Configuring COAP Proxy Server

- Restrictions for the COAP Proxy Server, on page 1
- Information About the COAP Proxy Server, on page 1
- How to Configure the COAP Proxy Server, on page 2
- Configuration Examples for the COAP Proxy Server, on page 5
- Monitoring COAP Proxy Server, on page 9
- Feature History for COAP, on page 10

Restrictions for the COAP Proxy Server

The following restrictions apply to COAP proxy server:

- Switch cannot advertise itself as CoAP client using ipv6 broadcast (CSCuw26467).
- Support for Observe Not Implemented.
- Blockwise requests are not supported. We handle block-wise responses and can generate block-wise responses.
- DTLS Support is for the following modes only RawPublicKey and Certificate Based.
- Switch does not act as DTLS client. DTLS for endpoints only.
- Endpoints are expected to handle and respond with CBOR payloads.
- Client side requests are expected to be in JSON.
- Switch cannot advertise itself to other Resource Directories as IPv6, due to an IPv6 broadcast issue.

Information About the COAP Proxy Server

The COAP protocol is designed for use with constrained devices. COAP works in the same way on constrained devices as HTTP works on servers in accessing information.

The comparison of COAP and HTTP is shown below:

- In the case of a webserver: **HTTP** is the protocol; **TCP** is the transport; and **HTML** is the most common information format transported.

- In case of a constrained device: **COAP** is the protocol; **UDP** is the transport; and **JSON/link-format/CBOR** is the popular information format.

COAP provides a means to access and control device using a similar **GET**/**POST** metaphor and restful API as in HTTP.

How to Configure the COAP Proxy Server

To configure the COAP proxy server, you can configure the COAP proxy and COAP endpoints in the configuration mode.

The commands are: **coap [proxy | endpoints]**.

Configuring the COAP Proxy

To start or stop the COAP proxy on the switch, perform the steps given below:

Procedure

	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	**coap proxy** **Example:** `Device(config)# coap proxy`	Enters the COAP proxy sub mode. **Note** To stop the coap proxy and delete all configurations under coap proxy, use the **no coap proxy** command.
Step 4	**security** [**none** [[**ipv4** | **ipv6**] {*ip-address ip-mask/prefix*} | **list** {*ipv4-list name* | *ipv6-list-name*}] | **dtls** [**id-trustpoint** {*identity-trustpoint label*}] [**verification-trustpoint** {*verification-trustpoint*} | [**ipv4** | **ipv6** {*ip-address ip-mask/prefix*}] | **list** {*ipv4-list name* | *ipv6-list-name*}]] **Example:** `Device(config-coap-proxy)# security none ipv4 1.1.0.0 255.255.0.0`	Takes the encryption type as argument. The two security modes supported are **none** and **dtls** • **none**: Indicates no security on that port. With **security none**, a maximum of 5 ipv4 and 5 ipv6 addresses can be associated. • **dtls**: The DTLS security takes RSA trustpoint and Verification trustpoint which are optional. Without Verification

	Command or Action	Purpose
		trustpoint it does the normal Public Key Exchange. With **security dtls**, a maximum of 5 ipv4 and 5 ipv6 addresses can be associated. **Note** To delete all security configurations under coap proxy, use the **no security** command.
Step 5	**max-endpoints** {*number*} **Example:** `Device(config-coap-proxy)# max-endpoints 10`	(Optional) Specifies the maximum number of endpoints that can be learnt on the switch. The default value is 10. The range is 1 to 500. **Note** To delete all max-endpoints configured under coap proxy, use the **no max-endpoints** command.
Step 6	**port-unsecure** {*port-num*} **Example:** `Device(config-coap-proxy)# port-unsecure 5683`	(Optional) Configures a port other than the default 5683. The range is 1 to 65000. **Note** To delete all port configurations under coap proxy, use the **no port-unsecure** command.
Step 7	**port-dtls** {*port-num*} **Example:** `Device(config-coap-proxy)# port-dtls 5864`	(Optional) Configures a port other than the default 5684. **Note** To delete all dtls port configurations under coap proxy, use the **no port-dtls** command.
Step 8	**resource-directory** [ipv4 | ipv6] {*ip-address*}] **Example:** `Device(config-coap-proxy)# resource-directory ipv4 192.168.1.1`	Configures a unicast upstream resource directory server to which the switch can act as a COAP client. With **resource-directory**, a maximum of 5 of ipv4 and 5 ipv6, ip addresses can be configured. **Note** To delete all resource directory configurations under coap proxy, use the **no resource-directory** command.
Step 9	**list** [**ipv4** | **ipv6**] {*list-name*} **Example:** `Device(config-coap-proxy)# list ipv4 trial_list`	(Optional) Restricts the IP address range where the lights and their resources can be learnt. Creates a named list of ip address/masks, to be used in the **security** [**none** | **dtls**] command options above.

	Command or Action	Purpose
		With **list**, a maximum of 5 ip-lists can be configured, irrespective of ipv4 or ipv6. We can configure a max of 5 ip addresses per ip-list. **Note** To delete any ip list on the COAP proxy server, use the **no list** [**ipv4** | **ipv6**] {*list-name*} command.
Step 10	**start** **Example:** `Device(config-coap-proxy)# start`	Starts the COAP proxy on this switch.
Step 11	**stop** **Example:** `Device(config-coap-proxy)# stop`	Stops the COAP proxy on this switch.
Step 12	**exit** **Example:** `Device(config-coap-proxy)# exit`	Exits the COAP proxy sub mode.
Step 13	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring COAP Endpoints

To configure the COAP Proxy to support multiple IPv4/IPv6 static-endpoints, perform the steps given below:

Procedure

	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	**coap endpoint** [**ipv4** | **ipv6**] {*ip-address*} **Example:**	Configures the static endpoints on the switch. • **ipv4**: Configures the IPv4 Static endpoints.

	Command or Action	Purpose
	Device(config)# **coap endpoint ipv4 1.1.1.1** Device(config)# **coap endpoint ipv6 2001::1**	• **ipv6**: Configures the IPv6 Static endpoints. **Note** To stop the coap proxy on any endpoint, use the **no coap endpoint** [**ipv4** |**ipv6**] {*ip-address*} command.
Step 4	**exit** **Example:** Device(config-coap-endpoint)# **exit**	Exits the COAP endpoint sub mode.
Step 5	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.

Configuration Examples for the COAP Proxy Server

Examples: Configuring the COAP Proxy Server

This example shows how you can configure the port number 5683 to support a maximum of 10 endpoints.

```
Device# coap proxy security none ipv4 2.2.2.2 255.255.255.0 port 5683 max-endpoints 10
```

This example shows how to configure COAP proxy on *ipv4 1.1.0.0 255.255.0.0* with **no** security settings.

```
Device(config-coap-proxy)# security ?
  dtls  dtls
  none  no security

Device(config-coap-proxy)# security none ?
  ipv4    IP address range on which to learn lights
  ipv6    IPv6 address range on which to learn lights
  list    IP address range on which to learn lights

Device(config-coap-proxy)# security none ipv4 ?
  A.B.C.D  {/nn || A.B.C.D}  IP address range on which to learn lights

Device(config-coap-proxy)# security none ipv4 1.1.0.0 255.255.0.0
```

This example shows how to configure COAP proxy on *ipv4 1.1.0.0 255.255.0.0* with **dtls id trustpoint** security settings.

```
Device(config-coap-proxy)# security dtls ?
  id-trustpoint DTLS RSA and X.509 Trustpoint Labels
  ipv4 IP address range on which to learn lights
  ipv6 IPv6 address range on which to learn lights
  list IP address range on which to learn lights
```

```
Device(config-coap-proxy)# security dtls id-trustpoint ?
  WORD  Identity TrustPoint Label

Device(config-coap-proxy)# security dtls id-trustpoint RSA-TRUSTPOINT ?
  verification-trustpoint  Certificate Verification Label
  <cr>

Device(config-coap-proxy)# security dtls id-trustpoint RSA-TRUSTPOINT

Device(config-coap-proxy)# security dtls ?
  id-trustpoint DTLS RSA and X.509 Trustpoint Labels
  ipv4 IP address range on which to learn lights
  ipv6 IPv6 address range on which to learn lights
  list IP address range on which to learn lights

Device(config-coap-proxy)#  security dtls ipv4 1.1.0.0 255.255.0.0
```

Note For configuring **ipv4 / ipv6 / list**, the **id-trustpoint** and (optional) **verification-trustpoint**, should be pre-configured, else the system shows an error.

This example shows how to configure a Trustpoint. This is a pre-requisite for COAP **security dtls** with **id trustpoint** configurations.

```
ip domain-name myDomain
crypto key generate rsa general-keys exportable label MyLabel modulus 2048

Device(config)# crypto pki trustpoint MY_TRUSTPOINT
Device(ca-trustpoint)# rsakeypair MyLabel 2048
Device(ca-trustpoint)# enrollment selfsigned
Device(ca-trustpoint)# exit

Device(config)# crypto pki enroll MY_TRUSTPOINT
% Include the router serial number in the subject name? [yes/no]: no
% Include an IP address in the subject name? [no]: no
Generate Self Signed Router Certificate? [yes/no]: yes
```

This example shows how to configure COAP proxy on *ipv4 1.1.0.0 255.255.0.0* with **dtls verification trustpoint** (DTLS with certificates or verification trustpoints)

```
Device(config-coap-proxy)# security dtls ?
  id-trustpoint DTLS RSA and X.509 Trustpoint Labels
  ipv4 IP address range on which to learn lights
  ipv6 IPv6 address range on which to learn lights
  list IP address range on which to learn lights

Device(config-coap-proxy)# security dtls id-trustpoint ?
  WORD  Identity TrustPoint Label

Device(config-coap-proxy)# security dtls id-trustpoint RSA-TRUSTPOINT ?
  verification-trustpoint  Certificate Verification Label
  <cr>
```

```
Device(config-coap-proxy)# security dtls id-trustpoint RSA-TRUSTPOINT verification-trustpoint
 ?
  WORD  Identity TrustPoint Label

Device(config-coap-proxy)# security dtls id-trustpoint RSA-TRUSTPOINT verification-trustpoint
 CA-TRUSTPOINT ?
  <cr>
```

This example shows how to configure Verification Trustpoint. This is a pre-requisite for COAP **security dtls** with **verification trustpoint** configurations.

```
Device(config)# crypto pki import CA-TRUSTPOINT pkcs12 flash:hostA.p12 password cisco123
% Importing pkcs12...
Source filename [hostA.p12]?
Reading file from flash:hostA.p12
CRYPTO_PKI: Imported PKCS12 file successfully.
```

This example shows how to create a list named trial-list, to be used in the security [none | dtls] command options.

```
Device(config-coap-proxy)# list ipv4 trial_list
Device(config-coap-proxy-iplist)# 1.1.0.0 255.255.255.0
Device(config-coap-proxy-iplist)# 2.2.0.0 255.255.255.0
Device(config-coap-proxy-iplist)# 3.3.0.0 255.255.255.0
Device(config-coap-proxy-iplist)# exit
Device(config-coap-proxy)# security none list trial_list
```

This example shows all the negation commands available in the coap-proxy sub mode.

```
Device(config-coap-proxy)# no ?
  ip-list             Configure IP-List
  max-endpoints       maximum number of endpoints supported
  port-unsecure       Specify a port number to use
  port-dtls           Specify a dtls-port number to use
  resource-discovery  Resource Discovery Server
  security            CoAP Security features
```

This example shows how you can configure multiple IPv4/IPv6 static-endpoints on the coap proxy.

```
Device(config)# coap endpoint ipv4 1.1.1.1
Device(config)# coap endpoint ipv4 2.1.1.1
Device(config)# coap endpoint ipv6 2001::1
```

This example shows how you can display the COAP protocol details.

```
Device# show coap version

CoAP version 1.0.0
RFC 7252
```

```
Device# show coap resources
```

```
Link format data =
</>
</1.1.1.6/cisco/context>
</1.1.1.6/cisco/actuator>
</1.1.1.6/cisco/sensor>
</1.1.1.6/cisco/lldp>
</1.1.1.5/cisco/context>
</1.1.1.5/cisco/actuator>
</1.1.1.5/cisco/sensor>
</1.1.1.5/cisco/lldp>
</cisco/flood>
</cisco/context>
</cisco/showtech>
</cisco/lldp>

------------------------------------------------------------------------------------------

Device# show coap globals

Coap System Timer Values :
   Discovery  : 120 sec
   Cache Exp  : 5 sec
   Keep Alive : 120 sec
   Client DB  : 60 sec
   Query Queue: 500 ms
   Ack delay  : 500 ms
   Timeout    : 5 sec

Max Endpoints      : 10
Resource Disc Mode : POST

------------------------------------------------------------------------------------------

Device# show coap stats

Coap Stats :
Endpoints : 2
Requests : 20
Ext Queries : 0

------------------------------------------------------------------------------------------

Device# show coap endpoints

List of all endpoints :

Code : D - Discovered , N - New
#    Status   Age(s)  LastWKC(s)  IP
-------------------------------------------------------------------------------
1    D        10      94          1.1.1.6
2    D        6       34          1.1.1.5

Endpoints - Total : 2 Discovered : 2 New : 0

------------------------------------------------------------------------------------------

Device# show coap dtls-endpoints

#     Index State  String State     Value   Port IP
------------------------------------------------------------
1     3     SSLOK       3           48969   20.1.1.30
2     2     SSLOK       3           53430   20.1.1.31
3     4     SSLOK       3           54133   20.1.1.32
4     7     SSLOK       3           48236   20.1.1.33

------------------------------------------------------------------------------------------
```

This example shows all options available to debug the COAP protocol.

```
Device# debug coap ?

all       Debug CoAP all
database  Debug CoAP Database
errors    Debug CoAP errors
events    Debug CoAP events
packet    Debug CoAP packet
trace     Debug CoAP Trace
warnings  Debug CoAP warnings
```

Monitoring COAP Proxy Server

To display the COAP protocol details, use the commands in the following table:

Table 1: Commands to Display to COAP specific data

show coap version	Shows the IOS COAP version and the RFC information.
show coap resources	Shows the resources of the switch and those learnt by it.
show coap endpoints	Shows the endpoints which are discovered and learnt.
show coap globals	Shows the timer values and end point values.
show coap stats	Shows the message counts for endpoints, requests and external queries.
show coap dtls-endpoints	Shows the dtls endpoint status.

Table 2: Commands to Clear COAP Commands

clear coap database	Clears the COAP learnt on the switch, and the internal database of endpoint information.

To debug the COAP protocol, use the commands in the following table:

Table 3: Commands to Debug COAP protocol

debug coap database	Debugs the COAP database output.
debug coap errors	Debugs the COAP errors output.
debug coap events	Debugs the COAP events output.
debug coap packets	Debugs the COAP packets output.
debug coap trace	Debugs the COAP traces output.
debug coap warnings	Debugs the COAP warnings output.
debug coap all	Debugs all the COAP output.

Note If you wish to disable the debugs, prepend the command with a **no** keyword.

Feature History for COAP

This table provides release and related information for features explained in this module.

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

Release	Feature	Feature Information
Cisco IOS XE Everest 16.5.1a	COAP	The COAP protocol is designed for use with constrained devices. COAP works in the same way on constrained devices as HTTP works on servers in accessing information.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to Cisco Feature Navigator.