroize rsa** コマンドの出力例を示します。

```
WARNING: All RSA keys will be removed.
WARNING: All device certs issued using these keys will also be removed.

Do you really want to remove these keys? [yes/no] y
```

AAA 認証の確立

カットスルー プロキシ認証を実行するトラフィックの AAA 認証を確立するには、次の手順を実行します。

ステップ 1　次のいずれかのコマンドを入力します。

```
hostname (config)# aaa authentication match

hostname (config)# aaa authentication include
```

aaa authentication match コマンドでは、TACACS+ または RADIUS ユーザ アカウンティング、あるいは、ホストまたは **aaa-server** コマンドで指定されたサーバで認証または許可するホストのネットワークのローカル IP アドレスを使用できます。

aaa authentication include コマンドの場合、**aaa-server** コマンドで指定されたサーバで認証または許可される TACACS+ または RADIUS ユーザ アカウンティング*だけ*を使用できます。

ステップ 2　セキュアな認証を HTTP クライアントに設定するには、次のコマンドを入力します。

```
hostname (config)# aaa authentication secure-http-client
```

コマンドの使用方法の詳細については、*『Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference』*を参照してください。

指定設定のコンフィギュレーションの確認

サードパーティ証明をインポートする前に、特定の AAA 設定、AAA サーバ、アクセス リスト、およびオプションで仮想 HTTP が設定されている必要があります。指定設定のコンフィギュレーションを確認するには、次の手順を実行します。

ステップ 1　必要なアクセス リストが設定されているか確認するには、次のコマンドを入力します。

```
hostname(config)# show run access-list
```

次に、設定されているアクセス リストを表示する **show run access-list** コマンドの出力例を示します。

```
access-list temp extended permit ip any any
```

ステップ 2　AAA サーバが設定されているか確認するには、次のコマンドを入力します。

```
hostname(config)# show run aaa-server
```

次に、設定されている AAA サーバを表示する **show run aaa-server** コマンドの出力例を示します。

```
aaa-server new protocol tacacs+
aaa-server new (outside) host 10.77.152.80
key cisco
```

ステップ 3　AAA 設定が設定されているか確認するには、次のコマンドを入力します。

```
hostname(config)# show run aaa
```

次に、設定されている AAA 設定を表示する **show run aaa** コマンドの出力例を示します。

```
aaa authentication match temp outside new
aaa authentication secure-http-client
```

ステップ 4　（任意）インポート済み証明書名およびその関連する IP アドレスを表示するには、次のコマンドを入力します。

```
hostname(config)# show run name
```

次に、インポート済み証明書名およびその関連する IP アドレスを表示する **show run name** コマンドの出力例を示します。

```
name 10.77.152.104 atl-lx-sbacchus.cisco.com
```

ステップ 5　（任意）仮想 IP アドレスを表示するには、次のコマンドを入力します。

```
hostname(config)# show running-config virtual
```

次に、仮想 IP アドレスにマッピングされているインポート済み証明書の名前を表示する **show running-config virtual** コマンドの出力例を示します。

```
virtual http atl-lx-sbacchus.cisco.com
```

キーペアおよび証明書のエクスポートおよびインポート

キーペアと、トラストポイント設定に関連付けられている発行済み証明書は、エクスポートおよびインポートできます。FWSM は、トラストポイントのエクスポートおよびインポートで PKCS12 形式をサポートします。

ここでは、次の内容について説明します。

キーペアおよび証明書のエクスポート

PKCS12 形式のトラストポイント コンフィギュレーションに関連付けられているキーペアおよび証明書をエクスポートするには、次のコマンドを入力します。

```
hostname (config)# crypto ca export pkcs12
```

この表示されたデータはコピーできます。トラストポイント データはパスワードで保護されますが、このデータをファイルに保存する場合は、そのファイルがセキュアな場所にあることを確認してください。

たとえば、cisco123 をパスフレーズとして使用する newton という名前のトラストポイントの PKCS12 データを手動でエクスポートするには、次のコマンドを入力します。

```
hostname (config)# crypto ca export newton pkcs12 cisco123

Exported pkcs12 follows:

[ PKCS12 data omitted ]

---End - This line not part of the pkcs12---
```

キーペアおよび証明書のインポート

トラストポイント コンフィギュレーションに関連付けられているキーペアおよび発行済み証明書を PKCS12 形式でインポートするには、次の手順を実行します。

ステップ 1　次のコマンドを入力します。

```
hostname (config)# crypto ca import pkcs12
```

トラストポイント コンフィギュレーション付きでインポートされるキー ペアには、作成するトラストポイントの名前と一致するラベルが割り当てられます。たとえば、エクスポートされるトラストポイントが *Default-RSA-Key* というラベルの RSA キーを使用している場合、PKCS12 形式をインポートして Main という名前のトラストポイントを作成すると、Main という名前のキー ペアが作成されます。

（注）　FWSM に同じ CA を共有するトラストポイントが複数ある場合、CA を共有するトラストポイントでユーザ証明書の検証に使用できるのは 1 つだけです。**crypto ca import pkcs12** コマンドを使用すると、このような状況を構成できます。CA を共有するどのトラストポイントを使用して、その CA が発行したユーザ証明書を検証するかを制御するには、**support-user-cert-validation** コマンドを入力します。

たとえば、パスフレーズ cisco123 を使用して PKCS12 データをトラストポイント newton に手動でインポートするには、次のコマンドを入力します。

```
hostname (config)# crypto ca import newton pkcs12 cisco123

Enter the base 64 encoded pkcs12.
End with a blank line or the word "quit" on a line by itself:
[ PKCS12 data omitted ]
quit
INFO: Import PKCS12 operation completed successfully
```

ステップ 2　インポート済み証明書を表示し、これが正しくインポートされているか確認するには、次のコマンドを入力します。

```
hostname (config)# show crypto ca certificates
```

次に、トラストポイントの名前 newton をリストする **show crypto ca certificates** コマンドの出力例を示します。

```
CA Certificate
  Status: Available
  Certificate Serial Number: 18dad19e267de8bb4a2158cdcc6b3b4a
  Certificate Usage: General Purpose
  Public Key Type: RSA (2048 bits)
  Issuer Name:
    cn=VeriSign Class 3 Public Primary Certification Authority - G5
    ou=(c) 2006 VeriSign\, Inc. - For authorized use only
    ou=VeriSign Trust Network
    o=VeriSign\, Inc.
    c=US
  Subject Name:
    cn=VeriSign Class 3 Public Primary Certification Authority - G5
    ou=(c) 2006 VeriSign\, Inc. - For authorized use only
    ou=VeriSign Trust Network
    o=VeriSign\, Inc.
    c=US
  Validity Date:
    start date: 23:00:00 IST Nov 7 2006
    end   date: 22:59:59 IST Jul 16 2036
  Associated Trustpoints: newton-1

Certificate
  Status: Available
  Certificate Serial Number: 5b178bf40eda86f320c4302cb055a743
  Certificate Usage: General Purpose
  Public Key Type: RSA (1024 bits)
  Issuer Name:
    cn=VeriSign Class 3 Extended Validation SSL CA
    ou=Terms of use at https://www.verisign.com/rpa (c)06
    ou=VeriSign Trust Network
    o=VeriSign\, Inc.
    c=US
  Subject Name:
    cn=atl-lx-sbacchus.cisco.com
    o=Cisco Systems\, Inc
    sa=170 West Tasman Dr
    l=San Jose
    st=California
    pc=95134
    c=US
    serialNumber=C1183477
    2.5.4.15=#131256312e302c20436c6175736520352e286229
    1.3.6.1.4.1.311.60.2.1.2=#130a43616c69666f726e6961
    1.3.6.1.4.1.311.60.2.1.3=#13025553
```

```
  CRL Distribution Points:
    [1]  http://EVSecure-crl.verisign.com/EVSecure2006.crl
  Validity Date:
    start date: 23:00:00 IST Sep 26 2007
    end   date: 22:59:59 IST Sep 26 2008
  Associated Trustpoints: newton

CA Certificate
  Status: Available
  Certificate Serial Number: 5b7759c61784e15ec727c0329529286b
  Certificate Usage: General Purpose
  Public Key Type: RSA (2048 bits)
  Issuer Name:
    cn=VeriSign Class 3 Public Primary Certification Authority - G5
    ou=(c) 2006 VeriSign\, Inc. - For authorized use only
    ou=VeriSign Trust Network
    o=VeriSign\, Inc.
    c=US
  Subject Name:
    cn=VeriSign Class 3 Extended Validation SSL CA
    ou=Terms of use at https://www.verisign.com/rpa (c)06
    ou=VeriSign Trust Network
    o=VeriSign\, Inc.
    c=US
  CRL Distribution Points:
    [1]  http://EVSecure-crl.verisign.com/pca3-g5.crl
  Validity Date:
    start date: 23:00:00 IST Nov 7 2006
    end   date: 22:59:59 IST Nov 7 2016
  Associated Trustpoints: newton
```

証明書のトラストポイントへのリンク

サードパーティ証明書をインポートしたら、これをトラストポイントにリンクする必要があります。リストすることで、複数のクライアントと通信できるようになります。

サードパーティ証明書のインポート中に使用した名前（newton）と同じ名前のトラストポイントの名前を表示するには、次のコマンドを入力します。

```
hostname (config)# show run ssl
ssl trust-point newton
```

設定例：カットスルー プロキシ認証

カットスルー プロキシ認証で FWSM を設定するには、次のコマンドを入力します。

```
hostname(config)# access-list FWACL extended permit tcp any any eq ftp
access-list FWACL extended permit tcp any any eq telnet
access-list FWACL extended permit tcp any any eq www
access-list FWACL extended permit tcp any any eq https
access-group FWACL in interface outside

timeout uauth 0:05:00 absolute

aaa-server TacacsServers protocol tacacs+
reactivation-mode depletion deadtime 2
```

```
aaa-server TacacsServers host 100.136.0.3
timeout 22
key secretpassword

aaa authentication match FWACL outside TacacsServers
aaa accounting match FWACL outside TacacsServers
aaa authentication secure-http-client

auth-prompt prompt (JCPIX249) Login:
auth-prompt accept (JCPIX249) Login Accepted!
auth-prompt reject (JCPIX249) Login Rejected!
```

コマンドの **access-list** シリーズは、FWSM を介して許可されるプロトコルを定義します。カットスルー プロキシ認証でサポートされているプロトコルは、この例で示すプロトコルと SSH のみです。

timeout uauth コマンドを使用すると、FWSM は、すべてのプロトコルの認証を 5 分で再要求できます。

aaa authentication コマンドは、カットスルー プロキシ認証です。このコマンドを実行すると、アクセス リストのプロトコルを照合し、代行受信して、ユーザ認証プロンプトを表示します。

コマンドの **auth-prompt** シリーズでは、ユーザに表示されるプロンプトが変更されるので、FWSM が要求を行っていることがわかります。