Application Visibility and Control

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 embedded wireless 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 embedded wireless controller for flex mode.

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

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 NetFlow exporter.

Prerequisites for Application Visibility and Control

- The access points should be AVC capable.
- 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

- Layer 2 roaming is not supported across embedded wireless controllercontrollers.
- Multicast traffic is not supported.
- AVC is supported only on the following 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.
- NBAR-based QoS policy configuration is allowed at client level and BSSID level, configured on policy profile.

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, 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.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**flow monitor** *monitor-name* **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.

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 embedded wireless 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 embedded wireless controller.

Procedure

	Command or Action	Purpose
Step 1	**flow exporter** *flow-export-name* **Example:** `Device(config)# flow exporter export-test`	Creates a flow monitor.
Step 2	**description** *string* **Example:** `Device(config-flow-exporter)#` **`description IPv4flow`**	Describes the flow record as a maximum 63-character string.
Step 3	**Example:** `Device(config-flow-exporter) #` **`destination`** `local wlc`	Specifies the local WLC to which the exporter sends data.
Step 4	**show flow exporter** **Example:** `Device #` **`show flow exporter`**	(Optional) Verifies your configuration.

Configure a WLAN for AVC

Follow the procedure given below to configure a WLAN for AVC:

Procedure

	Command or Action	Purpose
Step 1	**wlan** *wlan-avc* **1** *ssid-avc* **Example:** `Device(config)# wlan wlan1 1 ssid1`	Configures WLAN.
Step 2	**shutdown** **Example:** `Device(config-wlan)# shutdown`	Shuts down the WLAN.
Step 3	**no security wpa akm dot1x** **Example:** `Device(config-wlan)# no security wpa akm dot1x`	Disables security AKM for dot1x.
Step 4	**no security wpa wpa2 ciphers aes** **Example:** `Device(config-wlan)# no security wpa wpa2 ciphers aes`	Disables WPA2 ciphers for AES.

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** *policy-tag-name* **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** *avc-tag* **Example:** `Device(config)# wireless tag policy avc-tag`	Creates a policy tag.
Step 2	**wlan** *wlan-avc* **policy** *avc-policy* **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** *ap-ether-mac* **Example:** `Device(config)# ap 34a8.2ec7.4cf0`	Enters AP configuration mode.
Step 2	**policy-tag** *policy-tag* **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** *wlan-name* **top** *num-of-applications* **applications {aggregate | downstream | upstream}**
	Example:
	`Device# show avc wlan wlan_avc top 2 applications aggregate`
		Displays information about top applications and users using these 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 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 embedded wireless controller.
Step 5	**show ap tag summary** **Example:** `Device# show ap tag summary`	Displays a summary of all the access points with policy tags.

AVC-Based Selective Reanchoring

The AVC-Based Selective Reanchoring feature is designed to reanchor clients when they roam from one embedded wireless 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** *name* **Example:** `Device(config)# flow exporter avc-reanchor`	Creates a flow exporter and enters flow exporter configuration mode. **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** *monitor-name* **Example:**	Creates a flow monitor and enters Flexible NetFlow flow monitor configuration mode.

	Command or Action	Purpose
	`Device(config)# flow monitor fm_avc`	**Note** You can use this command to modify an existing flow monitor too.
Step 3	**exporter** *exporter-name* **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** *value* **Example:** `Device(config-flow-monitor)# cache timeout active 60`	Sets the active flow timeout, in seconds.
Step 6	**cache timeout inactive** *value* **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** *cmap-name* **Example:**	Configures the class map.

	Command or Action	Purpose
	`Device(config)# class-map AVC-Reanchor-Class`	
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.
Step 2	**wireless profile policy** *policy-name* **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** *monitor-name* **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** *monitor-name* **output** **Example:**	Specifies the name of the IPv4 egress flow monitor.

	Command or Action	Purpose
	`Device(config-wireless-policy)# ipv4 flow monitor fm_avc output`	
Step 7	**reanchor class** *class-name* **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 WNCD 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

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
!
```