



Application Visibility and Control

- [Information About Application Visibility and Control, on page 1](#)
- [Create a Flow Monitor, on page 4](#)
- [Configuring a Flow Monitor \(GUI\), on page 5](#)
- [Create a Flow Record, on page 6](#)
- [Create a Flow Exporter , on page 8](#)
- [Configuring a Policy Tag, on page 9](#)
- [Attaching a Policy Profile to a WLAN Interface \(GUI\), on page 9](#)
- [Attaching a Policy Profile to a WLAN Interface \(CLI\), on page 10](#)
- [Attaching a Policy Profile to an AP, on page 11](#)
- [Verify the AVC Configuration, on page 11](#)
- [Default DSCP on AVC, on page 12](#)
- [AVC-Based Selective Reanchoring, on page 15](#)
- [Restrictions for AVC-Based Selective Reanchoring, on page 15](#)
- [Configuring the Flow Exporter, on page 15](#)
- [Configuring the Flow Monitor, on page 16](#)
- [Configuring the AVC Reanchoring Profile, on page 17](#)
- [Configuring the Wireless WLAN Profile Policy , on page 17](#)
- [Verifying AVC Reanchoring, on page 18](#)

Information About Application Visibility and Control

Application Visibility and Control (AVC) is a subset of the entire Flexible NetFlow (FNF) package that can provide traffic information. The AVC feature employs a distributed approach that benefits from NBAR running on the access point (AP) or controller whose goal is to run deep packet inspection (DPI) and reports the results using FNF messages.

AVC enables you to perform real-time analysis and create policies to reduce network congestion, costly network link usage, and infrastructure upgrades. Traffic flows are analyzed and recognized using the NBAR2 engine. The specific flow is marked with the recognized protocol or application. This per-flow information can be used for application visibility using FNF. After the application visibility is established, a user can define control rules with policing mechanisms for a client.

Using AVC rules, you can limit the bandwidth of a particular application for all the clients joined on the WLAN. These bandwidth contracts coexist with per-client downstream rate limiting that takes precedence over the per-application rate limits.

FNF feature is supported in wireless, and relies on the NetFlow enablement on the controller for all modes: flex, local and Fabric.

In local mode, the NBAR runs on the controller hardware and the process client traffic flows through the data plane of the controller using the AP CAPWAP tunnels.

In FlexConnect or Fabric mode, NBAR runs on the AP, and only statistics are sent to the controller. When operating in these two modes, APs regularly send FNFv9 reports back to the controller. The controller's FNF feature consumes those FNFv9 reports to provide the application statistics shown by AVC.

The Fabric mode of operation does not populate the FNF cache. It relays the FNFv9 reports at the time they arrive. As a result, some configuration of flow monitors, for example, cache timeout, is not taken into account.

The behavior of the AVC solution changes based on the wireless deployments. The following sections describe the commonalities and differences in all scenarios:

Local Mode

- NBAR is enabled on the controller.
- AVC does not push the FNF configuration to the APs.
- Roaming events are ignored.

However, AVC supports L3 roams in local mode as traffic flows through the anchor controller (where NBAR was initially processing the roaming client's traffic when the client joined).

- IOSd needs to trigger NBAR attach.
- Supports flow monitor cache.
- Supports NetFlow exporter.

Flex Mode

- NBAR is enabled on an AP
- AVC pushes the FNF configuration to the APs.
- Supports context transfer for roaming in AVC-FNF.
- Supports flow monitor cache.
- Supports NetFlow exporter.

Fabric Mode

- NBAR is enabled on an AP.
- AVC pushes the FNF configuration to the APs.
- Supports context transfer for roaming in AVC-FNF.
- Flow monitor cache is not supported.
- Supports NetFlow exporter (for the C9800 embedded on Catalyst switches for SDA, there is no FNF cache on the box).

Prerequisites for Application Visibility and Control

- The access points should be AVC capable.
However, this requirement is not applicable in Local mode.
- For the control part of AVC (QoS) to work, the application visibility feature with FNF has to be configured.

Restrictions for Application Visibility and Control

- IPv6 (including ICMPv6 traffic) packet classification is not supported in FlexConnect mode and Fabric mode. However, it is supported in Local mode.
- Layer 2 roaming is not supported across controller controllers.
- Multicast traffic is not supported.
- AVC is supported only on the following access points:
 - Cisco Catalyst 9100 Series Access Points
 - Cisco Aironet 1800 Series Access Points
 - Cisco Aironet 2700 Series Access Point
 - Cisco Aironet 2800 Series Access Point
 - Cisco Aironet 3700 Series Access Points
 - Cisco Aironet 3800 Series Access Points
 - Cisco Aironet 4800 Series Access Points
- AVC is not supported on Cisco Aironet 702W, 702I (128 M memory), and 1530 Series access points.
- Only the applications that are recognized with App visibility can be used for applying QoS control.
- Data link is not supported for NetFlow fields in AVC.
- You cannot map the same WLAN profile to both the AVC-not-enabled policy profile and the AVC-enabled policy profile.
- AVC is not supported on the management port (Gig 0/0).
- NBAR-based QoS policy configuration is allowed only on wired physical ports. Policy configuration is not supported on virtual interfaces, for example, VLAN, port channel and other logical interfaces.

When AVC is enabled, the AVC profile supports only up to 23 rules, which includes the default DSCP rule. The AVC policy will not be pushed down to the AP, if rules are more than 23.

AVC Configuration Overview

To configure AVC, follow these steps:

1. Create a flow monitor using the **record wireless avc basic** command.

2. Create a wireless policy profile.
3. Apply the flow monitor to the wireless policy profile.
4. Create a wireless policy tag.
5. Map the WLAN to the policy profile
6. Attach the policy tag to the APs.

Create a Flow Monitor

The NetFlow configuration requires a flow record, a flow monitor, and a flow exporter. This configuration should be the first step in the overall AVC configuration.



Note In Flex mode and Local mode, the default values for **cache timeout active** and **cache timeout inactive** commands are not optimal for AVC. We recommend that you set both the values to 60 in the flow monitor. For Fabric mode, the cache timeout configuration does not apply.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	flow monitor <i>monitor-name</i> Example: Device(config)# flow monitor fm_avc	Creates a flow monitor.
Step 3	record wireless avc basic Example: Device(config-flow-monitor)# record wireless avc basic	Specifies the basic wireless AVC flow template. Note The record wireless avc basic command is same as record wireless avc ipv4 basic command. However, record wireless avc ipv4 basic command is not supported in Flex or Fabric modes. In such scenarios, use the record wireless avc basic command.
Step 4	record wireless avc {ipv4 ipv6} basic Example:	Specifies the basic IPv4 or IPv6 wireless AVC flow template.

	Command or Action	Purpose
	<pre>Device(config-flow-monitor)# record wireless avc ipv6 basic</pre>	<p>Note If you want to have both Application Performance Monitoring (APM) and AVC-FNF in the device simultaneously, use the record wireless avc {ipv4 ipv6} assurance command, which is a superset of the fields contained in record wireless avc {ipv4 ipv6} basic command. If the containing flow monitor is configured with the local exporter using destination wlc local command, AVC-FNF will populate the statistics exactly as that of the record wireless avc {ipv4 ipv6} basic configuration. As a result, both APM and AVC-FNF can be configured simultaneously with two flow monitors per direction, per IP version, in local (central switching) mode.</p> <p>Note The record wireless avc basic command is same as record wireless avc ipv4 basic command. However, record wireless avc ipv4 basic command is not supported in Flex or Fabric modes. In such scenarios, use the record wireless avc basic command.</p>
Step 5	<p>cache timeout active <i>value</i></p> <p>Example:</p> <pre>Device(config-flow-monitor)# cache timeout active 60</pre>	Sets the active flow timeout in seconds.
Step 6	<p>cache timeout inactive <i>value</i></p> <p>Example:</p> <pre>Device(config-flow-monitor)# cache timeout inactive 60</pre>	Sets the inactive flow timeout in seconds.

Configuring a Flow Monitor (GUI)

Before you begin

You must have created a flow exporter to export data from the flow monitor.

Procedure

- Step 1** Choose **Configuration > Services > Application Visibility** and go to the **Flow Monitor** tab .
- Step 2** In the **Monitor** area, click **Add** to add a flow monitor.
- Step 3** In the **Flow Monitor** window, add a flow monitor and a description.
- Step 4** Select the Flow exporter from the drop-down list to export the data from the flow monitor to a collector.

Note To export wireless netflow data, use the templates below:

- ETA (Encrypted Traffic Analysis)
- wireless avc basic
- wireless avc basic IPv6

- Step 5** Click **Apply to Device** to save the configuration.

Create a Flow Record

The default flow record cannot be edited or deleted. If you require a new flow record, you need to create one and map it to the flow monitor from CLI.

Procedure

	Command or Action	Purpose
Step 1	flow record <i>flow_record_name</i> Example: Device(config)# flow record record1	Creates a flow record. Note When a custom flow record is configured in Flex and Fabric modes, the optional fields (fields that are not present in record wireless avc basic) are ignored.
Step 2	description <i>string</i> Example: Device(config-flow-record)# description IPv4flow	(Optional) Describes the flow record as a maximum 63-character string.
Step 3	match ipv4 protocol Example: Device(config-flow-record)# match ipv4 protocol	Specifies a match to the IPv4 protocol.
Step 4	match ipv4 source address Example:	Specifies a match to the IPv4 source address-based field.

	Command or Action	Purpose
	Device(config-flow-record)# match ipv4 source address	
Step 5	match ipv4 destination address Example: Device(config-flow-record)# match ipv4 destination address	Specifies a match to the IPv4 destination address-based field.
Step 6	match transport source-port Example: Device(config-flow-record)# match transport source-port	Specifies a match to the transport layer's source port field.
Step 7	match transport destination-port Example: Device(config-flow-record)# match transport destination-port	Specifies a match to the transport layer's destination port field.
Step 8	match flow direction Example: Device(config-flow-record)# match flow direction	Specifies a match to the direction the flow was monitored in.
Step 9	match application name Example: Device(config-flow-record)# match application name	Specifies a match to the application name. Note This action is mandatory for AVC support because this allows the flow to be matched against the application.
Step 10	match wireless ssid Example: Device(config-flow-record)# match wireless ssid	Specifies a match to the SSID name identifying the wireless network.
Step 11	collect counter bytes long Example: Device(config-flow-record)# collect counter bytes long	Collects the counter field's total bytes.
Step 12	collect counter packets long Example: Device(config-flow-record)# collect counter bytes long	Collects the counter field's total packets.
Step 13	collect wireless ap mac address Example: Device(config-flow-record)# collect wireless ap mac address	Collects the BSSID with the MAC addresses of the access points that the wireless client is associated with.

	Command or Action	Purpose
Step 14	collect wireless client mac address Example: <pre>Device(config-flow-record)# collect wireless client mac address</pre>	Collects the MAC address of the client on the wireless network.

Create a Flow Exporter

You can create a flow exporter to define the export parameters for a flow. This is an optional procedure for configuring flow exporter parameters.



Note For the AVC statistics to be visible at the controller, you should configure a local flow exporter using the following commands:

- **flow exporter** *my_local*
- **destination local wlc**

Also, your flow monitor must use this local exporter for the statistics to be visible at the controller.

Procedure

	Command or Action	Purpose
Step 1	flow exporter <i>flow-export-name</i> Example: <pre>Device(config)# flow exporter export-test</pre>	Creates a flow monitor.
Step 2	description <i>string</i> Example: <pre>Device(config-flow-exporter)# description IPv4flow</pre>	Describes the flow record as a maximum 63-character string.
Step 3	destination { <i>hostname/ipv4address</i> <i>hostname/ipv6address</i> <i>local {wlc}</i> } Example: <pre>Device(config-flow-exporter)# destination local wlc</pre>	Specifies the hostname or IP address of the system or the local WLC to which the exporter sends data.
Step 4	transport udp <i>port-value</i> Example: <pre>Device(config-flow-exporter)# transport udp 1024</pre>	(Optional) Configures the destination UDP port to reach the external collector. The default value is 9995. Note This step is required only for external collectors; not required for local wlc collector.

	Command or Action	Purpose
Step 5	option application-table timeout <i>seconds</i> Example: Device(config-flow-exporter)# option application-table timeout 500	(Optional) Specifies the application table timeout option, in seconds. The valid range is from 1 to 86400.
Step 6	end Example: Device(config-flow-exporter)# end	Returns to privileged EXEC mode.
Step 7	show flow exporter Example: Device# show flow exporter	(Optional) Verifies your configuration.

Configuring a Policy Tag

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	wireless tag policy <i>policy-tag-name</i> Example: Device(config-policy-tag)# wireless tag policy rr-xyz-policy-tag	Configures policy tag and enters policy tag configuration mode.
Step 3	end Example: Device(config-policy-tag)# end	Saves the configuration and exits configuration mode and returns to privileged EXEC mode.

Attaching a Policy Profile to a WLAN Interface (GUI)

Procedure

-
- Step 1** Choose **Configuration > Tags & Profiles > Tags**.
 - Step 2** On the **Manage Tags** page, click **Policy** tab.
 - Step 3** Click **Add** to view the **Add Policy Tag** window.

- Step 4** Enter a name and description for the policy tag.
 - Step 5** Click **Add** to map WLAN and policy.
 - Step 6** Choose the WLAN profile to map with the appropriate policy profile, and click the tick icon.
 - Step 7** Click **Save & Apply to Device**.
-

Attaching a Policy Profile to a WLAN Interface (CLI)

Before you begin

- Do not attach different AVC policy profiles on the same WLAN across different policy tags.

The following is an example of incorrect configuration:

```
wireless profile policy avc_pol1
  ipv4 flow monitor fm-avc1 input
  ipv4 flow monitor fm-avc1 output
  no shutdown
wireless profile policy avc_pol2
  ipv4 flow monitor fm-avc2 input
  ipv4 flow monitor fm-avc2 output
  no shutdown
wireless tag policy avc-tag1
wlan wlan1 policy avc_pol1
wireless tag policy avc-tag2
wlan wlan1 policy avc_pol2
```

This example violates the restriction stated earlier, that is, the WLAN *wlan1* is mapped to 2 policy profiles, *avc_pol1* and *avc_pol2*. This configuration is, therefore, incorrect because the WLAN *wlan1* should be mapped to either *avc_pol1* or *avc_pol2* everywhere.

- Conflicting policy profiles on the same WLAN are not supported. For example, policy profile (with and without AVC) applied to the same WLAN in different policy tags.

The following is an example of an incorrect configuration:

```
wireless profile policy avc_pol1
  no shutdown
wireless profile policy avc_pol2
  ipv4 flow monitor fm-avc2 input
  ipv4 flow monitor fm-avc2 output
  no shutdown
wireless tag policy avc-tag1
wlan wlan1 policy avc_pol1
wireless tag policy avc-tag2
wlan wlan1 policy avc_pol2
```

In this example, a policy profile with and without AVC is applied to the same WLAN in different tags.

Procedure

	Command or Action	Purpose
Step 1	wireless tag policy <i>avc-tag</i> Example: Device(config)# wireless tag policy avc-tag	Creates a policy tag.
Step 2	wlan <i>wlan-avc</i> policy <i>avc-policy</i> Example: Device(config-policy-tag)# wlan wlan_avc policy avc_pol	Attaches a policy profile to a WLAN profile.

What to do next

- Run the **no shutdown** command on the WLAN after completing the configuration.
- If the WLAN is already in **no shutdown** mode, run the **shutdown** command, followed by **no shutdown** command.

Attaching a Policy Profile to an AP

Procedure

	Command or Action	Purpose
Step 1	ap <i>ap-ether-mac</i> Example: Device(config)# ap 34a8.2ec7.4cf0	Enters AP configuration mode.
Step 2	policy-tag <i>policy-tag</i> Example: Device(config)# policy-tag avc-tag	Specifies the policy tag that is to be attached to the access point.

Verify the AVC Configuration

Procedure

	Command or Action	Purpose
Step 1	show avc wlan <i>wlan-name</i> top <i>num-of-applications</i> applications { aggregate downstream upstream }	Displays information about top applications and users using these applications.

	Command or Action	Purpose
	Example: Device# show avc wlan wlan_avc top 2 applications aggregate	Note Ensure that wireless clients are associated to the WLAN and generating traffic, and then wait for 90 seconds (to ensure the availability of statistics) before running the command.
Step 2	show avc client mac top num-of-applications applications {aggregate downstream upstream} Example: Device# show avc client 9.3.4 top 3 applications aggregate	Displays information about the top number of applications. Note Ensure that wireless clients are associated to the WLAN and generating traffic, and then wait for 90 seconds (to ensure the availability of statistics) before running the command.
Step 3	show avc wlan wlan-name application app-name top num-of-clients aggregate Example: Device# show avc wlan wlan_avc application app top 4 aggregate	Displays information about top applications and users using these applications.
Step 4	show ap summary Example: Device# show ap summary	Displays a summary of all the access points attached to the controller .
Step 5	show ap tag summary Example: Device# show ap tag summary	Displays a summary of all the access points with policy tags.

Default DSCP on AVC

Configuring Default DSCP for AVC Profile (GUI)

Procedure

-
- Step 1** Choose **Configuration** > **Services** > **QoS**.
 - Step 2** Click **Add**.
 - Step 3** Enter the **Policy Name**.
 - Step 4** Click **Add Class-Maps**.
 - Step 5** Choose **AVC** in the **AVC/User Defined** drop-down list.
 - Step 6** Click either **Any** or **All** match type radio button.
 - Step 7** Choose **DSCP** in the **Mark Type** drop-down list.

- Step 8** a) Check the **Drop** check box to drop traffic from specific sources.
 b) If you do not want to drop the traffic, enter the **Police(kbps)** and choose the match type from the **Match Type** drop-down list. Choose the items from the available list and click move them to the selected list.
- Step 9** Click **Save**.
- Step 10** Click **Apply to Device**.

Configuring Default DSCP for AVC Profile

In Cisco Catalyst 9800 Series Wireless Controller, only up to 32 filters can be specified in the policy. As there was no way of classifying the packets that are not specified in the filters, now, you can mark down these packets in the policy.

The marking action can be applied to the traffic when creating a class map and creating a policy map.

Creating Class Map

Procedure

	Command or Action	Purpose
Step 1	Configure Terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	class class-map-name] Example: Device(config-pmap)# class-map avc-class	Creates a class map.
Step 3	match protocol { application-name attribute category category-name attribute sub-category sub-category-name attribute application-group application group-name Example: Device(config)# class-map avc-class Device(config-cmap)# match protocol avc-media Device(config)# class-map class-avc-category Device(config-cmap)# match protocol attribute category avc-media Device# class-map class-avc-sub-category Device(config-cmap)# match protocol attribute sub-category avc-media Device# class-map avcS-webex-application-group Device(config-cmap)# match protocol attribute application-group webex-media	Specifies match to the application name, category name, subcategory name, or application group.

	Command or Action	Purpose
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode. Alternatively, you can also press Ctrl-Z to exit global configuration mode.

Creating Policy Map

Procedure

	Command or Action	Purpose
Step 1	Configure Terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	policy-map <i>policy-map-name</i> Example: Device(config)#policy-map avc-policy	<p>Creates a policy map by entering the policy map name, and enters policy-map configuration mode.</p> <p>By default, no policy maps are defined.</p> <p>The default behavior of a policy map is to set the DSCP to 0 if the packet is an IP packet and to set the CoS to 0 if the packet is tagged. No policing is performed.</p> <p>Note To delete an existing policy map, use the no policy-map policy-map-name global configuration command.</p>
Step 3	class [<i>class-map-name</i> class-default] Example: Device(config-pmap)# class-map avc-class	<p>Defines a traffic classification, and enters policy-map class configuration mode.</p> <p>By default, no policy map and class maps are defined.</p> <p>If a traffic class has already been defined by using the class-map global configuration command, specify its name for class-map-name in this command.</p> <p>A class-default traffic class is predefined and can be added to any policy. It is always placed at the end of a policy map. With an implied match any is included in the class-default class, all packets that have not already matched the other traffic classes will match class-default .</p> <p>Note To delete an existing class map, use the no class class-map-name policy-map configuration command.</p>

	Command or Action	Purpose
Step 4	set dscp <i>new-dscp</i> Example: Device(config-pmap-c)# set dscp 45	Classifies IP traffic by setting a new value in the packet. For dscp new-dscp , enter a new DSCP value to be assigned to the classified traffic. The range is 0 to 63.
Step 5	class <i>class-default</i>	Specifies the default class so that you can configure or modify its policy.
Step 6	set dscp default	Configures the default DSCP.
Step 7	end	Returns to privileged EXEC mode. Alternatively, you can also press Ctrl-Z to exit global configuration mode.

AVC-Based Selective Reanchoring

The AVC-Based Selective Reanchoring feature is designed to reanchor clients when they roam from one controller to another. Reanchoring of clients prevents the depletion of IP addresses available for new clients in Cisco WLC. The AVC profile-based statistics are used to decide whether a client must be reanchored or deferred. This is useful when a client is actively running a voice or video application defined in the AVC rules.

The reanchoring process also involves deauthentication of anchored clients. The clients get deauthenticated when they do not transmit traffic for the applications listed in the AVC rules while roaming between WLCs.

Restrictions for AVC-Based Selective Reanchoring

- This feature is supported only in local mode. FlexConnect and fabric modes are not supported.
- This feature is not supported in guest tunneling and export anchor scenarios.
- The old IP address is not released after reanchoring, until IP address' lease period ends.

Configuring the Flow Exporter

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	flow exporter <i>name</i> Example:	Creates a flow exporter and enters flow exporter configuration mode.

	Command or Action	Purpose
	Device(config)# flow exporter avc-reanchor	Note You can use this command to modify an existing flow exporter too.
Step 3	destination local wlc Example: Device(config-flow-exporter)# destination local wlc	Sets the exporter as local.

Configuring the Flow Monitor

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	flow monitor <i>monitor-name</i> Example: Device(config)# flow monitor fm_avc	Creates a flow monitor and enters Flexible NetFlow flow monitor configuration mode. Note You can use this command to modify an existing flow monitor too.
Step 3	exporter <i>exporter-name</i> Example: Device(config-flow-monitor)# exporter avc-reanchor	Specifies the name of an exporter.
Step 4	record wireless avc basic Example: Device(config-flow-monitor)# record wireless avc basic	Specifies the flow record to use to define the cache.
Step 5	cache timeout active <i>value</i> Example: Device(config-flow-monitor)# cache timeout active 60	Sets the active flow timeout, in seconds.
Step 6	cache timeout inactive <i>value</i> Example: Device(config-flow-monitor)# cache timeout inactive 60	Sets the inactive flow timeout, in seconds.

Configuring the AVC Reanchoring Profile

Before you begin

- Ensure that you use the AVC-Reanchor-Class class map. All other class-map names are ignored by Selective Reanchoring.
- During boot up, the system checks for the existence of the AVC-Reanchor-Class class map. If it is not found, default protocols, for example, jabber-video, WiFi-calling, and so on, are created. If AVC-Reanchor-Class class map is found, configuration changes are not made and updates to the protocols that are saved to the startup configuration persist across reboots.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	class-map <i>cmap-name</i> Example: Device(config)# class-map AVC-Reanchor-Class	Configures the class map.
Step 3	match any Example: Device(config-cmap)# match any	Instructs the device to match with any of the protocols that pass through it.
Step 4	match protocol jabber-audio Example: Device(config-cmap)# match protocol jabber-audio	Specifies a match to the application name. You can edit the class-map configuration later, in order to add or remove protocols, for example, jabber-video, wifi-calling, and so on, if required.

Configuring the Wireless WLAN Profile Policy

Follow the procedure given below to configure the WLAN profile policy:

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 2	wireless profile policy <i>policy-name</i> Example: Device(config)# wireless profile policy default-policy-profile	Configures the WLAN policy profile and enters wireless policy configuration mode.
Step 3	shutdown Example: Device(config-wireless-policy)# shutdown	Disables the policy profile.
Step 4	central switching Example: Device(config-wireless-policy)# central switching	Enables central switching.
Step 5	ipv4 flow monitor <i>monitor-name</i> input Example: Device(config-wireless-policy)# ipv4 flow monitor fm_avc input	Specifies the name of the IPv4 ingress flow monitor.
Step 6	ipv4 flow monitor <i>monitor-name</i> output Example: Device(config-wireless-policy)# ipv4 flow monitor fm_avc output	Specifies the name of the IPv4 egress flow monitor.
Step 7	reanchor class <i>class-name</i> Example: Device(config-wireless-policy)# reanchor class AVC-Reanchor-Class	Configure a class map with protocols for the Selective Reanchoring feature.
Step 8	no shutdown Example: Device(config-wireless-policy)# no shutdown	Enables the policy profile.

Verifying AVC Reanchoring

Use the following commands to verify the AVC reanchoring configuration:

```
Device# show wireless profile policy detailed avc_reanchor_policy
```

```
Policy Profile Name      : avc_reanchor_policy
Description              :
Status                  : ENABLED
VLAN                    : 1
Wireless management interface VLAN      : 34
!
.
.
```

```

.
AVC VISIBILITY : Enabled
Flow Monitor IPv4
  Flow Monitor Ingress Name : fm_avc
  Flow Monitor Egress Name : fm_avc
Flow Monitor IPv6
  Flow Monitor Ingress Name : Not Configured
  Flow Monitor Egress Name : Not Configured
NBAR Protocol Discovery : Disabled
Reanchoring : Enabled
Classmap name for Reanchoring
  Reanchoring Classmap Name : AVC-Reanchor-Class
!
.
.
.
-----

```

Device# **show platform software trace counter tag wstatsd chassis active R0 avc-stats debug**

```

Counter Name Thread ID Counter Value
-----

```

```

Reanch_deassociated_clients 28340 1
Reanch_tracked_clients 28340 4
Reanch_deleted_clients 28340 3

```

Device# **show platform software trace counter tag wncd chassis active R0 avc-afc debug**

```

Counter Name Thread ID Counter Value
-----

```

```

Reanch_co_ignored_clients 30063 1
Reanch_co_anchored_clients 30063 5
Reanch_co_deauthed_clients 30063 4

```

Device# **show platform software wlavc status wncd**

Event history of WNCDB:

```

AVC key: [1,wlan_avc,N/A,Reanc,default-policy-tag]
Current state : READY
Wlan-id : 1
Wlan-name : wlan_avc
Feature type : Reanchoring
Flow-mon-name : N/A
Policy-tag : default-policy-tag
Switching Mode : CENTRAL

```

```

Timestamp FSM State Event RC Ctx
-----

```

```

06/12/2018 16:45:30.630342 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:45:28.822780 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:28.822672 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:15.172073 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:45:12.738367 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:12.738261 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:01.162689 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:44:55.757643 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:55.757542 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:04.468749 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:44:02.18857 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:02.18717 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:38:20.164304 2 :READY 3 :FSM_AFM_SWEEP 0 2
06/12/2018 16:35:20.163877 2 :READY 1 :FSM_AFM_BIND 0 2

```

```
06/12/2018 16:35:18.593257 1 :INIT 1 :FSM_AFM_BIND 0 2
06/12/2018 16:35:18.593152 1 :INIT 24:CREATE_FSM 0 0
```

```
AVC key: [1,wlan_avc,fm_avc,v4-In,default-policy-tag]
Current state : READY
Wlan-id : 1
Wlan-name : wlan_avc
Feature type : Flow monitor IPv4 Ingress
Flow-mon-name : fm_avc
Policy-tag : default-policy-tag
Switching Mode : CENTRAL
```

```
Timestamp FSM State Event RC Ctx
```

```
-----
06/12/2018 16:45:30.664772 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:45:28.822499 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:28.822222 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:15.207605 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:45:12.738105 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:12.737997 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:01.164225 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:44:55.757266 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:55.757181 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:04.472778 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:44:02.15413 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:02.15263 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:38:20.164254 2 :READY 3 :FSM_AFM_SWEEP 0 2
06/12/2018 16:35:20.163209 1 :INIT 1 :FSM_AFM_BIND 0 2
06/12/2018 16:35:20.163189 1 :INIT 24:CREATE_FSM 0 0
```

```
AVC key: [1,wlan_avc,fm_avc,v4-Ou,default-policy-tag]
Current state : READY
Wlan-id : 1
Wlan-name : wlan_avc
Feature type : Flow monitor IPv4 Egress
Flow-mon-name : fm_avc
Policy-tag : default-policy-tag
Switching Mode : CENTRAL
```

```
Timestamp FSM State Event RC Ctx
```

```
-----
06/12/2018 16:45:30.630764 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:45:28.822621 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:28.822574 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:15.172357 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:45:12.738212 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:12.738167 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:01.164048 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:44:55.757403 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:55.757361 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:04.472561 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:44:02.18660 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:02.18588 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:38:20.164293 2 :READY 3 :FSM_AFM_SWEEP 0 2
06/12/2018 16:35:20.163799 1 :INIT 1 :FSM_AFM_BIND 0 2
06/12/2018 16:35:20.163773 1 :INIT 24:CREATE_FSM 0 0
```

```
Device# show platform software wlavc status wncmgrd
```

```
Event history of WNCMgr DB:
```

```
AVC key: [1,wlan_avc,N/A,Reanc,default-policy-tag]
Current state : READY
Wlan-id : 1
```

```

Wlan-name : wlan_avc
Feature type : Reanchoring
Flow-mon-name : N/A
Policy-tag : default-policy-tag
Switching Mode : CENTRAL
Policy-profile : AVC_POL_PYATS

```

```
Timestamp FSM State Event RC Ctx
```

```

-----
06/12/2018 16:45:30.629278 3 :WLAN_READY 24:BIND_WNCD 0 0
06/12/2018 16:45:30.629223 3 :WLAN_READY 4 :FSM_BIND_ACK 0 0
06/12/2018 16:45:30.629179 3 :WLAN_READY 4 :FSM_BIND_ACK 0 0
06/12/2018 16:45:30.510867 2 :PLUMB_READY 22:BIND_IOSD 0 0
06/12/2018 16:45:30.510411 2 :PLUMB_READY 2 :FSM_WLAN_UP 0 0
06/12/2018 16:45:30.510371 2 :PLUMB_READY 1 :FSM_WLAN_FM_PLUMB 0 0
06/12/2018 16:45:28.886377 2 :PLUMB_READY 20:UNBIND_ACK_IOSD 0 0
!
```

```

AVC key: [1,wlan_avc,fm_avc,v4-In,default-policy-tag]
Current state : READY
Wlan-id : 1
Wlan-name : wlan_avc
Feature type : Flow monitor IPv4 Ingress
Flow-mon-name : fm_avc
Policy-tag : default-policy-tag
Switching Mode : CENTRAL
Policy-profile : AVC_POL_PYATS

```

```
Timestamp FSM State Event RC Ctx
```

```

-----
06/12/2018 16:45:30.664032 3 :WLAN_READY 24:BIND_WNCD 0 0
06/12/2018 16:45:30.663958 3 :WLAN_READY 4 :FSM_BIND_ACK 0 0
06/12/2018 16:45:30.663921 3 :WLAN_READY 4 :FSM_BIND_ACK 0 0
06/12/2018 16:45:30.511151 2 :PLUMB_READY 22:BIND_IOSD 0 0
06/12/2018 16:45:30.510624 2 :PLUMB_READY 2 :FSM_WLAN_UP 0 0
06/12/2018 16:45:30.510608 2 :PLUMB_READY 1 :FSM_WLAN_FM_PLUMB 0 0
06/12/2018 16:45:28.810867 2 :PLUMB_READY 20:UNBIND_ACK_IOSD 0 0
06/12/2018 16:45:28.807239 4 :READY 25:UNBIND_WNCD 0 0
06/12/2018 16:45:28.807205 4 :READY 23:UNBIND_IOSD 0 0
06/12/2018 16:45:28.806734 4 :READY 3 :FSM_WLAN_DOWN 0 0
!
```

```

AVC key: [1,wlan_avc,fm_avc,v4-Ou,default-policy-tag]
Current state : READY
Wlan-id : 1
Wlan-name : wlan_avc
Feature type : Flow monitor IPv4 Egress
Flow-mon-name : fm_avc
Policy-tag : default-policy-tag
Switching Mode : CENTRAL
Policy-profile : AVC_POL_PYATS

```

```
Timestamp FSM State Event RC Ctx
```

```

-----
06/12/2018 16:45:30.629414 3 :WLAN_READY 24:BIND_WNCD 0 0
06/12/2018 16:45:30.629392 3 :WLAN_READY 4 :FSM_BIND_ACK 0 0
06/12/2018 16:45:30.629380 3 :WLAN_READY 4 :FSM_BIND_ACK 0 0
06/12/2018 16:45:30.510954 2 :PLUMB_READY 22:BIND_IOSD 0 0
06/12/2018 16:45:30.510572 2 :PLUMB_READY 2 :FSM_WLAN_UP 0 0
06/12/2018 16:45:30.510532 2 :PLUMB_READY 1 :FSM_WLAN_FM_PLUMB 0 0
06/12/2018 16:45:28.886293 2 :PLUMB_READY 20:UNBIND_ACK_IOSD 0 0
06/12/2018 16:45:28.807844 4 :READY 25:UNBIND_WNCD 0 0
06/12/2018 16:45:28.807795 4 :READY 23:UNBIND_IOSD 0 0

```

```
06/12/2018 16:45:28.806990 4 :READY 3 :FSM_WLAN_DOWN 0 0  
!
```