Implementing Secure Socket Layer

This module describes how to implement SSL.

The Secure Socket Layer (SSL) protocol and Transport Layer Security (TLS) are application-level protocols that provide for secure communication between a client and server by allowing mutual authentication, the use of hash for integrity, and encryption for privacy. SSL and TLS rely on certificates, public keys, and private keys.

Certificates are similar to digital ID cards. They prove the identity of the server to clients. Certificates are issued by certification authorities (CAs), such as VeriSign or Thawte. Each certificate includes the name of the authority that issued it, the name of the entity to which the certificate was issued, the entity's public key, and time stamps that indicate the certificate's expiration date.

Public and private keys are the ciphers used to encrypt and decrypt information. Although the public key is shared quite freely, the private key is never given out. Each public-private key pair works together: Data encrypted with the public key can be decrypted only with the private key.

Note For a complete description of the Public Key Infrastructure (PKI) commands used in this chapter, see the *Public Key Infrastructure Commands on* Cisco CRS Router *Software*module of *Cisco IOS XR System Security Command Reference for the Cisco CRS Router* . For information on SSL commands, see the *Secure Socket Layer Protocol Commands on the Cisco IOS XR Software Software* module of *Cisco IOS XR System Security Command Reference for the Cisco CRS Router* . To locate documentation of other commands that appear in this chapter, use the command reference master index, or search online.

Feature History for Implementing Secure Socket Layer

Release	Modification
Release 2.0	This feature was introduced.
Release 3.8.0	Advanced Encryption Standard (AES) encryption support was added on the SSL server.

Prerequisites for Implementing Secure Socket Layer

The following prerequisites are required to implement SSL:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- You must install and activate the Package Installation Envelope (PIE) for the security software.

  For detailed information about optional PIE installation, refer to the *Cisco IOS XR Getting Started Guide for the Cisco CRS Router*.
- Before you can begin using SSL, you must generate either Rivest, Shamir, and Adelman (RSA) or Digital Signature Algorithm (DSA) key pairs, enroll with a CA, and obtain the CA certificate for the router key.
- SSL servers support Advanced Encryption Standard (AES), which has key sizes of 128, 192, and 256 bits.

  For more information on the commands required to perform these tasks, see the **crypto key generate rsa** , **crypto key generate dsa** , **crypto ca enroll** , and **crypto ca authenticate** commands in the *Public Key Infrastructure Commands on* the Cisco IOS XR Software module of the *Cisco IOS XR System Security Command Reference for the Cisco CRS Router*.

Information About Implementing Secure Socket Layer

To implement SSL you need to understand the following concept:

Purpose of Certification Authorities

Certification Authorities (CAs) are responsible for managing certificate requests and issuing certificates to participating IPSec network devices. These services provide centralized key management for the participating devices.

CAs simplify the administration of IPSec network devices. You can use a CA with a network containing multiple IPSec-compliant devices, such as routers.

Digital signatures, enabled by public key cryptography, provide a means of digitally authenticating devices and individual users. In public key cryptography, such as the RSA encryption system, each user has a key pair containing both a public and a private key. The keys act as complements, and anything encrypted with one of the keys can be decrypted with the other. In simple terms, a signature is formed when data is encrypted with a user's private key. The receiver verifies the signature by decrypting the message with the sender's public key. The fact that the message could be decrypted using the sender's public key indicates that the holder of the private key, the sender, must have created the message. This process relies on the receiver having a copy of the sender's public key and knowing with a high degree of certainty that it does belong to the sender and not to someone pretending to be the sender.

Digital certificates provide the link. A digital certificate contains information to identify a user or device, such as the name, serial number, company, department, or IP address. It also contains a copy of the entity's public

key. The certificate is itself signed by a CA, a third party that is explicitly trusted by the receiver to validate identities and to create digital certificates.

To validate the signature of the CA, the receiver must first know the CA's public key. Normally, this process is handled out-of-band or through an operation done at installation. For instance, most web browsers are configured with the public keys of several CAs by default. Internet Key Exchange (IKE), an essential component of IPSec, can use digital signatures to scalable authenticate peer devices before setting up security associations (SAs).

Without digital signatures, a user must manually exchange either public keys or secrets between each pair of devices that use IPSec to protect communication between them. Without certificates, every new device added to the network requires a configuration change on every other device with which it communicates securely. With digital certificates, each device is enrolled with a CA. When two devices want to communicate, they exchange certificates and digitally sign data to authenticate each other. When a new device is added to the network, a user simply enrolls that device with a CA, and none of the other devices needs modification. When the new device attempts an IPSec connection, certificates are automatically exchanged and the device can be authenticated.

How to Implement Secure Socket Layer

To configure SSL so that it can be used by any application, such as HTTP server or object request broker (ORB) server, perform the task described in the following section.

Configuring Secure Socket Layer

This task explains how to configure SSL.

SUMMARY STEPS

1. **crypto key generate rsa** [**usage-keys** | **general-keys**] [*keypair-label*]
2. configure
3. **domain ipv4 hos t** *host-name v4address1* [*v4address2...v4address8*] [**unicast** | **multicast**]
4. **crypto ca trustpoint** *ca-name*
5. **enrollment url CA-URL**
6. Use one of the following commands:
   - **end**
   - **commit**
7. RP/0/RP0/CPU0:router**crypto ca authenticate** *ca-name*
8. **crypto ca enroll** *ca-name*
9. show crypto ca certificates

DETAILED STEPS

	Command or Action	Purpose

Step 1	**crypto key generate rsa** [**usage-keys** | **general-keys**] [*keypair-label*]
	
	Example:
	`RP/0/RP0/CPU0:router# crypto key generate rsa general-keys`
	`The name for the keys will be: the_default`
	`% You already have keys defined for the_default`
	`Do you really want to replace them? [yes/no]:`
		Generates RSA key pairs.
		• RSA key pairs are used to sign and encrypt Internet Key Exchange (IKE) key management messages and are required before you can obtain a certificate for your router.
		• Use the **usage-keys** keyword to specify special usage keys; use the **general-keys** keyword to specifygeneral-purpose RSA keys.
		• The *keypair-label* argument is the RSA key pair label that names the RSA key pairs.
		• To generate DSA key pairs, use the **crypto key generate dsa** command in EXEC mode.

Step 2	configure **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 3	**domain ipv4 hos t** *host-name v4address1* [*v4address2...v4address8*] [**unicast** | **multicast**] **Example:** `RP/0/RP0/CPU0:router(config)# domain ipv4 host ultra5 192.168.7.18`	Defines a static hostname-to-address mapping in the host cache using IPv4. • To define a static hostname-to-address mapping in the host cache using IPv6, use the **domain ipv6 hos t** *hostname v6address1* [*v6address2...v6address8*] [**unicast** | **multicast**] command.
Step 4	**crypto ca trustpoint** *ca-name* **Example:** `RP/0/RP0/CPU0:router(config)# crypto ca trustpoint myca`	Configures a trusted point with a selected name so that your router can verify certificates issued to peers. • Enters trustpoint configuration mode.
Step 5	**enrollment url CA-URL** **Example:** `RP/0/RP0/CPU0:router(config-trustp)# enrollment url http://ca.domain.com/certsrv/mscep/mscep.dll`	Specifies the URL of the CA. • The URL should include any nonstandard cgi-bin script location.
Step 6	Use one of the following commands: • **end** • **commit**	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`

	Command or Action	Purpose

	Example:
	`RP/0/RP0/CPU0:router(config)# end`
	or
	`RP/0/RP0/CPU0:router(config)# commit`
		◦ Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		◦ Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		◦ Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 7	RP/0/RP0/CPU0:router**crypto ca authenticate** *ca-name* **Example:** `RP/0/RP0/CPU0:router# crypto ca authenticate myca`	This command authenticates the CA to your router by obtaining the CA certificate, which contains the public key for the CA. • When prompted, type **y** to accept the certificate.
Step 8	**crypto ca enroll** *ca-name* **Example:** `RP/0/RP0/CPU0:router# crypto ca enroll myca`	Requests certificates for all of your RSA key pairs. • This command causes your router to request as many certificates as there are RSA key pairs, so you need only perform this command once, even if you have special usage RSA key pairs. • This command requires you to create a challenge password that is not saved with the configuration. This password is required if your certificate needs to be revoked, so you must remember this password. • A certificate may be issued immediately or the router sends a certificate request every minute until the enrollment retry period is reached and a timeout occurs. If a timeout occurs, contact your system administrator to get your request approved, and then enter this command again. • Verify that the certificate has been granted by using the **show crypto ca certificates** command.
Step 9	show crypto ca certificates **Example:** `RP/0/RP0/CPU0:router# show crypto ca certificates`	Displays information about your certificate and the CA certificate.

Configuration Examples for Implementing Secure Socket Layer

This section provides the following configuration example:

Configuring Secure Socket Layer: Example

The following example shows how to generate the RSA keys for the router, configure a trust point, authenticate the CA server, obtain a certificate from the CA for the key, and display information about the certificate:

```
crypto key generate rsa general-keys commit configure domain ipv4 host
          xyz-ultra5 10.0.0.5 crypto ca trustpoint myca enrollment url http://xyz-ultra5
end
          crypto ca authenticate myca crypto ca enroll myca show crypto ca certificates
```

Additional References

The following sections provide references related to implementing SSL.

Related Documents

Related Topic	Document Title
PKI commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Public Key Infrastructure Commands on Cisco IOS XR software* module in *Cisco IOS XR System Security Command Reference for the Cisco CRS Router*
SSL commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Secure Socket Layer Protocol Commands on* Cisco IOS XR softwaremodule in *Cisco IOS XR System Security Command Reference for the Cisco CRS Router*
Certification authority information	*Implementing Certification Authority Interoperability on Cisco IOS XR software* module in *Cisco IOS XR System Security Configuration Guide for the Cisco CRS Router*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC 2246	The TLS Protocol, Version 1, T. Dierks, C. Allen. January 1999.

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport