Auto Identity

Auto Identity

The Auto Identity feature provides a set of built-in policies at global configuration and interface configuration modes. This feature is available only in Class-Based Policy Language (CPL) control policy-equivalent new-style mode. To convert all the relevant authentication commands to their CPL control policy-equivalents, use the **authentication convert-to new-style** command.

This module describes the feature and explains how to configure it.

Information About Auto Identity

Auto Identity Overview

The Cisco Identity-Based Networking Services (IBNS) solution provides a policy and identity-based framework in which edge devices can deliver flexible and scalable services to subscribers. IBNS allows the concurrent operation of IEEE 802.1x (dot1x), MAC authentication bypass (MAB), and web authentication methods, making it possible to invoke multiple authentication methods in parallel, on a single subscriber session. These authentication methods, dot1x, authentication, authorization, and accounting (AAA), and RADIUS are available in global configuration and interface configuration modes.

The Auto Identity feature uses the Cisco Common Classification Policy Language-based configuration that significantly reduces the number of commands used to configure both authentication methods and interface-level commands. The Auto Identity feature provides a set of built-in policies that are based on policy maps, class maps, parameter maps, and interface templates.

In global configuration mode, the **source template AI_GLOBAL_CONFIG_TEMPLATE** command enables the Auto Identity feature. In interface configuration mode, configure the AI_MONITOR_MODE, AI_LOW_IMPACT_MODE, or AI_CLOSED_MODE interface templates to enable the feature on interfaces.

You can configure multiple templates; however, you must bind multiple templates together using the **merge** command. If you do not bind the templates, the last configured template is used. While binding templates, if the same command is repeated in two templates with different arguments, the last configured command is used.

Note You can also enable user-defined templates that are configured using the **template** *name* command in global configuration mode .

Use the **show template interface** or **show template global** commands to display information about built-in templates. Built-in templates can be edited. Built-in template information is displayed in the output of the **show running-config** command, if the template is edited. If you delete an edited built-in template, the built-in template reverts to the default and is not deleted from the configuration. However; if you delete a user-defined template, it is deleted from the configuration.

Note Before you delete a template, ensure that it is not attached to a device.

Auto Identity Global Template

To enable the global template, configure the **source template** *template-name* command.

Note You must configure the RADIUS server commands, because these are not automatically configured when the global template is enabled.

The following example shows how to enable the global template:

```
Switch(config)# source template AI_GLOBAL_CONFIG_TEMPLATE
Switch(config)# radius server ISE
Switch(config-radius-server)# address ipv4 172.20.254.4 auth-port 1645 acct-port 1646
Switch(config-radius-server)# key cisco
Switch(config-radius-server)# end
```

The AI_GLOBAL_CONFIG_TEMPLATE automatically configures the following commands:

```
dot1x system-auth-control
aaa new-model
aaa authentication dot1x default group radius
aaa authorization network default group radius
aaa authorization auth-proxy default group radius
aaa accounting identity default start-stop group radius
aaa accounting system default start-stop group radius
radius-server attribute 6 on-for-login-auth
radius-server attribute 6 support-multiple
radius-server attribute 6 voice 1
radius-server attribute 8 include-in-access-req
radius-server attribute 25 access-request include
```

Auto Identity Interface Templates

The following interface templates are available in the Auto Identity feature:

- AI_MONITOR_MODE—Passively monitors sessions that have authentication in open mode.
- AI_LOW_IMPACT_MODE—Similar to monitor mode, but with a configured static policy such as a port access control list (PACL).

- AI_CLOSED_MODE—Secure mode in which data traffic is not allowed into the network, until authentication is complete. This mode is the default.

Note Multi-auth host mode is not supported with the LAN Lite license.

The following commands are inbuilt in the AI_MONITOR_MODE:

```
switchport mode access
 access-session port-control auto
 access-session host-mode multi-auth
 dot1x pae authenticator
 mab
 service-policy type control subscriber AI_DOT1X_MAB_POLICIES
```

The following commands are inbuilt in the AI_LOW_IMPACT_MODE:

```
switchport mode access
 access-session port-control auto
 access-session host-mode multi-auth
 dot1x pae authenticator
 mab
 ip access-group AI_PORT_ACL in
 service-policy type control subscriber AI_DOT1X_MAB_POLICIES
```

The following commands are inbuilt in the AI_CLOSED_MODE:

```
switchport mode access
 access-session closed
 access-session port-control auto
 access-session host-mode multi-auth
 dot1x pae authenticator
 mab
 service-policy type control subscriber AI_DOT1X_MAB_POLICIES
```

Auto Identity Built-in Policies

The following five built-in policies are available in the Auto Identity feature:

- AI_DOT1X_MAB_AUTH—Enables flexible authentication with dot1x, and then MAC Address Bypass (MAB).
- AI_DOT1X_MAB_POLICIES—Enables flexible authentication with dot1x, and then MAB. Applies critical VLAN in case the Authentication, Authorization, and Accounting (AAA) server is not reachable.
- AI_DOT1X_MAB_WEBAUTH—Enables flexible authentication with dot1x, MAB, and then web authentication.
- AI_NEXTGEN_AUTHBYBASS—Skips authentication if an IP phone device is detected. Enables the **device classifier** command in global configuration mode and the **voice-vlan** command in interface configuration mode to detect the device. This is a reference policy map, and users can copy the contents of this policy map to other policy maps.
- AI_STANDALONE_WEBAUTH—Defines standalone web authentication.

Auto Identity Class Maps Templates

The following built-in class maps are supported by the Auto Identity feature:

- AI_NRH—Specifies that the nonresponsive host (NRH) authentication method is enabled.
- AI_WEBAUTH_METHOD—Specifies that the web authentication method is enabled.
- AI_WEBAUTH_FAILED—Specifies that the web authentication method failed to authenticate.
- AI_WEBAUTH_NO_RESP—Specifies that the web authentication client failed to respond.
- AI_DOT1X_METHOD—Specifies that the dot1x method is enabled.
- AI_DOT1X_FAILED—Specifies that the dot1x method failed to authenticate.
- AI_DOT1X_NO_RESP—Specifies that the dot1x client failed to respond.
- AI_DOT1X_TIMEOUT—Specifies that the dot1x client stopped responding after the initial acknowledge (ACK) request.
- AI_MAB_METHOD—Specifies that the MAC Authentication Bypass (MAB) method is enabled.
- AI_MAB_FAILED—Specifies that the MAB method failed to authenticate.
- AI_AAA_SVR_DOWN_AUTHD_HOST—Specifies that the Authentication, Authorization, and Accounting (AAA) server is down, and the client is in authorized state.
- AI_AAA_SVR_DOWN_UNAUTHD_HOST—Specifies that the AAA server is down, and the client is in authorized state.
- AI_IN_CRITICAL_AUTH—Specifies that the critical authentication service template is applied.
- AI_NOT_IN_CRITICAL_AUTH—Specifies that the critical authentication service template is not applied.
- AI_METHOD_DOT1X_DEVICE_PHONE—Specifies that the method is dot1x and the device type is IP phone.
- AI_DEVICE_PHONE—Specifies that the device type is IP phone.

Auto Identity Parameter Maps

The following built-in parameter map templates are supported by the Auto Identity feature:

- AI_NRH_PMAP—Starts nonresponsive host (NRH) authentication.

  AI_WEBAUTH_PMAP—Starts web authentication.

Auto Identity Service Templates

Service templates are available inside builit-in policy maps. The following built-in service templates are supported by the Auto Identity feature:

- AI_INACTIVE_TIMER—Template to start the inactivity timer.
- AI_CRITICAL_ACL—Dummy template; users can configure this template as per their requirements.

How to Configure Auto Identity

Configuring Auto Identity Globally

Procedure

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	**sourcetemplate** {**AI_GLOBAL_CONFIG_TEMPLATE** | *template-name*} **Example:** `Switch(config)# source template AI_GLOBAL_CONFIG_TEMPLATE`	Configures an auto identity template. • **AI_GLOBAL_CONFIG_TEMPLATE** is a built-in template. • *template-name* is a user-defined template.
Step 4	**aaa new-model** **Example:** `Switch(config)# aaa new-model`	Enables the authentication, authorization, and accounting (AAA) access control mode.
Step 5	**radius server** *name* **Example:** `Switch(config)# radius server ISE`	Specifies the name for the RADIUS server configuration for Protected Access Credential (PAC) provisioning and enters RADIUS server configuration mode.
Step 6	**address ipv4** {*hostname* | *ipv4-address*} **Example:** `Switch(config-radius-server)# address ipv4 10.1.1.1`	Configures the IPv4 address for the RADIUS server accounting and authentication parameters. **Note** This command is not a part of the global template, and you must configure it.
Step 7	**key ipv4** {**0** *string* | **7** *string*} *string* **Example:** `Switch(config-radius-server)# key ipv4 cisco`	Specifies the authentication and encryption key for all RADIUS communications between the device and the RADIUS server. **Note** This command is not a part of the global template, and you must configure it.
Step 8	**end** **Example:** `Switch(config-radius-server)# end`	Exits RADIUS server configuration mode and returns to privileged EXEC mode.

Configuring Auto Identity at an Interface Level

When you configure two interface templates, you must configure the **merge** keyword. If you do not, the last configured template is used.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Switch(config)# interface gigabitethernet 1/0/1`	Configures an interface and enters interface configuration mode.
Step 4	**source template** {**AI_CLOSED_MODE** | **AI_LOW_IMPACT_MODE** | **AI_MONITOR_MODE** | *template-name*} [**merge**] **Example:** `Switch(config-if)# source template AI_CLOSED_MODE`	Configures a source template for the interface.
Step 5	**source template** {**AI_CLOSED_MODE** | **AI_LOW_IMPACT_MODE** | **AI_MONITOR_MODE** | *template-name*} [**merge**] **Example:** `Switch(config-if)# source template AI_MONITOR_MODE merge`	(Optional) Configures a source template for the interface and merges this template with the previously configured template • When you configure two templates, if you do not configure the **merge** keyword, the last configured template is used.
Step 6	**switchport access vlan** *vlan-id* **Example:** `Switch(config-if)# switchport access vlan 100`	Sets the VLAN when the interface is in access mode.
Step 7	**switchport voice vlan** *vlan-id* **Example:** `Switch(config-if)# switchport voice vlan 101`	Configures a voice VLAN on a multiple VLAN access port.
Step 8	Repeat Steps 4, 6, and 7 on all interfaces that must have the Auto Identity feature configured.	—
Step 9	**end** **Example:** `Switch(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Auto Identity

Example: Configuring Auto Identity Globally

```
Switch> enable
Switch# configure terminal
Switch(config)# source template AI_GLOBAL_CONFIG_TEMPLATE
Switch(config)# aaa new-model
Switch(config)# radius server ISE
Switch(config-radius-server)# address ipv4 10.1.1.1
Switch(config-radius-server)# key ipv4 cisco
Switch(config-radius-server)# end
```

Example: Configuring Auto Identity at an Interface Level

```
Switch> enable
Switch# configure terminal
Switch(config)# interface gigabitethernet 1/0/1
Switch(config-if)# source template AI_CLOSED_MODE
Switch(config-if)# source template AI_MONITOR_MODE merge
Switch(config-if)# switchport access vlan 100
Switch(config-if)# switchport voice vlan 101
Switch(config-if)# end
```

Verifying Auto Identity

Procedure

Step 1 **enable**

Example:

```
Switch> enable
```

Enables Privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show template interface source built-in all**

Displays all the configured built-in interface templates.

Example:

```
Switch# show template interface source built-in all

Template Name       : AI_CLOSED_MODE
Modified            : No
Template Definition :
 dot1x pae authenticator
 switchport mode access
 mab
 access-session closed
```

```
 access-session port-control auto
 service-policy type control subscriber AI_DOT1X_MAB_POLICIES
!
Template Name       : AI_LOW_IMPACT_MODE
Modified            : No
Template Definition :
 dot1x pae authenticator
 switchport mode access
 mab
 access-session port-control auto
 service-policy type control subscriber AI_DOT1X_MAB_POLICIES
 ip access-group AI_PORT_ACL in
!
Template Name       : AI_MONITOR_MODE
Modified            : No
Template Definition :
 dot1x pae authenticator
 switchport mode access
 mab
 access-session port-control auto
 service-policy type control subscriber AI_DOT1X_MAB_POLICIES
!
```

Step 3 **show template global source built-in all**

Displays all the configured global built-in templates.

Example:

```
Switch# show template global source built-in all

Global Template Name       : AI_GLOBAL_CONFIG_TEMPLATE
Modified                   : No
Global Template Definition : global
 dot1x system-auth-control
 aaa new-model
 aaa authentication dot1x default group radius
 aaa authorization network default group radius
 aaa authorization auth-proxy default group radius
 aaa accounting identity default start-stop group radius
 aaa accounting system default start-stop group radius
 radius-server attribute 6 on-for-login-auth
 radius-server attribute 6 support-multiple
 radius-server attribute 6 voice 1
 radius-server attribute 8 include-in-access-req
 radius-server attribute 25 access-request include
!
```

Step 4 **show derived-config | include aaa | radius-server**

Displays the composite results of all the configuration commands that apply to an interface, including commands that come from sources such as static templates, dynamic templates, dialer interfaces, and authentication, authorization, and accounting (AAA) per-user attributes.

Example:

```
Switch# show derived-config | inc aaa| radius-server

aaa new-model
aaa authentication dot1x default group radius
aaa authorization network default group radius
aaa authorization auth-proxy default group radius
aaa accounting identity default start-stop group radius
```

```
aaa accounting system default start-stop group radius
aaa session-id common
radius-server attribute 6 on-for-login-auth
radius-server attribute 6 support-multiple
radius-server attribute 6 voice 1
radius-server attribute 8 include-in-access-req
radius-server attribute 25 access-request include
radius-server host 10.25.18.42 key cisco
```

Step 5 **show derived-config | interface** *type-number*

Displays the composite results of all configuration for an interface.

Example:

```
Switch# show derived-config | interface gigabitethernet2/0/6

Building configuration...

Derived configuration : 267 bytes
!
interface GigabitEthernet2/0/6
 switchport mode access
 switchport voice vlan 100
 access-session closed
 access-session port-control auto
 mab
 dot1x pae authenticator
 spanning-tree portfast edge
 service-policy type control subscriber AI_DOT1X_MAB_POLICIES
end
```

Step 6 **show access-session | interface** *interface-type-number* **details**

Displays the policies applied to an interface.

Example:

```
Switch# show access-session interface gigabitethernet2/0/6 details

Interface:  GigabitEthernet2/0/6
          MAC Address:  c025.5c43.be00
         IPv6 Address:  Unknown
         IPv4 Address:  Unknown
            User-Name:  CP-9971-SEPC0255C43BE00
          Device-type:  Cisco-IP-Phone-9971
               Status:  Authorized
               Domain:  VOICE
       Oper host mode:  multi-auth
     Oper control dir:  both
      Session timeout:  N/A
    Common Session ID:  091A1C5B00000017002003EE
      Acct Session ID:  0x00000005
               Handle:  0xBB00000B
       Current Policy:  AI_DOT1X_MAB_POLICIES

Local Policies:

Server Policies:
           Vlan Group:  Vlan: 100
```

```
      Security Policy:  Must Not Secure
      Security Status:  Link Unsecure


Method status list:
       Method           State
       dot1x            Authc Success
```

Step 7 **show running-config interface** *type-number*

Displays the contents of the current running configuration file or the configuration for an interface.

Example:

```
Switch# show running-config interface gigabitethernet2/0/6

Building configuration...

Current configuration : 214 bytes
!
interface GigabitEthernet2/0/6
 switchport mode access
 switchport voice vlan 100
 access-session port-control auto
 spanning-tree portfast edge
 service-policy type control subscriber AI_NEXTGEN_AUTHBYPASS
end
```

Step 8 **show lldp neighbor**

Displays information about one or all neighboring devices discovered using the Link Layer Discovery Protocol (LLDP).

Example:

```
Switch# show lldp neighbor

Capability codes:
    (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
    (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

Device ID           Local Intf     Hold-time  Capability      Port ID
SEPC0255C43BE00     Gi2/0/6        180        B,T             C0255C43BE00:P1

Total entries displayed: 1
```

Feature Information for Auto Identity

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 Auto Identity

Feature Name	Releases	Feature Information
Auto Identity	Cisco IOS Release 15.2(4)E	The Auto Identity feature provides a set of built-in policies at the global configuration and interface configuration modes. This feature is available only in the Class-Based Policy Language (CPL) control policy-equivalent new-style mode. In Cisco IOS Release 15.2(4)E, this feature was implemented on Cisco Catalyst 2960–X Series Switches, Catalyst 3750–X Series Switches, and Cisco Catalyst 4500E Supervisor Engine 7-E. The following commands was introduced or modified: **source-template**.