SSH 認証の X.509v3 証明書

• SSH 認証の X.509v3 証明書 （1 ページ）

SSH 認証の X.509v3 証明書

SSH 認証の X.509v3 証明書機能は、サーバー内で X.509v3 デジタル証明書を使用し、セキュアシェル（SSH）サーバー側でユーザー認証を使用します。

このモジュールでは、デジタル証明書用のサーバおよびユーザ証明書プロファイルを設定する方法について説明します。

SSH 認証の X.509v3 証明書の前提条件

- SSH 認証の X.509v3 証明書機能では、**ip ssh server authenticate user** コマンドの代わりに **ip ssh server algorithm authentication** コマンドが導入されます。**ip ssh server authenticate user** コマンドを使用すると、次の警告メッセージが表示されます。

```
Warning: SSH command accepted but this CLI will be deprecated soon.
Please move to new CLI “ip ssh server algorithm authentication”.
Please configure “default ip ssh server authenticate user” to make the CLI ineffective.
```

default ip ssh server authenticate user コマンドを使用して、**ip ssh server authenticate user** コマンドを無効にします。その後、IOS セキュア シェル（SSH）サーバーは **ip ssh server algorithm authentication** コマンドを使用して起動します。

SSH 認証の X.509v3 証明書の制約事項

- SSH 認証の X.509v3 証明書機能の実装は、Cisco IOS XE セキュアシェル（SSH）サーバー側にのみ適用できます。
- SSH サーバーは、サーバーおよびユーザー認証について、x509v3-ssh-rsa アルゴリズムベースの証明書のみをサポートします。

SSH 認証用の X.509v3 証明書に関する情報

次に、デジタル証明書、およびサーバーとユーザーの認証について説明します。

デジタル証明書

認証の有効性は、公開署名キーとその署名者のアイデンティティとの関連の強さに依存します。X.509v3 形式（RFC5280）のデジタル証明書は、アイデンティティの管理を実行するために使用されます。信頼できるルート証明機関とその中間証明機関による署名の連鎖によって、指定の公開署名キーと指定のデジタル アイデンティティがバインドされます。

公開キーインフラストラクチャ（PKI）のトラストポイントは、デジタル証明書の管理に役立ちます。証明書とトラストポイントを関連付けることによって、証明書を追跡できます。トラストポイントには、認証局（CA）、さまざまなアイデンティティ パラメータ、およびデジタル証明書に関する情報が含まれています。複数のトラストポイントを作成して、異なる証明書に関連付けることができます。

X.509v3 を使用したサーバーおよびユーザー認証

サーバー認証の場合、Cisco IOS XE セキュアシェル（SSH）サーバーが確認のためにそれ自体の証明書を SSH クライアントに送信します。このサーバー証明書は、サーバー証明書プロファイル（ssh-server-cert-profile-server コンフィギュレーション モード）で設定されたトラストポイントに関連付けられます。

ユーザー認証の場合、SSH クライアントが確認のためにユーザーの証明書を SSH サーバーに送信します。SSH サーバーは、サーバー証明書プロファイル（ssh-server-cert-profile-user コンフィギュレーション モード）で設定された公開キーインフラストラクチャ（PKI）トラストポイントを使用して、受信したユーザー証明書を確認します。

デフォルトでは、証明書ベースの認証が SSH サーバー端末でサーバーおよびユーザーに対して有効になります。

SSH 認証用の X.509v3 証明書の設定方法

ここでは、SSH 認証用の X.509v3 証明書の設定方法について説明します。

サーバー認証にデジタル証明書を使用するための SSH サーバーの設定

サーバー認証にデジタル証明書を使用するように SSH サーバーを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip ssh server algorithm hostkey** {**x509v3-ssh-rsa** [**ssh-rsa**] | **ssh-rsa** [**x509v3-ssh-rsa**]} 例： `Device(config)# ip ssh server algorithm hostkey x509v3-ssh-rsa`	ホスト キー アルゴリズムの順序を定義します。セキュア シェル（SSH）クライアントとネゴシエートされるのは、設定済みのアルゴリズムのみです。 （注） IOS SSH サーバーには、1 つ以上の設定済みホスト キー アルゴリズムが必要です。 • ssh-rsa：公開キーベース認証 • x509v3-ssh-rsa：証明書ベース認証
ステップ 4	**ip ssh server certificate profile** 例： `Device(config)# ip ssh server certificate profile`	サーバー証明書プロファイルおよびユーザー証明書プロファイルを設定し、SSH 証明書プロファイル コンフィギュレーション モードを開始します。
ステップ 5	**server** 例： `Device(ssh-server-cert-profile)# server`	サーバー証明書プロファイルを設定し、SSH サーバー証明書プロファイルのユーザー コンフィギュレーション モードを開始します。
ステップ 6	**trustpoint sign** *PKI-trustpoint-name* 例： `Device(ssh-server-cert-profile-server)# trustpoint sign trust1`	公開キーインフラストラクチャ（PKI）トラストポイントをサーバ証明書プロファイルにアタッチします。SSH サーバは、この PKI トラストポイントに関連付けられた証明書をサーバ認証に使用します。
ステップ 7	**ocsp-response include** 例： `Device(ssh-server-cert-profile-server)# ocsp-response include`	（任意）Online Certificate Status Protocol（OCSP）の応答または OCSP ステープリングをサーバ証明書と一緒に送信します。 （注） デフォルトではこのコマンドの **no** 形式が設定されており、OCSP 応答はサーバー証明書と一緒には送信されません。

	コマンドまたはアクション	目的
ステップ 8	**end** 例： `Device(ssh-server-cert-profile-server)# end`	SSHサーバー証明書プロファイルのサーバー コンフィギュレーション モードを終了し、特権EXECモードに戻ります。

ユーザー認証用のデジタル証明書を確認するための SSH サーバーの設定

ユーザー認証にデジタル証明書を使用するように SSH サーバーを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip ssh server algorithm authentication** {**publickey** | **keyboard** | **password**} 例： `Device(config)# ip ssh server algorithm authentication publickey`	ユーザ認証アルゴリズムの順序を定義します。セキュア シェル（SSH）クライアントとネゴシエートされるのは、設定済みのアルゴリズムのみです。 （注） • SSH サーバーには、1 つ以上の設定済みユーザー認証アルゴリズムが必要です。 • ユーザー認証に証明書方式を使用するには、**publickey** キーワードを設定する必要があります。 • **ip ssh server algorithm authentication** コマンドは **ip ssh server authenticate user** コマンドの代わりに使用します。

	コマンドまたはアクション	目的
ステップ 4	**ip ssh server algorithm publickey** {**x509v3-ssh-rsa** [**ssh-rsa**] | **ssh-rsa** [**x509v3-ssh-rsa**]} 例： `Device(config)# ip ssh server algorithm publickey x509v3-ssh-rsa`	公開キー アルゴリズムの順序を定義します。SSH クライアントによってユーザ認証に許可されるのは、設定済みのアルゴリズムのみです。 （注） SSH クライアントには、1 つ以上の設定済み公開キー アルゴリズムが必要です。 • ssh-rsa：公開キーベース認証 • x509v3-ssh-rsa：証明書ベース認証
ステップ 5	**ip ssh server certificate profile** 例： `Device(config)# ip ssh server certificate profile`	サーバー証明書プロファイルおよびユーザー証明書プロファイルを設定し、SSH 証明書プロファイル コンフィギュレーション モードを開始します。
ステップ 6	**user** 例： `Device(ssh-server-cert-profile)# user`	ユーザ証明書プロファイルを設定し、SSH サーバ証明書プロファイルのユーザ コンフィギュレーション モードを開始します。
ステップ 7	**trustpoint verify** *PKI-trustpoint-name* 例： `Device(ssh-server-cert-profile-user)# trustpoint verify trust2`	受信したユーザ証明書の確認に使用される公開キー インフラストラクチャ（PKI）トラストポイントを設定します。 （注） 同じコマンドを複数回実行することで、複数のトラストポイントを設定します。最大 10 のトラストポイントを設定できます。
ステップ 8	**ocsp-response required** 例： `Device(ssh-server-cert-profile-user)# ocsp-response required`	（任意）受信したユーザ証明書による Online Certificate Status Protocol（OCSP）の応答の有無を要求します。 （注） デフォルトではこのコマンドの **no** 形式が設定されており、ユーザー証明書は OCSP 応答なしで受け入れられます。

	コマンドまたはアクション	目的
ステップ 9	**end** 例： `Device(ssh-server-cert-profile-user)# end`	SSHサーバー証明書プロファイルのユーザー コンフィギュレーション モードを終了し、特権EXECモードに戻ります。

トラストポイント認証の設定とデバイス証明書の作成

トラストポイント認証を設定してデバイス証明書を作成するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**crypto pki trustpoint** *name* 例： `Device(config)# crypto pki trustpoint trust1`	トラストポイントおよび設定された名前を宣言して、CA トラストポイント コンフィギュレーション モードを開始します。
ステップ 4	**enrollment url** *url* 例： `Device(ca-trustpoint)# enrollment url http://10.1.1.10:80`	デバイスが証明書要求を送信する CA の URL を指定します。
ステップ 5	**revocation-check none** 例： `Device(ca-trustpoint)# revocation-check none`	証明書の確認が無視されることを指定します。
ステップ 6	**rsakeypair** *key-label* [*key-size* [*encryption-key-size*]] 例： `Device(ca-trustpoint)# rsakeypair trust1 2048`	（任意）証明書に関連付けるキーペアを指定します。 *key-label* 引数付きのキーペアがまだ存在しない、あるいは **auto-enroll regenerate** コマンドが発行された場合、登録時に key-label 引数付きのキーペアが生成されます。

	コマンドまたはアクション	目的
		キーを生成するための *key-size* 引数を指定し、*encryption-key-size* 引数を指定して、個別の暗号化、署名キー、および証明書を要求します。*key-size* 引数の範囲は 512 ～ 4096 です。key-size と encryption-key-size は同じサイズでなければなりません。2048 未満の長さを指定することは推奨されません。 （注） このコマンドがイネーブルでない場合に、FQDN キーペアが使用されます。
ステップ 7	**exit** 例： `Device(ca-trustpoint)# exit`	CA トラストポイントコンフィギュレーションモードを終了し、グローバルコンフィギュレーションモードに戻ります。
ステップ 8	**crypto pki authenticate** *name* 例： `Device(config)# crypto pki authenticate trust1`	CA 証明書を取得して、認証します。証明書フィンガープリントをチェックするよう求められた場合、証明書フィンガープリントをチェックします。 （注） CA 証明書がコンフィギュレーションにすでにロードされている場合、このコマンドはオプションです。
ステップ 9	**crypto pki enroll** *name* 例： `Device(config)# crypto pki enroll trust1`	証明書要求が証明書サーバーに送信され、サーバーが ID またはデバイス証明書を発行します。証明書要求にルータの FQDN および IP アドレスを含めるかどうかなどの登録情報を求められます。
ステップ 10	**show crypto pki certificates** 例： `Device(config)# show crypto pki certificates verbose trust1`	（任意）ロールオーバー証明書などの、証明書に関する情報を表示します。

次のタスク

他の登録オプションを使用して証明書をインストールする方法の詳細については、「PKI 内での RSA キーの展開」を参照してください。

デジタル証明書を使用したサーバーおよびユーザー認証の設定の確認

デジタル証明書を使用したサーバーおよびユーザー認証の設定を確認するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。

ステップ 2	**show ip ssh**
	例：
	`Device# show ip ssh`
	
	`SSH Enabled - version 1.99`
	`Authentication methods:publickey,keyboard-interactive,password`
	`Authentication Publickey Algorithms:x509v3-ssh-rsa,ssh-rsa`
	`Hostkey Algorithms:x509v3-ssh-rsa,ssh-rsa`
	`Authentication timeout: 120 secs;`
	`Authentication retries: 3`
	`Minimum expected Diffie Hellman key size : 1024 bits`
		現在設定されている認証方式を表示します。証明書ベース認証の使用を確認するには、x509v3-ssh-rsa アルゴリズムが設定済みのホスト キー アルゴリズムであることを確認します。

SSH 認証用の X.509v3 証明書の設定例

ここでは、デジタル証明書を使用したユーザーおよびサーバー認証の例を示します。

例：サーバー認証にデジタル証明書を使用するための SSH サーバーの設定

この例では、サーバー認証用のデジタル証明書を使用するための SSH サーバーの設定方法を示します。

```
Device> enable
Device# configure terminal
Device(config)# ip ssh server algorithm hostkey x509v3-ssh-rsa
Device(config)# ip ssh server certificate profile
Device(ssh-server-cert-profile)# server
Device(ssh-server-cert-profile-server)# trustpoint sign trust1
Device(ssh-server-cert-profile-server)# end
```

例：ユーザー認証用のデジタル証明書を確認するための SSH サーバーの設定

この例では、ユーザー認証用のユーザーのデジタル証明書を確認するための SSH サーバーの設定方法を示します。

```
Device> enable
Device# configure terminal
Device(config)# ip ssh server algorithm authentication publickey
Device(config)# ip ssh server algorithm publickey x509v3-ssh-rsa
Device(config)# ip ssh server certificate profile
Device(ssh-server-cert-profile)# user
Device(ssh-server-cert-profile-user)# trustpoint verify trust2
Device(ssh-server-cert-profile-user)# end
```

SSH 認証用の X.509v3 証明書の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Gibraltar 16.11.1	SSH 認証の X.509v3 証明書	SSH 認証の X.509v3 証明書機能は、サーバ内でX.509v3デジタル証明書を使用し、SSHサーバ側でユーザ認証を使用します。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。

SSH 認証用の X.509v3 証明書の機能履歴

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。