X.509v3 Certificates for SSH Authentication

Feature History for X.509v3 Certificates for SSH Authentication

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 26.2.1ea	X.509v3 Certificates for SSH Authentication: X.509v3 Certificates for SSH Authentication feature support has been introduced.	Cisco C9550 Series Smart Switches
Cisco IOS XE 17.18.1	X.509v3 Certificates for SSH Authentication: X.509v3 Certificates for SSH Authentication enables certificate-based server and user authentication for SSH, utilizing PKI trustpoints and X.509v3 digital certificates for enhanced identity management.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

X.509v3 Certificates for SSH Authentication

The X.509v3 Certificates for Secure Shell (SSH) Authentication feature enables the use of X.509v3 digital certificates for both server and user authentication on the SSH server. This module describes how to configure certificate profiles for these authentication types.

Features of X.509v3 Certificates for SSH Authentication

Digital Certificates

Digital certificates in the X.509v3 format (RFC5280) provide identity management. They establish a linkage between a public signing key and the signer's identity through a chain of signatures from trusted root and intermediate certification authorities (CAs). Public Key Infrastructure (PKI) trustpoints help manage these digital certificates. A trustpoint contains information about the CA, identity parameters, and the digital certificate, allowing for tracking and association with different certificates.

Server and User Authentication

By default, certificate-based authentication is enabled for both server and user on the SSH server.

- **Server Authentication**: The Cisco IOS XE SSH server sends its certificate to the SSH client for verification. This server certificate is associated with the trustpoint configured in the conf-ssh-server-cert-profile-server configuration mode.
- **User Authentication**: The SSH client sends the user's certificate to the SSH server for verification. The SSH server validates the incoming user certificate using PKI trustpoints configured in the conf-ssh-server-cert-profile-user configuration mode.

Prerequisites for X.509v3 Certificates for SSH Authentication

The **ip ssh server algorithm authentication** command replaces the deprecated **ip ssh server authenticate user** command. If you use the old command, this warning message is displayed:

```
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.
```

To make the old command ineffective and ensure the SSH server uses the new command, **default ip ssh server authenticate user**.

Restrictions for X.509v3 Certificates for SSH Authentication

- This feature is applicable only on the Cisco IOS XE Secure Shell (SSH) server side.
- The SSH server supports only the x509v3-ssh-rsa algorithm-based certificate for server and user authentication.

How to Configure X.509v3 Certificates for SSH Authentication

To configure the SSH server to use digital certificates for server authentication, perform this procedure:

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode. Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip ssh server algorithm hostkey** {**x509v3-ssh-rsa** [**ssh-rsa**] | **ssh-rsa** [**x509v3-ssh-rsa**]}

Example:

```
Device(config)# ip ssh server algorithm hostkey x509v3-ssh-rsa
```

Defines the order of host key algorithms. Only the configured algorithm is negotiated with the secure shell (SSH) client.

Note
The IOS SSH server must have at least one configured host key algorithm:

- ssh-rsa: public key based authentication
- x509v3-ssh-rsa: certificate-based authentication

Step 4 **ip ssh server certificate profile**

Example:

```
Device(config)# ip ssh server certificate profile
```

Configures server certificate profile and user certificate profile and enters SSH certificate profile configuration mode.

Step 5 **server**

Example:

```
Device(conf-ssh-server-cert-profile)# server
```

Configures server certificate profile and enters SSH server certificate profile server configuration mode.

Step 6 **trustpoint sign** *PKI-trustpoint- name*

Example:

```
Device(conf-ssh-server-cert-profile-server)#trustpoint sign trust1
```

Attaches the public key infrastructure (PKI) trustpoint to the server certificate profile. The SSH server uses the certificate associated with this PKI trustpoint for server authentication.

Step 7 **ocsp-response include**

Example:

```
Device(conf-ssh-server-cert-profile-server)#ocsp-response include
```

(Optional) Sends the Online Certificate Status Protocol (OCSP) response or OCSP stapling along with the server certificate.

Note
By default the no form of this command is configured and no OCSP response is sent along with the server certificate.

Step 8 **end**

Example:

```
Device(conf-ssh-server-cert-profile-server)# end
```

Exits SSH server certificate profile server configuration mode and returns to privileged EXEC mode.

Configuring the SSH Server to Verify Digital Certificates for User Authentication

To configure the SSH Server to use digital certificates for user authentication, perform this procedure:

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode. Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip ssh server algorithm authentication** {**publickey** | **keyboard** | **password**}

Example:

```
Device(config)# ip ssh server algorithm authentication publickey
```

Defines the order of user authentication algorithms. Only the configured algorithm is negotiated with the secure shell (SSH) client.

Note
- The SSH server must have at least one configured user authentication algorithm.

- To use the certificate method for user authentication, the **publickey** keyword must be configured.
- The **ip ssh server algorithm authentication** command replaces the **ip ssh server authenticate user** command.

Step 4 **ip ssh server algorithm publickey** {**x509v3-ssh-rsa** [**ssh-rsa**] | **ssh-rsa** [**x509v3-ssh-rsa**]}

Example:

```
Device(config)# ip ssh server algorithm publickey x509v3-ssh-rsa
```

Defines the order of public key algorithms. Only the configured algorithm is accepted by the SSH client for user authentication.

Note
The SSH client must have at least one configured public key algorithm:

- ssh-rsa: publickey- based authentication
- x509v3-ssh-rsa: certificate-based authentication

Step 5 **ip ssh server certificate profile**

Example:

```
Device(config)# ip ssh server certificate profile
```

Configures server certificate profile and user certificate profile and enters SSH certificate profile configuration mode.

Step 6 **user**

Example:

```
Device(conf-ssh-server-cert-profile)# user
```

Configures user certificate profile and enters SSH server certificate profile user configuration mode.

Step 7 **trustpoint verify** *PKI-trustpoint-name*

Example:

```
Device(conf-ssh-server-cert-profile-user)# trustpoint verify trust2
```

Configures the public key infrastructure (PKI) trustpoint that is used to verify the incoming user certificate.

Note
Configure multiple trustpoints by executing the same command multiple times. A maximum of 10 trustpoints can be configured.

Step 8 **ocsp-response required**

Example:

```
Device(conf-ssh-server-cert-profile-user)# ocsp-response required
```

(Optional) Mandates the presence of the Online Certificate Status Protocol (OCSP) response with the incoming user certificate.

Note
By default the no form of this command is configured and the user certificate is accepted without an OCSP response.

Step 9 **end**

Example:

```
Device(conf-ssh-server-cert-profile-user)# end
```

Exits SSH server certificate profile user configuration mode and returns to privileged EXEC mode.

Configuring Trustpoint Authentication and Creating Device Certificate

To configure trustpoint authentication and create device certificate, perform this procedure:

Note We recommend that you use a new RSA keypair name for the newly configured PKI certificate. If you want to reuse an existing RSA keypair name (that is associated with an old certificate) for a new PKI certificate, do either of the following:

- Do not regenerate a new RSA keypair with an existing RSA keypair name, reuse the existing RSA keypair name. Regenerating a new RSA keypair with an existing RSA keypair name will make all the certificates associated with the existing RSA keypair invalid.
- Manually remove the old PKI certificate configurations first, before reusing the existing RSA keypair name for the new PKI certificate.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode. Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **crypto pki trustpoint** *name*

Example:

```
Device(config)# crypto pki trustpoint trust1
```

Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.

Step 4 **enrollment url** *url*

Example:

```
Device(ca-trustpoint)# enrollment url http://10.1.1.10:80
```

Specifies the URL of the CA on which your device should send certificate requests.

Step 5 **revocation-check none**

Example:

```
Device(ca-trustpoint)# revocation-check none
```

Specifies that certificate checking is ignored.

Step 6 **rsakeypair** *key-label* [*keysize* [*encryption-key-size*]]

Example:

```
Device(ca-trustpoint)# rsakeypair trust1 2048
```

(Optional) Specifies which key pair to associate with the certificate. A key pair with the key-label argument will be generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command was issued.

Specify the key-size argument for generating the key, and specify the encryption-keysize argument to request separate encryption, signature keys, and certificates. The keysize argument range is from 512 to 4096. The key-size and encryption-key-size must be the same size. Length of less than 2048 is not recommended.

Note
If this command is not enabled, the FQDN key pair is used.

Step 7 **exit**

Example:

```
Device(ca-trustpoint)# exit
```

Exits ca-trustpoint configuration mode and returns to global configuration mode.

Step 8 **crypto pki authenticate** *name*

Example:

```
Device(config)# crypto pki authenticate trust1
```

Retrieves the CA certificate and authenticates it. Check the certificate fingerprint if prompted.

Note
This command is optional if the CA certificate is already loaded into the configuration.

Step 9 **crypto pki enroll** *name*

Example:

```
Device(config)# crypto pki enroll trust1
```

Certificate request is sent to the certificate server and the server issues the ID or device certificate. You are prompted for enrollment information, such as whether to include the device FQDN and IP address in the certificate request.

Step 10 **show crypto pki certificates**

Example:

```
Device# show crypto pki certificates verbose trust1
```

(Optional) Displays information about your certificates, including any rollover certificates.

What to do next

For more information on how to install the certificate using other enrollment options, see Deploying RSA Keys Within a PKI.

Verifying Configuration for Server User Authentication Using Digital Certificates

To verify configuration for server and user authentication using digital certificates, perform this procedure:

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode. Enter your password, if prompted.

Step 2 **show ip ssh**

Example:

```
Device# show ip ssh
SSH Enabled - version 2.0
Authentication
methods:publickey,keyboardinteractive,
password
Authentication Publickey
Algorithms:x509v3-ssh-rsa,sshrsa
Hostkey Algorithms:x509v3-sshrsa,
ssh-rsa
Authentication timeout: 120
secs; Authentication retries: 3
Minimum expected Diffie Hellman
key size : 1024 bits
```

Displays the currently configured authentication methods. To confirm the use of certificate-based authentication, ensure that the x509v3-ssh-rsa algorithm is the configured host key algorithm.

Configuration Examples for X.509v3 Certificates for SSH Authentication

The following section provides examples for user and server authentication using digital certificates.

This example shows how to configure the SSH Server to use digital certificates for server authentication.

```
Device> enable
Device# configure terminal
Device(config)# ip ssh server algorithm hostkey x509v3-ssh-rsa
```

```
Device(config)# ip ssh server certificate profile
Device(conf-ssh-server-cert-profile)# server
Device(conf-ssh-server-cert-profile-server)# trustpoint sign trust1
Device(conf-ssh-server-cert-profile-server)# end
```

This example shows how to configure the SSH server to verify user's digital certificate for user authentication.

```
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(conf-ssh-server-cert-profile)# user
Device(conf-ssh-server-cert-profile-user)# trustpoint verify trust2
Device(conf-ssh-server-cert-profile-user)# end
```