WSMA Configuration Guide, Cisco IOS Release 12.2SY

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

Configuring the Web Services Management Agent

The Web Services Management Agent (WSMA) defines a set of web services through which a network device can be managed, configuration data information can be retrieved, and new configuration data can be uploaded and manipulated. WSMA uses XML-based data encoding, that is transported by the Simple Object Access Protocol (SOAP) for the configuration data and protocol messages.

You can use WSMA over Secure Shell Version 2 (SSHv2), HTTP, or HTTPS to access the entire Cisco command-line interface (CLI). Multiple WSMA clients can connect to the WSMA server running on Cisco software.

You can also use WSMA over SSHv2, HTTP, or HTTPS to initiate secure connections from Cisco software to applications over trusted and untrusted networks.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring WSMA

- Every Web Services Management Agent (WSMA) agent must be associated with a WSMA profile to perform any operations. If WSMA agents are not properly associated with profiles, the WSMA agents cannot send or receive any messages.
- WSMA over Secure Shell Version 2 (SSHv2) requires that a vty line be available for each WSMA session.

Restrictions for Configuring WSMA

- Secure Shell Version 1 (SSHv1) is not supported; only SSHv2 is supported.
- You must be running a crypto image in order to configure SSH or HTTPS.
- Notification services are not supported for Web Services Management Agent (WSMA) over HTTP in listener mode.
- WSMA keepalive messages must be configured for Config, Exec, and Filesys services for WSMA over HTTP in initiator mode.

Information About Configuring WSMA

Configuration WSMA Service

Web Services Management Agent (WSMA) is a family of embedded agents used by the point-to-point management application to fully manage a device.

The Configuration WSMA service provides services to change the configuration on Cisco devices and validates and applies a set of configuration commands to Cisco software. Any noninteractive configuration CLI command that can be applied using the Cisco console can also be applied using this WSMA. This service is available for all configuration CLI commands on the Cisco device. It treats a set of commands as a single operation.

Three types of configuration requests can occur:

- configTest—Validates the syntax of the configuration data but does not apply the data to the running configuration.
- configApply—modifies the running configuration with the supplied configuration data. Use the action-on-fail attribute to specify the error handling to perform if an error is encountered when applying the configuration. The level of error information returned in the response can be controlled using the details attribute.
- configPersist—copies the running configuration to the startup configuration so that it persists across reloads.

The configuration WSMA service allows you to specify the CLI commands using either the XML Programmatic Interface (XML-PI) mode, or as direct CLI commands. Configuration WSMA service requests use the following modes and attributes:

- block mode—use the **<cli-config-data-block>** tag to encapsulate a multiline block of CLI commands.
- cmd mode—use the **<cli-config-data>** tag to encapsulate a block of configuration settings where each CLI line is individually delimited by **<cmd>** tags.
- XML-PI mode—use the **<xml-config-data>** tag to encapsulate processing instructions. This format is compatible with Cisco Enhanced Device Interface (EDI).
- action-on-fail—use this attribute to specify the action to perform when an error is encountered. You can specify the following action values:
  - stop—stops the execution on the first error but preserves the system state. If the execution is stopped, the configuration could be partially applied.
  - continue—ignores the error(s) and continues implementing instructions.
  - rollback—stops processing at the first error and restores the configuration to the state before any configuration was applied. The rollback action value is only enabled if the **archive** command is configured.
- details—Use this attribute to control the level of error details. You can specify one of the following values:
  - brief—provides minimal detail in error responses.
  - errors—provides details on all error encountered.
  - all—provides the maximum level of details on errors.

For more information about the request and response messages for this service, see the WSMA configuration schema at ftp://ftp.cisco.com/pub/wsma/schema/wsma_config.xsd.

EXEC WSMA Service

Web Services Management Agent (WSMA) is a family of embedded agents used by the point-to-point management application to fully manage a device.

The EXEC WSMA provides services to retrieve operational data from the Cisco device and handles EXEC mode command-line operations such as **show** commands and other diagnostic commands on Cisco devices. Interactive EXEC commands have Expect and Response tags to allow you to configure the exchange sequence. The service can retrieve **show** command operational data in XML-Progammatic Interface (PI) format and it allows remote reloading of the Cisco device.

EXEC WSMA service requests consist of a single EXEC mode command encapsulated in an **<execCLI>** tag with the following tags and attributes:

- execTest—validates the syntax of the EXEC command but does not run the command.
- maxWait—the maximum time to accumulate data and wait for the EXEC command to complete. Once the interval expires the operation stops and all accumulated data is sent in the response.

- maxResponseSize—the maximum number of bytes allowed in the body of the response. The default is 0 (infinity), and the range is 0 to $2^{31} – 1$. If the response exceeds the specified size, the operation stops and all accumulated data is sent in the response.
- format—returns the results of EXEC commands in XML-PI format. You must specify the path to the spec file on the Cisco file system. To use the global spec file command in the Cisco file system and still return XML-PI format results use the attribute *format=""*.
- xsd—If this value is set to 1 then the XML schema of the EXEC command is sent instead of the output of the EXEC command.
- cmd—this mandatory tag contains the EXEC command to run.
- dialogue—this optional tag is used only for interactive EXEC commands. It specifies an expect and reply sequence. It includes a repeat attribute that is used if there are multiple identical expect and reply sequences.
- expect—the prompt the system expects. The value need not be an exact match to the specified string. The string match has two attributes:
  - caseSensitive—set this attribute to true to do case-sensitive match. The default is to be case-sensitive.
  - match—set to leading, trailing, embedded, or exact. The defaul is an exact match.
- reply—the answer to the prompt if it matches.

The order and number of the dialoge elements must match the actual prompts seen or the EXEC call will fail. All dialogues must be run otherwise an error message is seen.

For more information on the request and response messages for this service, see the WSMA EXEC schema at: ftp://ftp.cisco.com/pub/wsma/schema/wsma_exec.xsd

Filesystem WSMA Service

Web Services Management Agent (WSMA) is a family of embedded agents used by the point-to-point management application to fully manage a device.

The Filesystem WSMA service provides services to manage files on the Cisco device. It copies and validates files between local and remote file systems. This agent can be used to list directories, upgrade the software image running on the device and delete files. File copies can be validated using a Message Digest 5 (MD5) checksum if available.

There are three types of filesystem requests:

- fileList—provides a directory listing.
- fileDelete—specifies a list of files to be deleted using the deleteFileList tag.
- fileCopy—enables the copying of files to and from the local file system. The file is copied outside of the WSMA transport mechanism using the protocol specified in the srcURL and dstURL attributes. This copy process is similar to copying a file using the EXEC CLI shell. However, this process performs additional validation checks that are not available in the EXEC shell.

The fileCopy request option has the following attributes:

- filesize—This mandatory attribute is the number of bytes to be copied. If the filesize attribute does not match the size of the copied file, the operation fails.
- erase—This mandatory attribute is a Boolean True/False. If this attribute is set to TRUE, the file system is erased before the filecopy operation is performed. This attribute is useful where the new image does not fit on the disk.
- overwrite—This mandatory attribute is a Boolean True/False. If this attribute is set to TRUE, the current file on the file system is overwritten by the new file being copied.

- retries—This optional attribute specifies the number of times the file copy is attempted in the absence of a permanent failure.
- retry-interval—This optional attribute specifies the time interval between a fileCopy retry. The default is 10 seconds.

The fileCopy request option has the following tags:

- srcURL—Specifies the URL and protocol to use for the file transfer.
- dstURL—Specifies the URL of the location to which the file is copied.
- validationInfo—Specifies an optional MD5 checksum to provide additional security during downloading.
- deletefFilelist—Specifies an optional tag which is the same as fileDelete. The fileList is deleted before starting the copy.

There are three types of filesystem responses:

- fileSystemList—This response is a listing of every disk, directory and file on the device. This response also includes additional information such as name, size and flags. You can use this information to calculate the used and free space on the device to assist image download.
- fileDeleteStatus—This response provides an itemized response to the delete file request and displays the status of each file in the list and whether or not the file was deleted.
- fileCopyStatus—This response provides a report on whether the copy succeeded or failed. Success has an empty body with the *success* attribute set to 1 if the request succeeded.

The errorInfo response is seen only if the operation fails. The errorInfo response returns an error string of the error encountered that contains two error fields; errorCode and errorMessage.

The errorCode response details the possible error types and can include:

- BAD_PARAMETER—Indicates that at least one of the parameters on the request are invalid.
- INTERNAL_ERROR—Indicates that an unknown internal API error occurred.
- NO_MEMORY—Indicates that the system has run out of memory.
- OPERATION_FAILED—Indicates that the operation did not complete. The error message provides details.
- NO_MEMORY—Indicates that memory allocation failed.
- PERMISSION_DENIED—Indicates that authorization failed.

For more information about the request and response messages for this service see the WSMA filesystem schema at ftp://ftp.cisco.com/pub/wsma/schema/wsma_filesystem.xsd

Notification WSMA Service

Web Services Management Agent (WSMA) is a family of embedded agents used by the point-to-point management application to fully manage a device.

The Notification WSMA service collects configuration-change events and forwards the details to the management application that has subscribed to get the notifications.

Multiple management applications can receive the notifications by connecting to a listener profile. Each management application must explicitly subscribe to the notifications and can turn notification on or off on the profile without affecting the operation of other connected management applications. If a connection drops notifications are turned off.

Notifications are not cached or stored. If no management application is connected when an event happens then there is no record of that event.

Notification requests have three attributes:

- correlator—used to coordinate the acknowledgment to the request.
- type—a string representing the types of notifications to enable on the session. The only supported string is configChange.
- activate—turns notification on or off by sending the value 0 (off) or 1 (on).

Notification responses have the following attributes:

- correlator—used to coordinate the acknowledgment to the request
- success—this attribute is set to True if the requested notification type is successfully enabled or disabled.

For more information about the request and response messages for this service, see the WSMA notification schema at ftp://ftp.cisco.com/pub/wsma/schema/wsma_notify.xsd

Hello WSMA Service

Web Services Management Agent (WSMA) is a family of embedded agents used by the point-to-point management application to fully manage a device.

When a new WSMA session is established, the Cisco device sends a Hello message containing the WSMA ID and a list of WSMA services available on the session. The remote management application can query this information by sending a WSMA Hello request to the Cisco device.

This service is enabled by default on every WSMA profile.

For more information about the request and response messages for this service, see the WSMA hello schema at t ftp://ftp.cisco.com/pub/wsma/schema/wsma_hello.xsd

Keepalive WSMA Service

Web Services Management Agent (WSMA) is a family of embedded agents, used by the point-to-point management application to fully manage a device.

If a WSMA profile is configured to use keepalive messages, and if no WSMA service request has been received for the configured keepalive interval, the Cisco device sends a Keepalive request on the WSMA session. If the number of keepalive requests sent exceeds the configured retries, the WSMA session is closed.

A keepalive request has one attribute, correlator. The correlator attribute is a number that starts at 1 and increments each time a keepalive request is sent on a session. The correlator value used in a keepalive response must match the value in a keepalive request.

For more information about the request and response messages for this service, see the WSMA keepalive schema at ftp://ftp.cisco.com/pub/wsma/schema/wsma_keepalive.xsd

WSMA Profiles

Web Services Management Agent (WSMA) needs input from external management applications to cause actions on the device. A physical transport protocol must be configured and associated to a WSMA to allow the WSMA to communication with external management applications. The transport protocol and an encapsulation together form a WSMA profile. Any WSMA agent must be associated with a specific WSMA profile to perform valid operations. WSMA profiles demultiplex requests to the appropriate WSMA.

WSMA profiles work as a transport termination point and allow transport and XML encapsulation parameters to be configured:

- The configurable encapsulations for WSMA are Simple Object Access Protocol (SOAP)1.1 and SOAP 1.2.
- The transportation mechanisms for WSMA include Secure Shell (SSH), HTTP, and Secure HTTP (HTTPS). This mechanism opens listening sockets for listeners on the device or connecting sockets for clients on the device.

Service Listener

The service listener is a type of Web Services Managment Agent (WSMA) profile that listens for incoming connections and accepts devices from allowed addresses or accepted user IDs. The accepted addresses are configured by defining an access list.

Accepted user IDs are configured by defining the transport method that the service listener listens for. The transport method (Secure Shell (SSH) HTTP, or HTTPS) enforces the specific user ID that is accepted.

Note WSMA listener profiles cannot access Cisco devices that are located behind a firewall.

Service Initiator

The service initiator is a type of Web Services Management Agent (WSMA) profile that initiates secure connections from Cisco devices to management applications over trusted and untrusted networks.

The service initiator creates a dynamic socket that attempts to stay connected to a configured server address. Each initiator can be configured with retry, keepalive, timeout, and reconnect settings. In addition, each initiator can specify a backup connection to use if the primary connection fails.

The service initiator allows WSMA to initiate connections to devices behind a firewall or Network Address Translation(NAT), and in Zero Touch Deployment (ZTD) networks.

SOAP

Simple Object Access Protocol (SOAP) is an industry-standard protocol to exchange XML data between applications. It defines a common mechanism to handle corrupted XML messages. It has a header mechanism to collate metadata associated with a transaction.

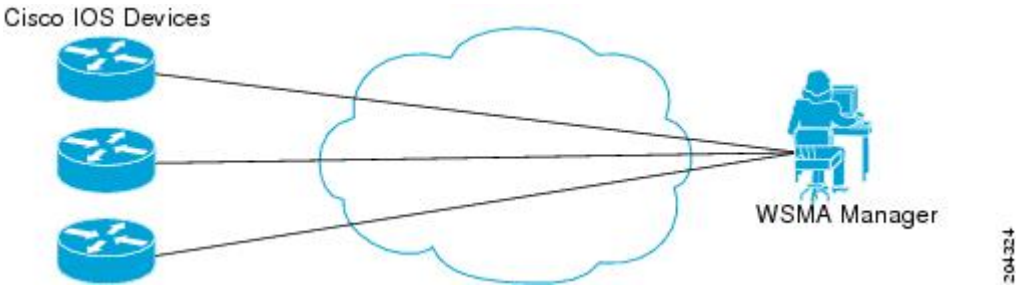
SOAP 1.1 and SOAP 1.2 have different schema definitions. They can coexist with no impact on the other. Cisco software has both SOAP 1.1 and SOAP 1.2 libraries. SOAP has mechanisms to handle XML framing and operational errors in a generic manner, allowing greater interoperability of XML-based applications.

WSMA over SSHv2

To run the WSMA over SSHv2 feature, the Web Services Management Agent (WSMA) agent must be configured to use a service profile that is using Secure Shell (SSH) as a transport method. The figure below shows a basic WSMA over SSHv2 network configuration. The client and server exchange keys for security and password encryption. The user ID and password of the SSHv2 session running WSMA are used for authorization and authentication purposes. The user privilege level is enforced and the client session may not have full access to the WSMA operations if the privilege level is not high enough. If authentication, authorization, and accounting (AAA) is configured, the AAA service is used as if a user had established an SSH session directly to the device. Using the existing security configuration makes the transition to WSMA almost seamless. Once the client has been successfully authenticated, the client invokes the SSH

connection protocol and the SSH session is established. After the SSH session is established, the user or application invokes WSMA as an SSH subsystem. The default name for the subsystem is “wsma.”

Figure 1 WSMA over SSHv2



SSHv2

SSHv2 runs on top of a reliable transport layer and provides strong authentication and encryption capabilities. SSHv2 provides a means to securely access and securely execute commands on another computer over a network.

Service listeners do not support SSHv1. The configuration for the SSHv2 server is similar to the configuration for SSHv1. Use the **ip ssh version** command to specify which version of SSH you want to configure. If you do not configure this command, SSH by default runs in compatibility mode; that is, both SSHv1 and SSHv2 connections are honored.

Note SSHv1 is a protocol that has never been defined in a standard. If you do not want your device to fall back to the undefined protocol (version 1), you should use the**ip ssh version** command and specify version 2.

Use the **ip ssh rsa keypair-name** command to enable an SSH connection using Rivest, Shamir, and Adelman (RSA) keys that you have configured. If you configure the **ip ssh rsa keypair-name** command with a key-pair name, SSH is enabled if the key pair exists, or SSH will be enabled if the key pair is generated later. If you use this command to enable SSH, you need not configure a hostname and a domain name.

WSMA over HTTP

To run the WSMA over HTTP feature, you must configure the Web Services Management Agent (WSMA) agent to use a service profile which is using either HTTP or Hypertext Transfer Protocol Secure (HTTPS) as a transport. For HTTPS, the client and server exchange keys for security and password encryption. The user ID and password of the HTTP or HTTPS session running WSMA are used for authorization and authentication purposes. The user privilege level is enforced and the client session may not have full access to the WSMA operations if the privilege level is not high enough. If Authentication, Authorization and Accounting (AAA) is configured, the AAA service is used as if a user had established an SSH session directly to the device. Using the existing security configuration makes the transition to WSMA almost seamless. After the HTTP or HTTPS session is established, the user or application invokes WSMA as an HTTP path. The default name for the path is “/wsma.”

When you use HTTP as the transport for a initiator profile, the WSMA Notification service is available without additional configuration. However, to use the Config, Exec, and Filesys services, you must first configure keepalive messages on the initiator profile. When keepalive messages are configured, the Cisco

device can periodically send a request to the remote WSMA application, which allows the remote HTTP server the opportunity to send a WSMA request.

When using HTTP as the transport for a listener profile, the WSMA Notification service is not supported since the Cisco device acting as a HTTP server cannot send HTTP requests, it can only respond to HTTP requests.

HTTP

HTTP is a reliable request/response protocol that runs on top of a reliable transport layer. HTTPS provides strong authentication and encryption capabilities.

HTTP is configured with the **ip http server** command and HTTPS is configured using the **ip http secure-server** command.

Access Lists

You can configure access lists for use with a service listener. An access list is a sequential collection of permit and deny conditions that applies to IP addresses. The Cisco software tests addresses against the conditions in an access list one by one. The first match determines whether the software accepts or rejects the address. Because the software stops testing conditions after the first match, the order of the conditions is critical. If no conditions match, the software rejects the address.

The two main tasks involved in using access lists are as follows:

1. Creating an access list by specifying an access list number or name and access conditions.
2. Applying the access list to interfaces or terminal lines.

WSMA ID

The Web Services Management Agent (WSMA) IDs allow Cisco networking devices to have unique IDs. Unique IDs are important in a Network Address Translation (NAT) or Dynamic Host Configuration Protocol (DHCP) network where all the device IP addresses are locally significant. In this type of deployment, the WSMA ID can be used to give each device a globally unique ID.

The WSMA ID can be explicitly configured based on other properties of the device such as:

- The hardware serial number
- The hostname
- The IP address of an interface
- The MAC address of an interface
- A user-defined string

Whenever the WSMA ID changes, all WSMA sessions are disconnected. This is to protect the management applications from synchronizing the state dynamically.

WSMA Security

Web Services Management Agent (WSMA) security is integrated with authentication, authorization, and accounting (AAA) configuration of Cisco software. The AAA associations configured on the transport layer are used by WSMA.

WSMA is designed for point-to-point operation and works over an encrypted transport. The security on the transport layer identifies and authenticates the users.

WSSE

The Web Services Security Header (WSSE) is the Simple Object Access Protocol (SOAP) security extension.

The WSMA profiles can be configured to expect or ignore additional security headers in the SOAP messages depending on the deployment mode. If WSMA is configured to contain a security header, the format of the header is as per the SOAP security extension, WSSE.

SOAP enforces authentication using the WSSE header. Any authentication errors are reported as SOAP faults. The authenticated message is passed on to the WSMA, which checks for the authorization level of the user before applying any operation. Authorization errors are reported as a WSMA error response.

If WSMA profiles are configured without the WSSE, then the security header is ignored and the transport login credentials are used for authentication. If the WSSE is expected, then the details of the security header are used to authenticate the user. If the security header is missing, the incoming message is discarded and a SOAP fault is issued.

WSMA Schema

Each Web Services Management Agent (WSMA) service publishes its XML schema. The schema describe the XML messages that the specific WSMA service can understand and execute. The WSMA schema define the entire data required to execute an operation and ensure operations can be performed identically regardless of the type of transport used to carry the message.

A full list of WSMA schema (XSD) files is available from the ftp://ftp.cisco.com/pub/wsma/schema/ FTP site.

How to Configure WSMA

Enabling SSHv2 Using a Hostname and Domain Name

Perform this task to enable Secure Shell Version 2 (SSHv2) on your device using a hostname and domain name.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hostname** *hostname*
4. **ip domain-name** *name*
5. **crypto key generate rsa**
6. **ip ssh** [**timeout** *seconds* | **authentication-retries** *integer*]
7. **ip ssh version 2**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
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	**hostname** *hostname* **Example:** `Device(config)# hostname host1`	Configures a hostname for your device.
Step 4	**ip domain-name** *name* **Example:** `Device(config)# ip domain-name example.com`	Configures a domain name for your device.
Step 5	**crypto key generate rsa** **Example:** `Device(config)# crypto key generate rsa`	Enables the SSH server for local and remote authentication.

	Command or Action	Purpose
Step 6	**ip ssh** [**timeout** *seconds* | **authentication-retries** *integer*] **Example:** `Device(config)# ip ssh timeout 120`	(Optional) Configures SSH control variables on your device.
Step 7	**ip ssh version 2** **Example:** `Device(config)# ip ssh version 2`	Specifies the version of SSH to be run on your device.
Step 8	**end** **Example:** `Device(config)# end`	Exits global configuration mode.

Enabling the HTTP Server

Perform this task to enable the HTTP server. The HTTP server is disabled by default. Once the HTTP server is enabled, you can configure optional server characteristics.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http server**
4. **ip http authentication** {**aaa** | **local**}

DETAILED STEPS

	Command or Action	Purpose
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.

	Command or Action	Purpose
Step 3	**ip http server** **Example:** `Device(config)# ip http server`	Enables the HTTP 1.1 server, including the Cisco web browser user interface. **Note** If you are enabling the HTTP over Secure Socket Layer (HTTPS) server using the **ip http secure-server** command, you should disable the standard HTTP server using the **no ip http server** command. This command is required to ensure only secure connections to the server.
Step 4	**ip http authentication** {**aaa** | **local**} **Example:** `Device(config)# ip http authentication aaa`	Specifies the authentication method for HTTP server users. • The **ip http authentication enable** command specifies that the enable password is used for authentication. This authentication method cannot be used to access the WSMA.

Enabling the HTTPS Server

To disable the standard HTTP server and configure the HTTPS server with Secure Socket Layer (SSL) version 3.0, complete this task.

If a certificate authority is to be used for certification, you should declare the certificate authority (CA) trustpoint on the routing device before enabling the secure HTTP server.

SUMMARY STEPS

1. **enable**
2. **show ip http server status**
3. **configure terminal**
4. **no ip http server**
5. **ip http secure-server**
6. **ip http secure-port** *port-number*
7. **ip http secure-ciphersuite** [**3des-ede-cbc-sha**] [**rc4-128-sha**] [**rc4-128-md5**] [**des-cbc-sha**]
8. **ip http secure-client-auth**
9. **ip http secure-trustpoint** *name*
10. **end**
11. **show ip http server secure status**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip http server status** **Example:** `Device# show ip http server status`	(Optional) Displays the status of the HTTP server. • If you are unsure whether the secure HTTP server is supported in the software image you are running, enter this command and look for the line "HTTP secure server capability: {Present | Not present}". • This command displays the status of the standard HTTP server (enabled or disabled).
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**no ip http server** **Example:** `Device(config)# no ip http server`	Disables the standard HTTP server. **Note** When enabling the HTTPS server you should always disable the standard HTTP server to prevent insecure connections to the same services. This is a precautionary step (typically, the HTTP server is disabled by default).
Step 5	**ip http secure-server** **Example:** `Device(config)# ip http secure-server`	Enables the HTTPS server.
Step 6	**ip http secure-port** *port-number* **Example:** `Device(config)# ip http secure-port 1025`	(Optional) Specifies the port number that should be used for the HTTPS server. The default port number is 443. • Valid options are 443 or any number in the range 1025 to 65535.

	Command or Action	Purpose
Step 7	**ip http secure-ciphersuite** [**3des-ede-cbc-sha**] [**rc4-128-sha**] [**rc4-128-md5**] [**des-cbc-sha**] **Example:** `Device(config)# ip http secure-ciphersuite rc4-128-sha rc4-128-md5`	(Optional) Specifies the CipherSuites (encryption algorithms) that should be used for encryption over the HTTPS connection. • This command allows you to restrict the list of CipherSuites that the server offers the connecting clients. For example, you may want to allow only the most secure CipherSuite to be used. • Unless you have a reason to specify the CipherSuites that should be used, or you are unfamiliar with the details of these CipherSuites, you should leave this command unconfigured and let the server and client negotiate the CipherSuite that they both support (this is the default).
Step 8	**ip http secure-client-auth** **Example:** `Device(config)# ip http secure-client-auth`	(Optional) Configures the HTTP server to request an X.509v3 certificate from the client in order to authenticate the client during the connection process. • In the default connection and authentication process, the client requests a certificate from the HTTP server, but the server does not attempt to authenticate the client. Authenticating the client provides more security than server authentication by itself, but not all clients may be configured for CA authentication.
Step 9	**ip http secure-trustpoint** *name* **Example:** `Device(config)# ip http secure-trustpoint trustpoint-01`	Specifies the CA trustpoint that should be used to obtain an X.509v3 security certificate and to authenticate the connecting client's certificate. • Use of this command assumes you have already declared a CA trustpoint using the **crypto pki trustpoint** command and associated submode commands. • Use the same trustpoint name that you used in the associated **crypto pki trustpoint** command.
Step 10	**end** **Example:** `Device(config)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 11	**show ip http server secure status** **Example:** `Device# show ip http server secure status`	Displays the status of the HTTP secure server configuration.

Verifying the Status of the SSH Connection

To display the status of the Secure Shell (SSH) connection on your device, use the **show ssh** and **show ip ssh** commands.

The **ip ssh version** command can be used for troubleshooting your SSH configuration. By changing versions, you can determine which SSH version has a problem.

SUMMARY STEPS

1. **enable**
2. **show ssh**
3. **show ip ssh**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ssh** **Example:** `Device# show ssh`	Displays the status of SSH server connections.
Step 3	**show ip ssh** **Example:** `Device# show ip ssh`	Displays the version and configuration data for SSH.

Examples

The following sample output from the **show ssh** command displays status about SSHv2 connections:

```
Device# show ssh

Connection Version Mode Encryption  Hmac                State
Username
1          2.0     IN   aes128-cbc  hmac-md5     Session started       lab
1          2.0     OUT  aes128-cbc  hmac-md5     Session started       lab
%No SSHv1 server connections running.
```

The following sample output from the **show ip ssh** command displays the version of SSH that is enabled, the authentication timeout values, and the number of authentication retries:

```
Device# show ip ssh

SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
```

Enabling a WSMA Service Initiator

If you configure service initiator over HTTP or Secure HTTP (HTTPS), you must configure keepalive settings so that the Cisco device can periodically send a HTTP Request to the remote Web Services Management Agent (WSMA) application, thus giving the remote WSMA application a chance to send WSMA requests.

HTTPS

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **wsma profile initiator** *profile-name*
4. **encap** {**soap11** | **soap12**}
5. {[**backup**] **transport** {**http** | **https** | **ssh** *remote-host* [*initiator-port-number*] **path** *path-name* [**user** *username* [**0** | **6**] *password*] } | [**source** *source-interface*]}
6. **keepalive** *interval* [**retries** *number*]
7. **idle-timeout** *minutes*
8. **max-message** *message-size*
9. **backup hold** *minutes*
10. **backup excluded** *seconds*
11. **reconnect** *seconds*
12. **stealth**
13. **wsse**
14. **end**

DETAILED STEPS

	Command or Action	Purpose
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	**wsma profile initiator** *profile-name* **Example:** `Device(config)# wsma profile initiator prof1`	Creates a service initiator and enters WSMA initiator configuration mode.
Step 4	**encap** {**soap11**	**soap12**} **Example:** `Device(config-wsma-init)# encap soap12`	(Optional) Configures an encapsulation for the service listener profile.

	Command or Action	Purpose
Step 5	{[**backup**] **transport** {**http** | **https** | **ssh** *remote-host* [*initiator-port-number*] **path** *path-name* [**user** *username* [**0** | **6**] *password*] } | [**source** *source-interface*]} **Example:** `Device(config-wsma-init)# transport ssh sshserver path /mypath/bin/mywsma-app.sh user user1 6 encrypted-password`	Defines a transport configuration for the WSMA profile.
Step 6	**keepalive** *interval* [**retries** *number*] **Example:** `Device(config-wsma-init)# keepalive 100 retries 10`	(Optional) Enables keepalive messages and configures interval and retry values for a WSMA profile. • To ensure that the Cisco device allows the remote WSMA application to send WSMA requests, keepalive messages must be enabled on HTTP and HTTPS initiator connections.
Step 7	**idle-timeout** *minutes* **Example:** `Device(config-wsma-init)# idle-timeout 345`	(Optional) Specifies the amount of time (in minutes) to keep the session alive in the absence of any data traffic.
Step 8	**max-message** *message-size* **Example:** `Device(config-wsma-init)# max-message 290`	(Optional) Specifies the maximum receive message size (from 1 to 2000 kilobytes).
Step 9	**backup hold** *minutes* **Example:** `Device(config-wsma-init)# backup hold 233`	(Optional) Sets the time (in minutes) that the WSMA profile remains connected to the backup transport configuration.
Step 10	**backup excluded** *seconds* **Example:** `Device(config-wsma-init)# backup excluded 30`	(Optional) Sets the time (in seconds) that the WSMA profile must wait before attempting to connect to the backup transport configuration after a connection is lost.

	Command or Action	Purpose
Step 11	**reconnect** *seconds* **Example:** `Device(config-wsma-init)# reconnect 434`	(Optional) Specifies the time for the WSMA initiator profile to wait before attempting to reconnect a session.
Step 12	**stealth** **Example:** `Device(config-wsma-init)# stealth`	(Optional) Configures the service to not send Simple Object Access Protocol (SOAP) fault messages in response to corrupted XML messages.
Step 13	**wsse** **Example:** `Device(config-wsma-init)# wsse`	(Optional) Enables the Web Services Security Header (WSSE) for a WSMA profile. • By default, the WSSE is enabled. Enter the **no wsse** command to disable the WSSE.
Step 14	**end** **Example:** `Device(config-wsma-init)# end`	Ends the current configuration session and returns you to privileged EXEC mode.

Enabling a WSMA Service Listener

Before you configure service listener over SSH, you must first configure SSH. For more information, see Enabling SSHv2 Using a Hostname and Domain Name, page 10.

Before you configure service listener over HTTP, you must first configure HTTP. For more information, see the Enabling the HTTP Server, page 12 section and Enabling the HTTPS Server, page 13 section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **wsma profile listener** *profile-name*
4. **encap** {**soap11** | **soap12**}
5. **transport** {**http** | **https** [**path** *path-name*] | **ssh** [**subsys** *subsys-name*]}
6. **idle-timeout** *minutes*
7. **max-message** *message-size*
8. **keepalive** *interval* [**retries** *number*]
9. **acl** *acl-number*
10. **stealth**
11. **wsse**
12. **end**

DETAILED STEPS

	Command or Action	Purpose
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	**wsma profile listener** *profile-name* **Example:** `Device(config)# wsma profile listener prof1`	Creates a service listener and enters WSMA listener configuration mode.
Step 4	**encap** {**soap11**	**soap12**} **Example:** `Device(config-wsma-listen)# encap soap12`	(Optional) Configures an encapsulation for the service listener profile.

	Command or Action	Purpose
Step 5	**transport** {**http** | **https** [**path** *path-name*] | **ssh** [**subsys** *subsys-name*]} **Example:** `Device(config-wsma-listen)# transport ssh subsys wsma`	Defines a transport configuration for the Web Services Management Agent (WSMA) profile.
Step 6	**idle-timeout** *minutes* **Example:** `Device(config-wsma-listen)# idle-timeout 345`	(Optional) Specifies the amount of time (in minutes) to keep the session alive in the absence of any data traffic.
Step 7	**max-message** *message-size* **Example:** `Device(config-wsma-listen)# max-message 290`	(Optional) Specifies the maximum receive message size (from 1 to 2000 kilobytes).
Step 8	**keepalive** *interval* [**retries** *number*] **Example:** `Device(config-wsma-listen)# keepalive 100 retries 10`	(Optional) Enables keepalive messages and configures interval and retry values for a WSMA profile. • Keepalive messages are not sent on HTTP or HTTPS listener connections.
Step 9	**acl** *acl-number* **Example:** `Device(config-wsma-listen)# acl 34`	(Optional) Defines the access control list (ACL) group to use.
Step 10	**stealth** **Example:** `Device(config-wsma-listen)# stealth`	(Optional) Configures the service to not send Simple Object Access Protocol (SOAP) fault messages in response to corrupted XML messages.
Step 11	**wsse** **Example:** `Device(config-wsma-listen)# wsse`	(Optional) Enables the Web Services Security Header (WSSE) for a WSMA profile. • By default, WSSE is enabled. Enter the **no wsse** command to disable the WSSE.

	Command or Action	Purpose
Step 12	**end** **Example:** `Device(config-wsma-listen)# end`	Ends the current configuration session and returns you to privileged EXEC mode.

Enabling WSMA Services

Perform this task to enable a specific Web Services Management Agent (WSMA) service and associate it with a profile.

A WSMA initiator or listener profile must be configured and enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **wsma agent** {**config** | **exec** | **filesys** | **notify**} **profile** *profile-name*

DETAILED STEPS

	Command or Action	Purpose
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	**wsma agent** {**config**	**exec**	**filesys**	**notify**} **profile** *profile-name* **Example:** `Device(config)# wsma agent config profile prof1`	Enables the WSMA and associates it with a profile.

Assigning WSMA IDs

Perform this task to assign unique Web Services Management Agent (WSMA) IDs to Cisco networking devices.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **wsma id** {**hardware-serial** | **hostname** | **ip-address** *interface-type* | **mac-address** *interface-type* | **string** *value*}

DETAILED STEPS

	Command or Action	Purpose
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	**wsma id** {**hardware-serial**	**hostname**	**ip-address** *interface-type*	**mac-address** *interface-type*	**string** *value*} **Example:** `Device(config)# wsma id ip-address fastethernet 0/1`	Assigns unique WSMA IDs to Cisco networking devices.

Monitoring and Maintaining WSMA Services

SUMMARY STEPS

1. **enable**
2. **show wsma agent** {**counters** | **schema**} [**config** | **exec** | **filesys** | **notify**]
3. **debug wsma agent** [**config** | **exec** | **filesys** | **notify**]
4. **clear wsma agent** [**config** | **exec** | **filesys** | **notify**] **counters**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show wsma agent** {**counters** | **schema**} [**config** | **exec** | **filesys** | **notify**] **Example:** `Device# show wsma agent config counters`	Displays the specified statistics counters, or schema for the Web Services Management Agent (WSMA).
Step 3	**debug wsma agent** [**config** | **exec** | **filesys** | **notify**] **Example:** `Device# debug wsma agent config`	Enables debugging of the WSMA.
Step 4	**clear wsma agent** [**config** | **exec** | **filesys** | **notify**] **counters** **Example:** `Device# clear wsma agent filesys counters`	Clears WSMA statistics counters.

Examples

The following example shows how to display the WSMA configuration agent counters. The counters return the following information:

- messages received—The total number of messages that were passed from the service profile into the WSMA.
- replies sent—The total number of reply messages sent to the services profile.
- faults—The number of faults that prevented a received message producing a reply.
- notifications—The total number of notification messages sent to the services profile.

```
Device# show wsma agent counters

WSMA Exec Agent Statistics:
messages received 0, replies sent 0, faults 0
WSMA Config Agent Statistics:
messages received 4, replies sent 4, faults 0
WSMA Filesys Agent Statistics:
messages received 1, replies sent 1, faults 0
WSMA Notification Agent Statistics:
config silent
messages received 0, replies sent 0, notifications sent 0, faults 0
```

The following example shows how to display the WSMA configuration schema:

```
Device#show wsma agent config schema
```

```
New Name Space 'urn:cisco:wsma-config'
<VirtualRootTag> [0, 1] required
  <WSMA-Config> [0, 1] required
    <request> 1 required
      <config-data> 1 required
        <cli-config-data> [0, 1] required
          <cmd> 1+ required
        <cli-config-data-block> [0, 1] required
        <xml-config-data> [0, 1] required
          <Device-Configuration> [0, 1] required
            <> any subtree is allowed
```

Monitoring and Maintaining WSMA Profiles

Perform this task to monitor and maintain Web Services Management Agent (WSMA) profiles.

SUMMARY STEPS

1. **enable**
2. **show wsma profile** {**connections** | **counters** | **schema**} [**name** *profile-name*]
3. **debug wsma profile** [**listener** | **initiator**]
4. **clear wsma profile** [*profile-name*] {**connections** | **counters**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show wsma profile** {**connections**	**counters**	**schema**} [**name** *profile-name*] **Example:** `Device# show wsma profile connections`	Displays the specified service profile connections, statistics counters, or schema.
Step 3	**debug wsma profile** [**listener**	**initiator**] **Example:** `Device# debug wsma profile listener`	Enables debugging of WSMA profiles.
Step 4	**clear wsma profile** [*profile-name*] {**connections**	**counters**} **Example:** `Device# clear wsma profile prof1 counters`	Clears WSMA profile sessions or statistic counters.

Delivering WSMA Payloads

An XML payload is typically wrapped in a Simple Object Access Protocol (SOAP) message for data transportation. Without a correct design of SOAP messages, an XML payload may not be exchanged properly even if the payload follows common XML schema. The XML payload over all transports is identical. Web Services Management Agent (WSMA) supports both SOAP1.1 and SOAP1.2. The SOAP header supports two modes of security, no wsse and wsse.

Use the following XML schema to deliver WSMA payloads:

WSMA EXEC Request: Ping

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <SOAP:Body>
            <request xmlns="urn:cisco:wsma-exec" correlator="01">
                <execCLI>
                    <cmd>ping oz-dirt</cmd>
                </execCLI>
            </request>
    </SOAP:Body>
</SOAP:Envelope>]]>]]>
```

WSMA EXEC Response: Ping

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <SOAP:Body>
            <response xmlns="urn:cisco:wsma-exec" correlator="01" success="1">
                <execLog>
                    <dialogueLog>
                        <sent>ping oz-dirt</sent>
                        <received>Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.1.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms</received>
                    </dialogueLog>
                </execLog>
            </response>
    </SOAP:Body>
</SOAP:Envelope>]]>]]>
```

WSMA Config Request: CMD Data Model

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <SOAP:Body>
            <request xmlns="urn:cisco:wsma-config" correlator="4.1">
                <configApply details="all">
                    <config-data>
                        <cli-config-data>
                            <cmd>no cns config partial mixy</cmd>
                            <cmd>no stupid</cmd>
                            <cmd>no cns exec 80 </cmd>
                        </cli-config-data>
```

```
                </config-data>
            </configApply>
        </request>
    </SOAP:Body>
</SOAP:Envelope>]]>]]>
```

WSMA Config Response: CMD Data Model

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <SOAP:Body>
        <response xmlns="urn:cisco:wsma-config" correlator="4.1" success="1">
            <resultEntry lineNumber="1" cliString="no cns config partial mixy">
                <success change="NO_CHANGE" mode="IMMEDIATE" />
            </resultEntry>
      <resultEntry lineNumber="2" cliString="no stupid">
                <failure errorType="TEMPORARY" errorCode="PARSE_ERROR_NOMATCH" />
            </resultEntry>
            <resultEntry lineNumber="3" cliString="no cns exec 80 ">
                <success change="NO_CHANGE" mode="IMMEDIATE" />
            </resultEntry>
        </response>
    </SOAP:Body>
</SOAP:Envelope>]]>]]>
```

WSMA Config Request: Block Data Model

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <SOAP:Body>
        <request xmlns="urn:cisco:wsma-config" correlator="5.1">
            <configApply details="all">
                <config-data>
                    <cli-config-data-block>no cns config partial mixy
no stupid
no cns exec 80</cli-config-data-block>
                </config-data>
            </configApply>
        </request>
    </SOAP:Body>
</SOAP:Envelope>]]>]]>
```

WSMA Config Response: Block Data Model

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <SOAP:Body>
        <response xmlns="urn:cisco:wsma-config" correlator="5.1" success="1">
            <resultEntry lineNumber="1" cliString="no cns config partial mixy">
                <success change="NO_CHANGE" mode="IMMEDIATE" />
            </resultEntry>
            <resultEntry lineNumber="2" cliString="no stupid">
                <failure errorType="TEMPORARY" errorCode="PARSE_ERROR_NOMATCH" />
            </resultEntry>
            <resultEntry lineNumber="3" cliString="no cns exec 80">
                <success change="NO_CHANGE" mode="IMMEDIATE" />
            </resultEntry>
        </response>
    </SOAP:Body>
</SOAP:Envelope>]]>]]>
```

WSMA Config Request: Enhanced Device Interface (EDI) Data Model

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <SOAP:Body>
            <request xmlns="urn:cisco:wsma-config" correlator="6.1">
            <configApply details="all">
                <config-data>
                    <xml-config-data>
                        <Device-Configuration><cns operation="delete" > <config><partial>
<HostNameAddressConfigurationServer>mixy</HostNameAddressConfigurationServer>
<PortNumberConfigServiceDefault80>80</PortNumberConfigServiceDefault80></partial></
config></cns>
<stupid operation="delete" /><cns operation="delete" ><exec><P>80</P></exec></cns> </
Device-Configuration>
                    </xml-config-data>
                </config-data>
            </configApply>
            </request>
    </SOAP:Body>
</SOAP:Envelope>]]>]]>
```

WSMA Config Response: EDI Data Model

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <SOAP:Body>
            <response xmlns="urn:cisco:wsma-config" correlator="6.1" success="1">
                <resultEntry lineNumber="1" cliString="no cns config partial mixy 80">
                    <success change="NO_CHANGE" mode="IMMEDIATE" />
                </resultEntry>
                <resultEntry lineNumber="2" cliString="no stupid">
                    <failure errorType="TEMPORARY" errorCode="PARSE_ERROR_NOMATCH" />
                </resultEntry>
                <resultEntry lineNumber="3" cliString="no cns exec 80">
                    <success change="NO_CHANGE" mode="IMMEDIATE" />
                </resultEntry>
            </response>
    </SOAP:Body>
</SOAP:Envelope>]]>]]>
```

WSMA File List Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <SOAP:Body>
        <request xmlns="urn:cisco:wsma-filesystem" correlator="2"><fileList/></request>
    </SOAP:Body>
</SOAP:Envelope>]]>]]>
```

WSMA File List Response

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <SOAP:Body>
        <response xmlns="urn:cisco:wsma-filesystem" correlator="2" success="1">
```

```
            <fileSystemList>
                <fileSystem name="nvram" type="nvram" size="522232" freespace="516471"
readable="true" writeable="true">
                    <directory name="/" fullName="nvram:/" readFlag="true"
writeFlag="true">
                        <file name="startup-config" fullName="nvram:/startup-config"
size="2134" readFlag="true" writeFlag="true"/>
                        <file name="private-config" fullName="nvram:/private-config"
size="1527" readFlag="false" writeFlag="false"/>
                        <file name="underlying-config" fullName="nvram:/underlying-
config" size="2134" readFlag="true" writeFlag="true"/>
                        <file name="persistent-data" fullName="nvram:/persistent-data"
size="99" readFlag="false" writeFlag="false"/>
                        <file name="ifIndex-table" fullName="nvram:/ifIndex-table"
size="0" readFlag="true" writeFlag="true"/>
                    </directory>
                </fileSystem>
                <fileSystem name="disk2" type="disk" size="64229376" freespace="63987712"
readable="true" writeable="true">
                    <directory name="/" fullName="disk2:/" readFlag="true"
writeFlag="true" modDate="1979-11-30T00:00:00.000Z">
                        <file name="spec.odm" fullName="disk2:/spec.odm" size="131739"
readFlag="true" writeFlag="true" modDate="2007-08-31T05:11:36.000Z"/>
                    </directory>
                </fileSystem>
                <fileSystem name="bootflash" type="flash" size="14942208"
freespace="8455208" readable="true" writeable="true">
                    <directory name="/" fullName="bootflash:/" readFlag="true"
writeFlag="true">
                        <file name="c7200-kboot-mz.bw" fullName="bootflash:/c7200-kboot-
mz.bw" size="5131872" readFlag="true" writeFlag="true"
modDate="1999-11-30T00:01:47.000Z"/>
                        <file name="startup-config.base" fullName="bootflash:/startup-
config.base" size="1808" readFlag="true" writeFlag="true"
modDate="1999-11-30T00:23:26.000Z"/>
                        <file name="startup-config.12dec03.balam" fullName="bootflash:/
startup-config.12dec03.balam" size="1598" readFlag="true" writeFlag="true"
modDate="2000-01-05T22:54:50.000Z"/>
                    </directory>
                </fileSystem>
            </fileSystemList>
        </response>
    </SOAP:Body>
</SOAP:Envelope>]]>]]>
```

WSMA File Copy Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <SOAP:Body>
            <request xmlns="urn:cisco:wsma-filesystem" correlator="12">
                <fileCopy erase="0" overwrite="1" filesize="131739">
                    <srcURL>tftp://oz-dirt/jbalestr/spec.odm</srcURL>
                    <dstURL>test</dstURL>
                </fileCopy>
            </request>
    </SOAP:Body>
</SOAP:Envelope>]]>]]>
```

WSMA File Copy Response

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <SOAP:Body>
```

```
            <response xmlns="urn:cisco:wsma-filesystem" correlator="12" success="1">
                <copyStatus></copyStatus>
            </response>
    </SOAP:Body>
</SOAP:Envelope>]]>]]>
```

WSMA File Delete Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <SOAP:Body>
            <request xmlns="urn:cisco:wsma-filesystem" correlator="6">
                <fileDelete>
                    <deleteFileList>
                        <filename>brick</filename>
                    </deleteFileList>
                </fileDelete>
            </request>
    </SOAP:Body>
</SOAP:Envelope>]]>]]>
```

WSMA File Delete Response

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <SOAP:Body>
            <response xmlns="urn:cisco:wsma-filesystem" correlator="6" success="1">
                <deleteStatusList>
                    <deleteStatus>
                        <fileName>brick</fileName>
                        <status>DELETED</status>
                    </deleteStatus>
                </deleteStatusList>
            </response>
    </SOAP:Body>
</SOAP:Envelope>]]>]]>
```

Configuration Examples for WSMA

Example: Enabling SSHv2 Using a Hostname and Domain Name

```
configure terminal
hostname host1
ip domain-name example.com
crypto key generate rsa
ip ssh timeout 120
ip ssh version 2
```

Example: Enabling SSHv2 Using RSA Keys

```
configure terminal
 ip ssh rsa keypair-name sshkeys
 crypto key generate rsa usage-keys label sshkeys modulus 768
 ip ssh timeout 120
 ip ssh version 2
```

Example: Configuring a WSMA Service

```
configure terminal
 wsma agent config profile prof
```

Example: Configuring the WSMA Initiator Profile

```
configure terminal
 wsma profile initiator ssh-test
 transport ssh sshserver path /mypath/bin/mywsma-app.sh user user1 6 encrypted-password
```

Example: Configuring the WSMA Listener Profile

```
configure terminal
 wsma profile listener mySession
  encap soap12
  transport ssh subsys wsma
  acl 34
  exit
```

Example: Displaying WSMA Profile Parameters

```
Device# show wsma profile connections

Listener Profile http: 0 open connections: 0 closing connections
Encap: soap11
WSSE header is required
Max message (RX) is 50 Kbytes
SOAP Faults are sent
Idle timeout infinite
Keepalive not configured
Listening via http
Listening to path /wsma.  Max Idle 0 ms.  Accepting post on plaintext connections.
Established at 01:11:04.207 UTC Tue Jan 12 2010
Tx 493475 bytes (90 msg), Tx 0 errors,
Last message sent at 05:18:08.539 UTC Sat Feb 20 2010
Rx 59457 bytes (90 msg), 0 empty msg
Last message received at 05:18:08.295 UTC Sat Feb 20 2010
Listener Profile ssh: 2 open connections: 0 closing connections
Encap: soap11
WSSE header is required
Max message (RX) is 50 Kbytes
SOAP Faults are sent
Idle timeout infinite
Keepalive not configured
Listening via ssh
SSH listener, 10 sessions accepted, 0 sessions rejected
Connected sessions...
Remote connection via SSH by user(cisco) from 172.16.29.134:44457, state connect
Established at 01:14:03.184 UTC Thu Mar 11 2010
Tx 1183 bytes (2 msg), Tx 0 errors,
Last message sent at 01:14:48.565 UTC Thu Mar 11 2010
```

```
Rx 10 bytes (1 msg), 0 empty msg
Last message received at 01:14:48.565 UTC Thu Mar 11 2010
Remote connection via SSH by user(cisco) from 172.16.154.90:45404, state connect
Established at 01:14:28.041 UTC Thu Mar 11 2010
Tx 1183 bytes (2 msg), Tx 0 errors,
Last message sent at 01:14:54.437 UTC Thu Mar 11 2010
Rx 7 bytes (1 msg), 1 empty msg
Last message received at 01:14:54.437 UTC Thu Mar 11 2010
Initiator Profile ssh-init: 0 open connections: 0 closing connections
Encap: soap11
WSSE header is required
Max message (RX) is 50 Kbytes
SOAP Faults are sent
Idle timeout infinite
Keepalive not configured
Reconnect time 60 seconds
No transport configured
```

The following example shows how to display information about Web Services Management Agent (WSMA) profile counters:

```
Device# show wsma profile counters

Statistics for profile http
incoming total 90, bad XML 0, authentication errors 0, oversized 0
outgoing total 90, absorbed 0
message internal errors 0
Connection Accepts 90, local hangup 0, remote hangup 90, keepalive hangup 0
session internal errors 0
Statistics for profile ssh
incoming total 9, bad XML 2, authentication errors 0, oversized 0
outgoing total 20, absorbed 0
message internal errors 0
Connection Accepts 8, local hangup 0, remote hangup 8, keepalive hangup 0
session internal errors 0
```

The following example shows how to display information about WSMA profile schema:

```
Device# show wsma profile schema

Schema http
New Name Space ''
<VirtualRootTag> [0, 1] required
  New Name Space 'http://schemas.xmlsoap.org/soap/envelope/'
  <Envelope> 1+ required
    <Header> any subtree is allowed
    <Body> 1 required
      <Fault> [0, 1] required
        <faultcode> 1 required
        <faultstring> 1 required
        <faultactor> [0, 1] required
        <detail> any subtree is allowed
      New Name Space 'urn:cisco:exec'
        <request> [0, 1] required
          <execCLI> 1+ required
            <cmd> 1 required
            <dialogue> 0+ required
              <expect> 1 required
              <reply> 1 required
      New Name Space 'urn:cisco:wsma-config'
        <request> [0, 1] required
<config-data> 1 required
            <cli-config-data> [0, 1] required
              <cmd> 1+ required
            <cli-config-data-block> [0, 1] required
            <xml-config-data> [0, 1] required
              <Device-Configuration> [0, 1] required
                <> any subtree is allowed
      New Name Space 'urn:cisco:wsma-filesystem'
        <request> [0, 1] required
          <fileList> [0, 1] required
          <fileDelete> [0, 1] required
```

```
            <deleteFileList> 1 required
              <filename> 1+ required
          <fileCopy> [0, 1] required
            <srcURL> 1 required
            <dstURL> 1 required
            <validationInfo> [0, 1] required
              <md5CheckSum> 1 required
            <deleteFileList> [0, 1] required
              <filename> 1+ required
      New Name Space 'urn:cisco:wsma-notify'
 <request> [0, 1] required
Schema example1
New Name Space ''
<VirtualRootTag> [0, 1] required
  New Name Space 'http://schemas.xmlsoap.org/soap/envelope/'
  <Envelope> 1+ required
    <Header> any subtree is allowed
    <Body> 1 required
      <Fault> [0, 1] required
        <faultcode> 1 required
        <faultstring> 1 required
        <faultactor> [0, 1] required
        <detail> any subtree is allowed
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
WSMA commands	*Cisco IOS Web Services Management Agent Command Reference*
IP access lists	*Security Configuration Guide: Access Control Lists* in the *Securing the Data Plan Configuration Guide Library*
IP access lists commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*
Public Key Infrastructure	*Public Key Infrastructure Configuration Guide* in the *Secure Connectivity Configuration Guide Library*
Secure Shell and Secure Shell Version 2	*Secure Shell Configuration Guide* in the *Securing User Services Configuration Guide Library*
Security commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*
WSMA schema files in XSD format	ftp://ftp.cisco.com/pub/wsma/schema/

RFCs

RFC	Title
RFC 2132	*DHCP Options and BOOTP Vendor Extensions*
RFC 2246	*The TLS Protocol Version 1.0*
RFC 4251	*The Secure Shell (SSH) Protocol Architecture*
RFC 4252	*The Secure Shell (SSH) Authentication Protocol*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Web Services Management Agent

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1 Feature Information for Web Services Management Agent

Feature Name	Releases	Feature Information
Web Services Management Agent	12.2(50)SY 12.4(24)T 15.1(1)SG 15.1(1)T Cisco IOS XE Release 3.3SG	The WSMA feature enables you to perform network configurations via the Cisco CLI over an encrypted transport. The WSMA protocol defines a set of web services through which a network device can be managed, configuration data information can be retrieved, and new configuration data can be uploaded and manipulated. WSMA uses an XML-based data encoding for configuration data and protocol messages. In Cisco IOS Release 15.1(1)T this feature was modified to include support for both listener and initiator profiles. The following commands were introduced: **acl**, **clear wsma agent**, **clear wsma profile**, **debug wsma agent**, **debug wsma profile**, **encap**, **idle-timeout**, **max-message**, **show wsma agent**, **show wsma id**, **show wsma profile**, **stealth**, **transport**, **wsma agent**, **wsma id**, **wsma profile**.

Glossary

SSHv2 —Secure Shell Version 2. SSH runs on top of a reliable transport layer and provides strong authentication and encryption capabilities. SSHv2 provides a means to securely access and securely execute commands on another computer over a network.

WSMA —Web Services Management Agent. A protocol that defines a set of web services through which a network device can be managed, configuration data information can be retrieved, and new configuration data can be uploaded and manipulated.

XML —Extensible Markup Language. A standard maintained by the World Wide Web Consortium (W3C) that defines a syntax that lets you create markup languages to specify information structures. Information structures define the type of information (for example, subscriber name or address), not how the information looks (bold, italic, and so on). External processes can manipulate these information structures and publish them in a variety of formats. XML allows you to define your own customized markup language.

Web Services Management Agent with TLS

The Web Services Management Agent (WSMA) defines a set of web services through which a network device can be managed, configuration data information can be retrieved, and new configuration data can be uploaded and manipulated. WSMA uses XML-based data encoding that is transported by the Simple Object Access Protocol (SOAP), for the configuration data and protocol messages.

You can use WSMA over Transport Layer Security (TLS) to access the entire Cisco CLI. Multiple WSMA clients can connect to the WSMA server running on Cisco software.

You can also use WSMA over TLS to initiate secure connections from Cisco software to applications over trusted and untrusted networks.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for WSMA over TLS

- WSMA over TLS requires a certificate authority (CA) server to be available on the network.

Restrictions for WSMA over TLS

- You must be running a crypto image on your device in order to configure Transport Layer Security (TLS).

Information About WSMA with TLS

WSMA over TLS

The Web Services Management Agent (WSMA) agent needs to be configured to use a service profile which is using Transport Layer Security (TLS) as a transport to run the WSMA over TLS feature. The TLS protocol uses endpoint authentication and encryption to provide secure connections over any network. Encryption protects against eavesdropping, and digital certificates (signed by a trusted CA) protect against tampering and message forgery by authenticating the endpoints.

The WSMA listener and initiator profiles use the TLS server and client adapters to create and accept TLS connections. The TLS server uses a default port (13000) to listen for incoming connections; similarly, the TLS client uses the same default port to initiate connections. You can change the default port setting by changing the profile configuration.

Trusted Certificates

The WSMA over TLS feature requires a CA server to be available on the network. The CA's public key is made known to the client, and the public key must correspond to the private key used to sign the server's certificate. The Cisco device and the remote WSMA application use the CA server to validate the certificates sent between them.

WSMA Profiles with TLS

Web Services Management Agent (WSMA) needs input from external management applications to cause actions on the device. A physical transport protocol must be configured and associated to a WSMA to allow the WSMA to communication with external management applications. The transport protocol and an encapsulation together form a WSMA profile. Any WSMA agent must be associated with a specific WSMA profile to perform valid operations. WSMA profiles demultiplex requests to the appropriate WSMA..

WSMA profiles work as a transport termination point, and allow transport and XML encapsulation parameters to be configured:

- The configurable encapsulations for WSMA are SOAP 1.1 and SOAP 1.2.
- The transportation mechanisms for WSMA are Secure Shell (SSH), HTTP, Secure HTTP (HTTPS), and TLS. This mechanism opens listening sockets for listeners on the device or connecting sockets for clients on the device.

Service Listener with TLS

The service listener is a type of Web Services Management Agent (WSMA) profile that listens for incoming connections and accepts devices from allowed addresses or accepted user IDs. The accepted addresses are configured by defining an access list.

Accepted user IDs are configured by defining the transport method that the service listener listens for. The Transport Layer Security (TLS) transport method enforces the specific user ID that is accepted.

Note WSMA listener profiles cannot access Cisco devices that are located behind a firewall.

WSMA over TLS Authentication and Authorization

Web Services Management Agent (WSMA) security is integrated with authentication, authorization, and accounting (AAA) configuration of Cisco software. The AAA associations configured on the transport layer are used by WSMA.

WSMA is designed for point-to-point operation and works over an encrypted transport. The security on the transport layer identifies and authenticates the users.

Unlike Secure Shell (SSH) or Secure HTTP (HTTPS) connections, TLS connections do not require that a user log in to a Cisco device. TLS certificates provide host-level authentication but do not always provide user-level authentication. Therefore, the Web Services Security Header (WSSE) header (if configured) is used to authenticate and authorize different users from a specified host.

For TLS listener profiles, all WSMA requests are authenticated using the Simple Object Access Protocol (SOAP) WSSE header. After the request is authenticated, the user is authorized to perform operations based on the configured privilege level. The user can be configured on the Cisco device or an the AAA server. The identity of the remote host is validated using the TLS client-side certificate.

For TLS initiator profiles, the identity of the remote endpoint is verified using the certificate authority (CA) server as part of the TLS connection setup. After a connection is established, all incoming WSMA requests are authenticated using the WSSE header. After the request is authenticated, the user is authorized to perform operations based on the configured privilege level. The user can be configured on the Cisco device or on the AAA server.

If the WSSE SOAP header is disabled for a TLS listener or initiator profile, user-level authentication is not possible, and the following process is used to decide the authorization level to assign to the profile:

- The authorization level set using the **no wsse authorization level** command is used for all agents associated with the profile.
- If no authorization level is set, the default privilege level is used. The default privilege level is set to 1 (the minimum level).

How to Configure WSMA with TLS

Configuring Certificate Validation on the TLS Client for WSMA Initiator Mode

To use the Transport Layer Security (TLS) protocol to connect to the remote host, the Cisco device (acting as the TLS client) must validate the signed certificate of the Web Services Management Agent (WSMA) application host (acting as the TLS server). To allow the device to validate the certificate and trust all certificates signed by the certificate authority (CA), you must configure a trustpoint for the CA on the device and instruct the device to download a self-signed certificate from the CA that authenticates the CA to the device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment url** *url*
5. **exit**
6. **crypto pki authenticate** *name*
7. **end**
8. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
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 my_CA`	Declares the CA that the device should use and enters ca-trustpoint configuration mode.

	Command or Action	Purpose
Step 4	**enrollment url** *url* **Example:** `Device(ca-trustpoint)# enrollment url http://myCAurl:80`	Specifies the URL of the CA.
Step 5	**exit** **Example:** `Device(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.

Step 6	**crypto pki authenticate** *name*
	
	Example:
	`Device(config)# crypto pki authenticate my_CA`
	`Certificate has the following attributes:`
	`Fingerprint MD5: AC3B4A2B FD027F65 0B4650BF 018B1F79`
	`Fingerprint SHA1: BC183062 A013FFDC 1E8E79B3 0150DEBF B887CD15`
	`% Do you accept this certificate? [yes/no]: yes`
	`Trustpoint CA certificate accepted.`
		Authenticates the CA to the device by obtaining the self-signed certificate of the CA that contains the public key of the CA.
		• Because the CA signs its own certificate, you should manually authenticate the public key of the CA by contacting the CA administrator when you perform this command.
		• After the device obtains the certificate, it displays a prompt asking you to accept the certificate.

Step 7	**end** **Example:** `Device(config)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 8	**show running-config** **Example:** `Device# show running-config`	Displays the status of the server configuration, including CA and certificate details.

Enabling a WSMA Service Initiator over TLS

If you configure service initiator over Transport Layer Security (TLS), you must first configure the certificate authority (CA) settings on the Cisco device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **wsma profile initiator** *profile-name*
4. **encap** {**soap11** | **soap12**}
5. [**backup**] **transport tls** *remote-host* [*initiator-port-number*] [**localcert** *trustpoint-name*] [**remotecert** *trustpoint-name*] [**source** *source-interface*]}
6. **keepalive** *interval* [**retries** *number*]
7. **idle-timeout** *minutes*
8. **max-message** *message-size*
9. **backup hold** *minutes*
10. **backup excluded** *seconds*
11. **reconnect** *seconds*
12. **stealth**
13. **wsse**
14. **end**

DETAILED STEPS

	Command or Action	Purpose
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	**wsma profile initiator** *profile-name* **Example:** `Device(config)# wsma profile initiator prof1`	Creates a service initiator and enters WSMA initiator configuration mode.
Step 4	**encap** {**soap11** | **soap12**} **Example:** `Device(config-wsma-initiator)# encap soap12`	(Optional) Configures an encapsulation for the service listener profile.

	Command or Action	Purpose
Step 5	[**backup**] **transport tls** *remote-host* [*initiator-port-number*] [**localcert** *trustpoint-name*] [**remotecert** *trustpoint-name*] [**source** *source-interface*]} **Example:** `Device(config-wsma-initiator)# transport tls 192.2.1.10`	Defines a transport configuration for the WSMA profile. • The port that the remote WSMA TLS application is listening on must be known. By default this is port 13000. If the server is listening on a port other than 13000, then the correct port must be configured using the *initiator-port-number* argument.
Step 6	**keepalive** *interval* [**retries** *number*] **Example:** `Device(config-wsma-initiator)# keepalive 100 retries 10`	(Optional) Enables keepalive messages and configures interval and retry values for a WSMA profile.
Step 7	**idle-timeout** *minutes* **Example:** `Device(config-wsma-initiator)# idle-timeout 345`	(Optional) Specifies the amount of time (in minutes) to keep the session alive in the absence of any data traffic.
Step 8	**max-message** *message-size* **Example:** `Device(config-wsma-initiator)# max-message 290`	(Optional) Specifies the maximum receive message size (from 1 to 2000 kilobytes).
Step 9	**backup hold** *minutes* **Example:** `Device(config-wsma-initiator)# backup hold 233`	(Optional) Sets the time (in minutes) that the WSMA profile remains connected to the backup transport configuration.
Step 10	**backup excluded** *seconds* **Example:** `Device(config-wsma-initiator)# backup excluded 30`	(Optional) Sets the time that the WSMA profile must wait before attempting to connect to the backup transport configuration after a connection is lost.

	Command or Action	Purpose
Step 11	**reconnect** *seconds* **Example:** `Device(config-wsma-initiator)# reconnect 434`	(Optional) Specifies the time for the WSMA initiator profile to wait before attempting to reconnect a session.
Step 12	**stealth** **Example:** `Device(config-wsma-initiator)# stealth`	(Optional) Configures the service to not send Simple Object Access Protocol (SOAP) fault messages in response to corrupted XML messages.
Step 13	**wsse** **Example:** `Device(config-wsma-initiator)# wsse`	(Optional) Enables the Web Services Security Header (WSSE) for a WSMA profile. • By default, the WSSE is enabled. Enter the **no wsse** command to disable the WSSE.
Step 14	**end** **Example:** `Device(config-wsma-initiator)# end`	Ends the current configuration session and returns to privileged EXEC mode.

Configuring Certificates on the TLS Server for WSMA Listener Mode

To configure CA certificates for WSMA listener mode using the TLS protocol on the Cisco IOS device, you must configure a trustpoint for the CA on the device and instruct the device to download a self-signed certificate from the CA which authenticates the CA to the device. You must then instruct the device to request it’s own certificate signed by the CA.

To enable certificates for WSMA listener mode, perform the following tasks:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment** {**url***url* | **terminal**}
5. **exit**
6. **crypto pki authenticate** *name*
7. **crypto pki enroll** *name*
8. **crypto pki import** *name* **certificate**
9. **end**
10. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
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 my_CA`	Declares the CA that the device should use and enter ca-trustpoint configuration mode.
Step 4	**enrollment** {**url***url*	**terminal**} **Example:** `Device(ca-trustpoint)# enrollment url http://myCAurl:80`	Specifies the URL of the CA. • Use the **enrollment terminal** command to specify manual cut-and-paste certificate enrollment.
Step 5	**exit** **Example:** `Device(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.

	Command or Action	Purpose

Step 6	**crypto pki authenticate** *name*
	
	Example:
	
	`Device(config)# crypto pki authenticate my_CA`
	
	`Certificate has the following attributes:`
	`Fingerprint MD5: AC3B4A2B FD027F65 0B4650BF 018B1F79`
	`Fingerprint SHA1: BC183062 A013FFDC 1E8E79B3 0150DEBF B887CD15`
	`% Do you accept this certificate? [yes/no]: yes`
	`Trustpoint CA certificate accepted.`
		Authenticates the CA to the device by obtaining the self-signed certificate of the CA that contains the public key of the CA.
		
		• Because the CA signs its own certificate, you should manually authenticate the public key of the CA by contacting the CA administrator when you perform this command.
		• If you specified manual cut-and-paste certificate enrollment in step 4, you will now be prompted to enter the encoded CA certificate.
		• After the device obtains the certificate, it displays a prompt asking you to accept the certificate.

Step 7	**crypto pki enroll** *name*
	
	Example:
	
	`Device(config)# crypto pki enroll my_CA`
	
	`% Start certificate enrollment ..`
	`% Create a challenge password. You will need to verbally provide this password to the CA Administrator in order to revoke your certificate. For security reasons your password will not be saved in the configuration. Please make a note of it.`
	`Password:`
	`Re-enter password:`
	`% The subject name in the certificate will include: devicename.cisco.com`
	`% Include the router serial number in the subject name? [yes/no]: yes`
	`% The serial number in the certificate will be: 34835646`
	`% Include an IP address in the subject name? [no]:`
	`Request certificate from CA? [yes/no]: yes`
	`% Certificate request sent to Certificate Authority`
	`% The 'show crypto pki certificate verbose my_CA' command will show the fingerprint.`
		Enrolls the device with the CA and requests certificates for this device from the CA.
		
		• The device prompts you to enter a challenge password and to select configuration options during the enrollment process.

Step 8	**crypto pki import** *name* **certificate** **Example:** `Device(config)# crypto pki import my_CA certificate`	(Optional) Manually imports a certificate to the device. • This command is required only if you selected manual cut-and-paste in step 4. • The device displays a certificate request on the console terminal. The certificate request must be copied to the CA. • The CA creates a signed certificate for the device. • The signed certificate is imported into the device using this command.

	Command or Action	Purpose
Step 9	**end** **Example:** `Device(config)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 10	**show running-config** **Example:** `Device# show running-config`	Displays the status of the server configuration, including CA and certificate details.

Enabling a WSMA Service Listener over TLS

If you configure service listener over Transport Layer Security (TLS), you must first configure the certificate authority (CA) settings on the device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **wsma profile listener** *profile-name*
4. **encap** {**soap11** | **soap12**}
5. **transport tls** [*listener-port-number*] [**localcert** *trustpoint-name*] [**disable-remotecert-validation** | **remotecert** *trustpoint-name*]
6. **idle-timeout** *minutes*
7. **max-message** *message-size*
8. **keepalive** *interval* [**retries** *number*]
9. **acl** *acl-number*
10. **stealth**
11. **wsse**
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**wsma profile listener** *profile-name* **Example:** `Device(config)# wsma profile listener prof1`	Creates a service listener and enters the Web Services Management Agent (WSMA) listener configuration mode.
Step 4	**encap** {**soap11** | **soap12**} **Example:** `Device(config-wsma-listen)# encap soap12`	(Optional) Configures an encapsulation for the service listener profile.
Step 5	**transport tls** [*listener-port-number*] [**localcert** *trustpoint-name*] [**disable-remotecert-validation** | **remotecert** *trustpoint-name*] **Example:** `Device(config-wsma-listen)# transport tls 65534`	Defines a transport configuration for the WSMA profile.
Step 6	**idle-timeout** *minutes* **Example:** `Device(config-wsma-listen)# idle-timeout 345`	(Optional) Specifies the amount of time (in minutes) to keep the session alive in the absence of any data traffic.
Step 7	**max-message** *message-size* **Example:** `Device(config-wsma-listen)# max-message 290`	(Optional) Specifies the maximum receive message size (from 1 to 2000 kilobytes).
Step 8	**keepalive** *interval* [**retries** *number*] **Example:** `Device(config-wsma-listen)# keepalive 100 retries 10`	(Optional) Enables keepalive messages and configures interval and retry values for a WSMA profile. • Keepalive messages are not sent on HTTP or Secure HTTP (HTTPS) listener connections.

	Command or Action	Purpose
Step 9	**acl** *acl-number* **Example:** `Device(config-wsma-listen)# acl 34`	(Optional) Defines the access control list (ACL) group to use.
Step 10	**stealth** **Example:** `Device(config-wsma-listen)# stealth`	(Optional) Configures the service to not send Simple Object Access Protocol (SOAP) fault messages in response to corrupted XML messages.
Step 11	**wsse** **Example:** `Device(config-wsma-listen)# wsse`	(Optional) Enables the Web Services Security Header (WSSE) for a WSMA profile. • By default, the WSSE is enabled. Enter the **no wsse** command to disable the WSSE.
Step 12	**end** **Example:** `Device(config-wsma-listen)# end`	Ends the current configuration session and returns to privileged EXEC mode.

Configuration Examples for WSMA with TLS

Example: Configuring Certificates on the TLS Server for WSMA Listener Mode

```
configure terminal
 crypto pki trustpoint my_CA
  enrollment terminal
  exit
crypto pki authenticate my_CA
 .
 .
 .
crypto pki import my_CA certificate
 .
 .
 .
```

```
end
```

Example: Enabling a WSMA Service Initiator over TLS

```
configure terminal
 wsma profile initiator profile1
  encap soap12
  keepalive 100 retries 10
  idle-timeout 120
  max-message 290
  backup hold 233
  backup excluded 30
  reconnect 434
  stealth
  wsse
```

Example: Enabling Certificate Validation on the TLS Client for WSMA Initiator Mode

```
configure terminal
 crypto pki trustpoint my_CA
  enrollment url http://myCAurl:80
  exit
crypto pki authenticate my_CA
```

Example: Enabling a WSMA Service Listener over TLS

```
configure terminal
 wsma profile listener profile1
  encap soap12
  transport tls 65534
  idle-timeout 345
  max-message 290
  keepalive 100 retries 10
  stealth
  wsse
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
WSMA commands	*Cisco IOS Web Services Management Agent Command Reference*

Related Topic	Document Title
IP access lists	*Security Configuration Guide: Access Control Lists* in the *Securing the Data Plan Configuration Guide Library*
IP access lists commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*
Public Key Infrastructure	*Public Key Infrastructure Configuration Guide* in the *Secure Connectivity Configuration Guide Library*
Secure Shell and Secure Shell Version 2	*Secure Shell Configuration Guide* in the *Securing User Services Configuration Guide Library*
Security commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*
WSMA schema files in XSD format	ftp://ftp.cisco.com/pub/wsma/schema/

RFCs

RFC	Title
RFC 2132	*DHCP Options and BOOTP Vendor Extensions*
RFC 2246	*The TLS Protocol Version 1.0*
RFC 4251	*The Secure Shell (SSH) Protocol Architecture*
RFC 4252	*The Secure Shell (SSH) Authentication Protocol*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Web Services Management Agent with TLS

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 2 Feature Information for Web Services Management Agent with TLS

Feature Name	Releases	Feature Information
Web Services Management Agent with TLS	12.2(50)SY 15.1(1)T	This feature enables support for the TLS encryption protocol for WSMA initiator and listener profiles. The following commands were introduced or modified by this feature: **backup excluded**, **backup hold**, **debug wsma profile**, **encap**, **idle-timeout**, **keepalive**, **max-message**, **reconnect**, **stealth**, **transport**, **wsma profile initiator**, **wsma profile listener**, **wsse**.