Configuring Authentication Authorization and Accounting Services

The fabric network devices are configured with Authentication, Authorization, and Accounting (AAA) policies to provide secure fabric access to the endpoints. Authentication is the process of establishing and confirming the identity of a client requesting access to the network. Authorization is the process of authorizing access to some set of network resources. Accounting is process of recording what was done and accessed by the client. The AAA policies are enforced at the access layer of the network (the fabric edge node to which an endpoint connects), using SGTs for segmentation within the virtual network and dynamic VLAN assignments for mapping endpoints to the virtual networks.

Configure Username and Password on the Switch

You can configure username and password pairs, which are locally stored on the switch. These pairs are assigned to lines or ports and authenticate each user before that user can access the switch. If you have defined privilege levels, you can also assign a specific privilege level (with associated rights and privileges) to each username and password pair.

To configure a local username and password on the switch, perform this task:

Procedure

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

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**username** *name* [**privilege** *level*] {**password** { *encryption_type password* }	Sets the username, privilege level, and password for each user. **Example:** `Device(config)# username admin privilege 15 password 7 user-password` • For *name*, specify the user ID as one word or the MAC address. Spaces and quotation marks are not allowed. • (Optional) For *level*, specify the privilege level the user has after gaining access. The range is 0 to 15. Level 15 gives privileged EXEC mode access. Level 1 gives user EXEC mode access. • For *encryption-type*, enter 0 to specify that an unencrypted password will follow. Enter 7 to specify that a hidden password will follow. • For password, specify the password the user must enter to gain access to the Switch. The password must be from 1 to 25 characters, can contain embedded spaces, and must be the last option specified in the **username** command.
Step 4	**enable secret** [**level** *level*] {*password* | *encryption-type encrypted-password*}	Defines a secret password, which is saved using a nonreversible encryption method. **Example:** `Device(config)# enable secret level 1 secret-pwd` • (Optional) For *level*, the range is from 0 to 15. Level 1 is normal user EXEC mode privileges. The default level is 15 (privileged EXEC mode privileges). • For *password*, specify a string from 1 to 25 alphanumeric characters. The string cannot start with a number, is case sensitive, and allows spaces but ignores leading spaces. By default, no password is defined. • (Optional) For *encryption-type*, enter either 0, or 5, or 8, or 9. If you specify an encryption type, you must provide an encrypted password—an encrypted password that you copy from another switch configuration.

	Command or Action	Purpose
		• 0: Specifies an UNENCRYPTED password will follow • 5: Specifies a MD5 HASHED secret will follow • 8: Specifies a PBKDF2 HASHED secret will follow • 9: Specifies a SCRYPT HASHED secret will follow **Note** If you specify an encryption type and then enter a clear text password, you can not re-enter privileged EXEC mode. You cannot recover a lost encrypted password by any method.
Step 5	**end** **Example:** `Device(config)# end`	Exits the configuration mode and returns to privileged EXEC mode.

Configure Login Authentication Using AAA

The AAA security services facilitate a variety of login authentication methods. Use the **aaa authentication login** command to enable AAA authentication no matter which of the supported login authentication methods you decide to use. With the **aaa authentication login** command, you create one or more lists of authentication methods that are tried at login. These lists are applied using the **login authentication** line configuration command.

To configure login authentication by using AAA, use the following commands beginning in global configuration mode:

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.

	Command or Action	Purpose
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables AAA.

Step 4	**aaa authentication login** {**default** | *list-name*} *method1*[*method2...*]
	Example:
	`Device(config)# aaa authentication login default local`
	`Device(config)# aaa authentication login cts-list group client-radius-group local`
		Creates a local authentication list.

Step 5	**line** [**aux** | **console** | **tty** | **vty**] **line-number** [**ending-line-number**] **Example:** `Device(config)# line vty 1`	Enters line configuration mode for the lines to which you want to apply the authentication list.
Step 6	**login local** **Example:** `Device(config-line)# login local`	Enables local password checking at login time. Authentication is based on the username password that is specified earlier.
Step 7	**end** **Example:** `Device(config-line)# end`	Exits line configuration mode and returns to privileged EXEC mode.

Configure 802.1x Authentication Using AAA

To configure dot1x authentication by using AAA, use the following commands beginning in global configuration mode:

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	**aaa new-model** **Example:**	Enables AAA.

	Command or Action	Purpose
	`Device(config)# aaa new-model`	
Step 4	**aaa authentication dot1x** { **default**} *method1* **Example:** `Device(config)# aaa authentication dot1x default group client-radius-group`	Enables AAA accounting and to create method lists defining specific accounting methods on a per-line or per-interface basis for IEEE 802.1x sessions. Creates an IEEE 802.1x authentication method list. To create a default list that is used when a named list is not specified in the **authentication** command, use the **default** keyword followed by the method that is to be used in default situations. The default method list is automatically applied to all ports. For *method1*, enter the **group radius** keywords to use the list of all RADIUS servers for authentication. **Note** Though other keywords are visible in the command-line help string, only the **group radius** keywords are supported.
Step 5	**dot1x system-auth-control** **Example:** `Device(config)# dot1x system-auth-control`	Globally enables 802.1x port-based authentication.
Step 6	**end** **Example:** `Device(config)# end`	Exits the configuration mode and returns to privileged EXEC mode.

Configure AAA Authorization Using Named Method Lists

To configure AAA authorization using named method lists, use the following commands beginning in global configuration mode:

Procedure

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**aaa authorization** {**auth-proxy** | **network** | **exec** | **commands** *level* | **reverse-access** | **configuration** | **ipmobile**} {**default** | *list-name*} [*method1* [*method2...*]]
	Example:
	`Device(config)# aaa authorization exec default local`
	`Device(config)# aaa authorization network default group client-radius-group`
	`Device(config)# aaa authorization network cts-list group client-radius-group`
		Creates an authorization method list for a particular authorization type and enable authorization.

Step 4	Do one of the following:
	• **line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*]
	• **interface** *interface-type interface-number*
	Example:
	`Device(config)# line 1`
	`Device(config)# interface gigabitethernet 0/1/1`
		Enters the line configuration mode for the lines to which you want to apply the authorization method list.
		Alternately, enters the interface configuration mode for the interfaces to which you want to apply the authorization method list.

Step 5	Do one of the following:
	• **authorization**{**arap** | **commands** *level* | **exec** | **reverse-access**} {**default** | *list-name*}
	• **ppp authorization**{**default** | *list-name*}
	Example:
	`Device(config-line)# authorization commands default`
	`Device(config-if)# ppp authorization default`
		Applies the authorization list to a line or set of lines.
		Alternately, applies the authorization list to an interface or set of interfaces.

Step 6	**end**
	Example:
	`Device(config-line)# end`
	`Device(config-if)# end`
		Exits line configuration mode and returns to privileged EXEC mode.
		Exits interface configuration mode and returns to privileged EXEC mode.

Configure AAA Accounting Using Named Method Lists

To configure AAA Accounting using named method lists, perform the following steps:

Note System accounting does not use named method lists. For system accounting, define only the default method list.

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	**aaa accounting identity** { *name* | **default** } **start-stop** { **broadcast group** { *name* | **radius** | **tacacs+**} [**group** { *name* | **radius** | **tacacs+**} ...] | **group** { *name* | **radius** | **tacacs+**} [**group** { *name* | **radius** | **tacacs+**} ...]}
	Example:
	`Device(config)# aaa accounting Identity default start-stop group client-radius-group`
	`Device(config)# aaa accounting update newinfo periodic 2880`
		Enables accounting for IEEE 802.1x, MAC authentication bypass (MAB), and web authentication sessions

Step 4	Do one of the following: • **line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*] • **interface** *interface-type interface-number* **Example:** `Device(config)# line aux line1`	Enters the line configuration mode for the lines to which the accounting method list is applied. or Enters the interface configuration mode for the interfaces to which the accounting method list is applied.
Step 5	Do one of the following: • **accounting** {**arap** | **commands** *level* | **connection** | **exec**} {**default** | *list-name*} • **ppp accounting**{**default** | *list-name*} **Example:** `Device(config-line)# accounting arap default`	Applies the accounting method list to a line or set of lines. or Applies the accounting method list to an interface or set of interfaces.
Step 6	**end** **Example:** `Device(config-line)# end`	(Optional) Exits line configuration mode and returns to privileged EXEC mode.

Configure CoA on the Device

Follow these steps to configure CoA on a device.

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	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables AAA.
Step 4	**aaa server radius dynamic-author** **Example:** `Device(config)# aaa server radius dynamic-author`	Configures the device as an authentication, authorization, and accounting (AAA) server to facilitate interaction with an external policy server, and enters dynamic authorization local server configuration mode.
Step 5	**client** {*ip-address* | *name*} [**vrf** *vrfname*] [**server-key** *string*] **Example:** `Device(config-locsvr-da-radius)# client 172.16.2.1 server-key 7 server-pwd`	Specifies a RADIUS client from which a device will accept CoA and disconnect requests. Specify all the Policy Administration Nodes (PANs) or Policy Services Nodes (PSNs), if you have a multi-node deployment.
Step 6	**end** **Example:** `Device(config-locsvr-da-radius)# end`	Exits dynamic authorization local server configuration mode and returns to privileged EXEC mode.

Identify the RADIUS Server Host

To apply these settings globally to all RADIUS servers communicating with the device, use the three unique global configuration commands: **radius-server timeout**, **radius-server retransmit**, and **key** *string*.

You can configure the device to use AAA server groups to group existing server hosts for authentication.

You also need to configure some settings on the RADIUS server. These settings include the IP address of the device and the key string to be shared by both the server and the device.

Follow these steps to configure per-server RADIUS server communication.

Before you begin

If you configure both global and per-server functions (timeout, retransmission, and key commands) on the device, the per-server timer, retransmission, and key value commands override global timer, retransmission, and key value commands.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. TEST Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**radius server** *server name* **Example:** `Device(config)# radius server radius_172.16.2.1`	Specifies the name for the RADIUS server configuration for Protected Access Credential (PAC) provisioning, and enters RADIUS server configuration mode.
Step 4	**address** {**ipv4 | ipv6**}*ip address*{ **auth-port** *port number* | **acct-port** *port number*} **Example:** `Device(config-radius-server)# address ipv4 172.16.2.1 auth-port 1812 acct-port 1813`	(Optional) Specifies the RADIUS server parameters. For **auth-port** *port-number*, specify the UDP destination port for authentication requests. The default is 1645. The range is 0 to 65536. For **acct-port** *port-number*, specify the UDP destination port for accounting requests. The default is 1646.
Step 5	**timeout** *seconds* **Example:** `Device(config-radius-server)# timeout 2`	(Optional) Specifies the time interval that the device waits for the RADIUS server to reply before sending a request again. The range is 1 to 1000. This setting overrides the **radius-server timeout** global configuration command setting. We recommend a timeout value of two seconds.
Step 6	**retransmit** *value* **Example:** `Device(config-radius-server)# retransmit 1`	(Optional) Specifies the number of times a RADIUS request is resent when the server is not responding or responding slowly. The range is 1 to 100. This setting overrides the **radius-server retransmit** global configuration command setting.

	Command or Action	Purpose
Step 7	**automate-tester username** *user* [**ignore-auth-port**] [**ignore-acct-port**] [**idle-time** *minutes*] **probe-on** **Example:** `Device(config-radius-server)# automate-tester username dummy ignore-acct-port probe-on`	Enables RADIUS automated testing for a non-default VRF.
Step 8	**pac key** *encryption-key* **Example:** `Device(config-radius-server)# pac key 7 pac-key`	Specifies the Protected Access Credential (PAC) encryption key.
Step 9	**exit** **Example:** `Device(config-radius-server)# exit`	Exits RADIUS server configuration mode, and enters global configuration mode.

Step 10	**radius-server attribute** *attribute* {**on-for-login-auth** | **support-multiple** | **include-in-access-req** | **access-request include** | **mac format ietf upper-case** | **send nas-port-detail mac-only**}
	Example:
	`Device(config)# radius-server attribute 6 on-for-login-auth`
	`Device(config)# radius-server attribute 6 support-multiple`
	`Device(config)# radius-server attribute 8 include-in-access-req`
	`Device(config)# radius-server attribute 25 access-request include`
	`Device(config)# radius-server attribute 31 mac format ietf upper-case`
	`Device(config)# radius-server attribute 31 send nas-port-detail mac-only`
		Provides for the presence of the Service-Type attribute in RADIUS Access-Accept messages.

Step 11	**radius-server dead-criteria** [**time** *seconds*] [**tries** *number-of-tries*] **Example:** `Device(config)# radius-server dead-criteria time 5 tries 3`	Forces one or both of the criteria, used to mark a RADIUS server as dead, to be the indicated constant.
Step 12	**radius-server deadtime** *minutes* **Example:** `Device(config)# radius-server deadtime 3`	Improves RADIUS response times when some servers might be unavailable and causes the unavailable servers to be skipped immediately.
Step 13	**end** **Example:**	Exits global configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# end`	

Configure the Source Interface on RADIUS Server Group

Follow these steps to configure the source interface and for authentication and accounting on RADIUS server groups:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa group server radius** *group_name* **Example:** `Device(config)# aaa group server radius client-radius-group`	Defines the RADIUS server group configuration and enters RADIUS server group configuration mode.
Step 4	**server name** *name* **Example:** `Device(config-sg-radius)# server name radius_172.16.2.1`	Associates the RADIUS server to the server group.
Step 5	{**ip** | **ipv6**} **radius source-interface** *type number* **Example:** `Device(config-sg-radius)# ip radius source-interface Loopback0`	Specifies an interface to use for the source address in RADIUS server.
Step 6	**end** **Example:** `Device(config-radius-server)# end`	Exits RADIUS server mode and enters privileged EXEC mode.

Configure IBNS

To configure IBNS, perform the following tasks:

Configure a Control Class

A control class defines the conditions under which the actions of a control policy are executed. You define whether all, any, or none of the conditions must evaluate true to execute the actions of the control policy. Control classes are evaluated based on the event specified in the control policy.

Note This procedure shows all of the match conditions that you can configure in a control class. You must specify at least one condition in a control class to make it valid. All other conditions, and their corresponding steps, are optional (steps 4 through 18 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	**class-map type control subscriber** {**match-all** | **match-any** | **match-none**} *control-class-name* **Example:** `Device(config)# class-map type control subscriber match-all DOT1X_NO_AGENT`	Creates a control class and enters control class-map filter mode. • **match-all**: All of the conditions in the control class must evaluate true. • **match-any**: At least one of the conditions in the control class must evaluate true. • **match-none**: All of the conditions in the control class must evaluate false.
Step 4	{**match** | **no-match**} **activated-service-template** *template-name* **Example:** `Device(config-filter-control-classmap)# match activated-service-template SVC_1`	(Optional) Creates a condition that evaluates true based on the service template activated on a session.
Step 5	{**match** | **no-match**} **authorization-status** {**authorized** | **unauthorized**} **Example:** `Device(config-filter-control-classmap)# match authorization-status authorized`	(Optional) Creates a condition that evaluates true based on a session's authorization status.
Step 6	{**match** | **no-match**} **authorizing-method-priority** {**eq** | **gt** | **lt**} *priority-value*	(Optional) Creates a condition that evaluates true based on the priority of the authorization method.

	Command or Action	Purpose
	Example: `Device(config-filter-control-classmap)# match authorizing-method-priority eq 10`	• **eq**: Current priority is equal to *priority-value*. • **gt**: Current priority is greater than *priority-value*. • **lt**: Current priority is less than *priority-value*. • *priority-value*: Priority value to match. Range: 1 to 254, where 1 is the highest priority and 254 is the lowest.
Step 7	{**match** | **no-match**} **client-type** {**data** | **switch** | **video** | **voice**} **Example:** `Device(config-filter-control-classmap)# match client-type data`	(Optional) Creates a condition that evaluates true based on an event's device type.
Step 8	{**match** | **no-match**} **current-method-priority** {**eq** | **gt** | **lt**} *priority-value* **Example:** `Device(config-filter-control-classmap)# match current-method-priority eq 10`	(Optional) Creates a condition that evaluates true based on the priority of the current authentication method.
Step 9	{**match** | **no-match**} **ip-address** *ip-address* **Example:** `Device(config-filter-control-classmap)# match ip-address 10.10.10.1`	(Optional) Creates a condition that evaluates true based on an event's source IPv4 address.
Step 10	{**match** | **no-match**} **ipv6-address** *ipv6-address* **Example:** `Device(config-filter-control-classmap)# match ipv6-address FE80::1`	(Optional) Creates a condition that evaluates true based on an event's source IPv6 address.
Step 11	{**match** | **no-match**} **mac-address** *mac-address* **Example:** `Device(config-filter-control-classmap)# match mac-address aabb.cc00.6500`	(Optional) Creates a condition that evaluates true based on an event's MAC address.
Step 12	{**match** | **no-match**} **method** {**dot1x** | **mab** | **webauth**} **Example:** `Device(config-filter-control-classmap)# match method dot1x`	(Optional) Creates a condition that evaluates true based on an event's authentication method.

	Command or Action	Purpose
Step 13	{**match** | **no-match**} **port-type** {**l2-port** | **l3-port** | **dot11-port**} **Example:** `Device(config-filter-control-classmap)# match port-type l2-port`	(Optional) Creates a condition that evaluates true based on an event's interface type.
Step 14	{**match** | **no-match**} **result-type** [**method** {**dot1x** | **mab** | **webauth**}] *result-type* **Example:** `Device(config-filter-control-classmap)# match result-type agent-not-found`	(Optional) Creates a condition that evaluates true based on the specified authentication result. • To display the available result types, use the question mark (**?**) online help function.
Step 15	{**match** | **no-match**} **service-template** *template-name* **Example:** `Device(config-filter-control-classmap)# match service-template svc_1`	(Optional) Creates a condition that evaluates true based on an event's service template.
Step 16	{**match** | **no-match**} **tag** *tag-name* **Example:** `Device(config-filter-control-classmap)# match tag tag_1`	(Optional) Creates a condition that evaluates true based on the tag associated with an event.
Step 17	{**match** | **no-match**} **timer** *timer-name* **Example:** `Device(config-filter-control-classmap)# match timer restart`	(Optional) Creates a condition that evaluates true based on an event's timer.
Step 18	{**match** | **no-match**} **username** *username* **Example:** `Device(config-filter-control-classmap)# match username josmiths`	(Optional) Creates a condition that evaluates true based on an event's username.
Step 19	**end** **Example:** `Device(config-filter-control-classmap)# end`	(Optional) Exits control class-map filter configuration mode and returns to privileged EXEC mode.
Step 20	**show class-map type control subscriber** {**all** | **name** *control-class-name*} **Example:** `Device# show class-map type control subscriber all`	(Optional) Displays information about Identity-Based Networking Services control classes.

Example: Control Class

The following example shows a control class that is configured with two match conditions:

```
class-map type control subscriber match-all DOT1X_NO_AGENT
 match method dot1x
 match result-type agent-not-found
```

Configure a Control Policy

Control policies determine the actions that the system takes in response to specified events and conditions. The control policy contains one or more control policy rules that associate a control class with one or more actions. The actions that you can configure in a policy rule depend on the type of event that you specify.

Note This task includes all of the actions that you can configure in a control policy regardless of the event. All of these actions, and their corresponding steps, are optional (steps 6 through 21 below). To display the supported actions for a particular event, use the question mark (**?**) online help function.

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	**policy-map type control subscriber** *control-policy-name* **Example:** `Device(config)# policy-map type control PMAP_DefaultWiredDot1xClosedAuth_1X_MAB`	Defines a control policy for subscriber sessions.
Step 4	**event** *event-name* [**match-all** | **match-first**] **Example:** `Device(config-event-control-policymap)# event session-started match-all`	Specifies the type of event that triggers actions in a control policy if conditions are met. • **match-all** is the default behavior. • To display the available event types, use the question mark (**?**) online help function. For a complete description of event types, see the **event** command.

	Command or Action	Purpose
Step 5	*priority-number* **class** {*control-class-name* | **always**} [**do-all** | **do-until-failure** | **do-until-success**] **Example:** `Device(config-class-control-policymap)# 10 class always do-until-failure`	Associates a control class with one or more actions in a control policy. • A named control class must first be configured before specifying it with the *control-class-name* argument. • **do-until-failure** is the default behavior.
Step 6	*action-number* **activate** {**policy type control subscriber** *control-policy-name* [**child** [**no-propagation** | **concurrent**] | **service-template** *template-name* [**aaa-list** *list-name*] [**precedence** *number*] [**replace-all**]} **Example:** `Device(config-action-control-policymap)# 10 activate service-template DefaultCriticalAuthVlan_SRV_TEMPLATE`	(Optional) Activates a control policy or service template on a subscriber session.
Step 7	*action-number* **authenticate using** {**dot1x** | **mab** | **webauth**} [**aaa** {**authc-list** *authc-list-name* | **authz-list** *authz-list-name*]} [**merge**] [**parameter-map** *map-name*] [**priority** *priority-number*] [**replace** | **replace-all**] [**retries** *number* {**retry-time** *seconds*}] **Example:** `Device(config-action-control-policymap)# 20 authenticate using dot1x retries 2 retry-time 0 priority 10`	(Optional) Initiates the authentication of a subscriber session using the specified method.
Step 8	*action-number* **authentication-restart** *seconds* **Example:** `Device(config-action-control-policymap)# 20 authentication-restart 60`	(Optional) Sets a timer to restart the authentication process after an authentication or authorization failure.
Step 9	*action-number* **authorize** **Example:** `Device(config-action-control-policymap)# 30 authorize`	(Optional) Initiates the authorization of a subscriber session.
Step 10	*action-number* **clear-authenticated-data-hosts-on-port** **Example:** `Device(config-action-control-policymap)# 20 clear-authenticated-data-hosts-on-port`	(Optional) Clears authenticated data hosts on a port after an authentication failure.

	Command or Action	Purpose
Step 11	*action-number* **clear-session** **Example:** `Device(config-action-control-policymap)# 10 clear-session`	(Optional) Clears an active subscriber session.
Step 12	*action-number* **deactivate** {**policy type control subscriber** *control-policy-name* | **service-template** *template-name*} **Example:** `Device(config-action-control-policymap)# 20 deactivate service-template`	(Optional) Deactivates a control policy or service template on a subscriber session.
Step 13	*action-number* **err-disable** **Example:** `Device(config-action-control-policymap)# 10 err-disable`	(Optional)Temporarily disables a port after a session violation event.
Step 14	*action-number* **pause reauthentication** **Example:** `Device(config-action-control-policymap)# 40 pause reauthentication`	(Optional) Pauses reauthentication after an authentication failure.
Step 15	*action-number* **protect** **Example:** `Device(config-action-control-policymap)# 10 protect`	(Optional) Silently drops violating packets after a session violation event.
Step 16	*action-number* **replace** **Example:** `Device(config-action-control-policymap)# 10 replace`	(Optional) Clears the existing session and creates a new session after a violation event.
Step 17	*action-number* **restrict** **Example:** `Device(config-action-control-policymap)# 10 restrict`	(Optional) Drops violating packets and generates a syslog entry after a session violation event.
Step 18	*action-number* **resume reauthentication** **Example:** `Device(config-action-control-policymap)# 10 resume reauthentication`	(Optional) Resumes the reauthentication process after an authentication failure.
Step 19	*action-number* **set-timer** *timer-name seconds* **Example:** `Device(config-action-control-policymap)# 20 set-timer RESTART 60`	(Optional) Starts a named policy timer.

	Command or Action	Purpose
Step 20	*action-number* **terminate** {**dot1x** | **mab** | **webauth**} **Example:** `Device(config-action-control-policymap)# 10 terminate mab`	(Optional) Terminates an authentication method on a subscriber session.
Step 21	*action-number* **unauthorize** **Example:** `Device(config-action-control-policymap)# 20 unauthorize`	(Optional) Removes all authorization data from a subscriber session.
Step 22	**end** **Example:** `Device(config-action-control-policymap)# end`	(Optional) Exits control policy-map action configuration mode and returns to privileged EXEC mode.
Step 23	**show policy-map type control subscriber** {**all** | **name** *control-policy-name*} **Example:** `Device# show policy-map type control subscriber name PMAP_DefaultWiredDot1xClosedAuth_1X_MAB`	(Optional) Displays information about identity control policies.

Example: Control Policy

The following example shows a simple control policy with the minimum configuration necessary for initiating authentication:

```
policy-map type control subscriber POLICY_1
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using dot1x
```

Configure Interface Templates

You can create an interface template using the **template** command in global configuration mode. In template configuration mode, enter the required commands. The following commands can be entered in template configuration mode:

Note

- System builtin templates are not displayed in the running configuration. These templates show up in the running configuration only if you edit them.
- When you configure an interface template, we recommend that you enter all the required dependent commands on the same template. we do not recommend to configure the dependent commands on two different templates.

Command	Description
access-session	Configures access session specific interface commands.
authentication	Configures authentication manager Interface Configuration commands.
carrier-delay	Configures delay for interface transitions.
dampening	Enables event dampening.
default	Sets a command to its defaults.
description	Configures interface-specific description.
dot1x	Configures interface configuration commands for IEEE 802.1X.
hold-queue	Sets hold queue depth.
ip	Configures IP template.
keepalive	Enables keepalive.
load-interval	Specifies interval for load calculation for an interface.
mab	Configures MAC authentication bypass Interface.
peer	Configures peer parameters for point to point interfaces.
service-policy	Configures CPL service policy.
source	Gets configurations from another source.
spanning-tree	Configures spanning tree subsystem.
storm-control	Configures storm control.
subscriber	Configures subscriber inactivity timeout value.
switchport	Sets switching mode configurations.
trust	Sets trust value for the interface.

To configure interface templates, perform this task:

Procedure

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**template** *name*
	Example:
	`Device(config)# template DefaultWiredDot1xClosedAuth`
	
	`dot1x pae authenticator`
	` dot1x timeout supp-timeout 7`
	` dot1x max-req 3`
	` switchport mode access`
	` switchport voice vlan 2046`
	` mab`
	` access-session closed`
	` access-session port-control auto`
	` authentication periodic`
	` authentication timer reauthenticate server`
	` service-policy type control subscriber PMAP_DefaultWiredDot1xClosedAuth_1X_MAB`
		Creates a user template and enters template configuration mode.
		Note Builtin template are system-generated.

Step 4	**end** **Example:** `Device(config-template)# end`	Returns to privileged EXEC mode.

Enabling Central Web Authentication

Web authentication allows users to get authenticated through a web browser on a client, with minimal configuration on the client side. Central web authentication is typically used for guest authentication. A RADIUS server (such as Cisco ISE) is mandatory when you enable central web authentication.

Perform the following task on the fabric edge node to redirect the clients based on the HTTP traffic.

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	**ip http server** **Example:**	Enables the HTTP server. The web-based authentication feature uses the HTTP server to

	Command or Action	Purpose
	`Device(config)# ip http server`	communicate with the hosts for user authentication.
Step 4	**end** **Example:** `Device(config)#end`	Returns to privileged EXEC mode.

Create Extended Named ACLs

Follow these steps to create an extended ACL using names:

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	**ip access-list extended** *name* **Example:** `Device(config)# ip access-list extended ACL_WEBAUTH_REDIRECT`	Defines an extended IPv4 access list using a name, and enter access-list configuration mode. The name can be a number from 100 to 199.

Step 4	*sequence-number* {**deny** | **permit**} *protocol* {*source* [*source-wildcard*] | **host** *source* | **any**} {*destination* [*destination-wildcard*] | host *destination* | **any**} [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log**] [**time-range** *time-range-name*]
	Example:
	`Device(config-ext-nacl)# 260 deny ip any host 172.16.2.1`
	`Device(config-ext-nacl)# 500 permit tcp any any eq www`
	`Device(config-ext-nacl)# 600 permit tcp any any eq 443`
	`Device(config-ext-nacl)# 700 permit tcp any any eq 8443`
	`Device(config-ext-nacl)# 800 deny udp any any eq domain`
	`Device(config-ext-nacl)# 900 deny udp any eq bootpc any eq bootps`
		In access-list configuration mode, specify the sequence number (1 to 32767) and the conditions that are to be allowed or denied. Use the **log** keyword to get access list logging messages, including violations.
		• **host** *source*: A source and source wildcard of *source* 0.0.0.0.
		• **host** *destintation*: A destination and destination wildcard of *destination* 0.0.0.0.
		• **any**: A source and source wildcard or destination and destination wildcard of 0.0.0.0 255.255.255.255.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-ext-nacl)# end`	Exits access-list configuration mode and returns to privileged EXEC mode.

When you are creating extended ACLs, remember that, by default, the end of the ACL contains an implicit deny statement for everything if it did not find a match before reaching the end. For standard ACLs, if you omit the mask from an associated IP host address access list specification, 0.0.0.0 is assumed to be the mask.

After you create an ACL, any additions are placed at the end of the list. You cannot selectively add ACL entries to a specific ACL. However, you can use **no permit** and **no deny** access-list configuration mode commands to remove entries from a named ACL.

Being able to selectively remove lines from a named ACL is one reason you might use named ACLs instead of numbered ACLs.

What to do next

After creating a named ACL, you can apply it to interfaces or to VLANs .

Configure IPv6 ACLs

To filter IPv6 traffic, perform this procedure.

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	**ipv6 access-list** {*list-name* | **log-update threshold** | **role-based** *list-name*} **Example:** `Device(config)# ipv6 access-list IPV6_PRE_AUTH_ACL`	Defines an IPv6 ACL name, and enters IPv6 access list configuration mode.
Step 4	*sequence-number* {**deny** | **permit**} protocol {*source-ipv6-prefix/* |*prefix-length* |**any threshold**| **host** *source-ipv6-address*} [operator [*port-number*]] { *destination-ipv6-prefix/ prefix-length* | **any** | **host** *destination-ipv6-address*} [operator	• For protocol, enter the name or number of an IP: **ahp**, **esp**, **icmp**, **ipv6**, **pcp**, **stcp**, **tcp**, or **udp**, or an integer in the range 0 to 255 representing an IPv6 protocol number.

	Command or Action	Purpose

	[*port-number*]][**dscp** *value*] [**fragments**] [**log**] [**log-input**][**sequence** *value*] [**time-range** *name*]
	Example:
	`Device(config-ipv6-acl)# sequence 10 permit udp any any eq bootps`
	`Device(config-ipv6-acl)# sequence 20 permit udp any any eq bootpc`
	`Device(config-ipv6-acl)# sequence 30 permit udp any any eq domain`
	`Device(config-ipv6-acl)# sequence 40 deny ipv6 any any`
		• The *source-ipv6-prefix/prefix-length* or *destination-ipv6-prefix/ prefix-length* is the source or destination IPv6 network or class of networks for which to set deny or permit conditions, specified in hexadecimal and using 16-bit values between colons (see RFC 2373).
		• Enter **any** as an abbreviation for the IPv6 prefix ::/0.
		• For **host** *source-ipv6-address* or *destination-ipv6-address*, enter the source or destination IPv6 host address for which to set deny or permit conditions, specified in hexadecimal using 16-bit values between colons.
		• (Optional) For operator, specify an operand that compares the source or destination ports of the specified protocol. Operands are **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range**.
		If the operator follows the *source-ipv6-prefix/prefix-length* argument, it must match the source port. If the operator follows the *destination-ipv6-prefix/prefix-length* argument, it must match the destination port.
		• (Optional) The **port-number** is a decimal number from 0 to 65535 or the name of a TCP or UDP port. You can use TCP port names only when filtering TCP. You can use UDP port names only when filtering UDP.
		• (Optional) Enter **dscp** value to match a differentiated services code point value against the traffic class value in the Traffic Class field of each IPv6 packet header. The acceptable range is from 0 to 63.
		• (Optional) Enter **fragments** to check noninitial fragments. This keyword is visible only if the protocol is ipv6.
		• (Optional) Enter **log** to cause an logging message to be sent to the console about the packet that matches the entry. Enter **log-input** to include the input interface in

	Command or Action	Purpose
		the log entry. Logging is supported only for router ACLs. • (Optional) Enter **sequence** *value* to specify the sequence number for the access list statement. The acceptable range is from 1 to 4,294,967,295. • (Optional) Enter **time-range** name to specify the time range that applies to the deny or permit statement.
Step 5	**end** **Example:** `Device(config-ipv6-acl)# end`	Exits IPv6 access list configuration mode and returns to privileged EXEC mode.
Step 6	**show ipv6 access-list** **Example:** `Device# show ipv6 access-list`	Verifies that IPv6 ACLs are configured correctly.

Configure Host Onboarding Interfaces

To configure host onboarding interfaces, perform this task:

Note The example configurations in this procedure are for Closed Authentication mode on the interface.

You can follow the same procedure for the Open Authentication and Low Impact authentication modes on the interface. Whatever interface configuration mode you deploy, ensure you use the respective dot1x interface template (DefaultWiredDot1xOpenAuth or DefaultWiredDot1xLowImpactAuth).

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	**interface** *type number* **Example:**	Specifies the interface type and number and enters interface configuration mode.

	Command or Action	Purpose
	`Device(config)# interface GigabitEthernet1/0/10`	
Step 4	**switchport access vlan** *vlan-id* **Example:** `Device(config-if)# switchport access vlan 50`	Assigns the port to a VLAN. Valid VLAN IDs are 1 to 4094.
Step 5	**switchport mode access** **Example:** `Device(config-if)# switchport mode access`	Defines the VLAN membership mode for the port (Layer 2 access port).
Step 6	**switchport voice vlan** *vlan-id* **Example:** `Device(config-if)# switchport voice vlan 51`	Configures the voice VLAN. Valid VLAN IDs are 1 to 4094.
Step 7	**device-tracking attach-policy** *policy_name* **Example:** `Device(config-if)# device-tracking attach-policy IPDT_POLICY`	Attaches the device tracking policy to the specified VLANs across all switch interfaces.
Step 8	**load-interval** *seconds* **Example:** `Device(config-if)# load-interval 30`	Changes the length of time for which data is used to compute load statistics. Value is a multiple of 30, from 30 to 600 (30, 60, 90, 120, and so on). The default is 300 seconds.
Step 9	**access-session inherit disable interface-template-sticky** **Example:** `Device(config-if)# access-session inherit disable interface-template-sticky`	Disables the Autoconf feature on a specific interface.
Step 10	**access-session inherit disable autoconf** **Example:** `Device(config-if)# access-session inherit disable autoconf`	Manually disables Autoconf at the interface level, even when Autoconf is enabled at the global level.
Step 11	**dot1x timeout tx-period** *seconds* **Example:** `Device(config-if)# dot1x timeout tx-period 7`	Configures the number of seconds between retransmission of EAP request ID packets (assuming that no response is received) to the client. The range is from 1 to 65535. The default is 30.
Step 12	**dot1x max-reauth-req** *number* **Example:**	Sets the maximum number of times the authenticator sends an Extensible Authentication Protocol (EAP) request/identity

	Command or Action	Purpose
	`Device(config-if)# dot1x max-reauth-req 3`	frame (assuming that no response is received) to the client. The range is 1 through 10. The default is 2.
Step 13	**no macro auto processing** **Example:** `Device(config-if)# no macro auto processing`	Disables Auto Smartports macros on an interface.
Step 14	**source template** *template* **Example:** `Device(config-if)# source template DefaultWiredDot1xClosedAuth`	Sources the interface template along with the other interface-specific commands for the desired ports. This example is for a Closed Authentication mode of 802.1x deployment. You can also use the Open Authentication or Low Impact authentication modes on the interface. Whatever authentication mode you deploy, ensure you use the correct dot1x interface template (DefaultWiredDot1xOpenAuth or DefaultWiredDot1xLowImpactAuth, which were defined earlier).
Step 15	**spanning-tree portfast** **Example:** `Device(config-if)# spanning-tree portfast`	Enables PortFast mode, where the interface is immediately put into the forwarding state upon linkup without waiting for the timer to expire.
Step 16	**spanning-tree bpduguard enable** **Example:** `Device(config-if)# spanning-tree bpduguard enable`	Enables bridge protocol data unit (BPDU) guard on the interface.
Step 17	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuration Example for IEEE 802.1x on Fabric Edge

A fabric edge node is configured as an authenticator to interface with the AAA server or Cisco ISE and authenticate the endpoints. This is a sample configuration for IEEE 802.1x on a fabric edge node; Cisco ISE is configured with an IP address of 172.16.2.1

```
username admin privilege 15 password 7 user-password
enable secret level 1 secret-pwd
!
aaa new-model
dot1x system-auth-control
```

```
aaa session-id common
!
aaa authentication login default local
aaa authentication login cts-list group client-radius-group local
aaa authentication dot1x default group client-radius-group
aaa authorization exec default local
aaa authorization network default group client-radius-group
aaa authorization network cts-list group client-radius-group
aaa accounting Identity default start-stop group client-radius-group
aaa accounting update newinfo periodic 2880
!
aaa server radius dynamic-author
 client 172.16.2.1 server-key 7 server-pwd
!
!
radius server radius_172.16.2.1
 address ipv4 172.16.2.1 auth-port 1812 acct-port 1813
 timeout 2
 retransmit 1
 automate-tester username dummy ignore-acct-port probe-on
 pac key 7 pac-key
!
!
radius-server attribute 6 on-for-login-auth
radius-server attribute 6 support-multiple
radius-server attribute 8 include-in-access-req
radius-server attribute 25 access-request include
radius-server attribute 31 mac format ietf upper-case
radius-server attribute 31 send nas-port-detail mac-only
radius-server dead-criteria time 5 tries 3
radius-server deadtime 3
!
aaa group server radius client-radius-group
 server name radius_172.16.2.1
 ip radius source-interface Loopback0
!
!
!
!
!
ip radius source-interface Loopback0
Identify Based Networking Services(IBNS)
class-map type control subscriber match-all AAA_SVR_DOWN_AUTHD_HOST
 match authorization-status authorized
 match result-type aaa-timeout
!
class-map type control subscriber match-all AAA_SVR_DOWN_UNAUTHD_HOST
 match authorization-status unauthorized
 match result-type aaa-timeout
!
class-map type control subscriber match-all AUTHC_SUCCESS-AUTHZ_FAIL
 match authorization-status unauthorized
 match result-type success
!
class-map type control subscriber match-all DOT1X
 match method dot1x
!
class-map type control subscriber match-all DOT1X_FAILED
 match method dot1x
 match result-type method dot1x authoritative
!
class-map type control subscriber match-all DOT1X_MEDIUM_PRIO
 match authorizing-method-priority gt 20
!
```

```
class-map type control subscriber match-all DOT1X_NO_RESP
 match method dot1x
 match result-type method dot1x agent-not-found
!
class-map type control subscriber match-all DOT1X_TIMEOUT
 match method dot1x
 match result-type method dot1x method-timeout
!
class-map type control subscriber match-any IN_CRITICAL_AUTH
 match activated-service-template DefaultCriticalVoice_SRV_TEMPLATE
!
class-map type control subscriber match-any IN_CRITICAL_AUTH_CLOSED_MODE
 match activated-service-template DefaultCriticalAuthVlan_SRV_TEMPLATE
 match activated-service-template DefaultCriticalVoice_SRV_TEMPLATE
!
class-map type control subscriber match-all MAB
 match method mab
!
class-map type control subscriber match-all MAB_FAILED
 match method mab
 match result-type method mab authoritative
!
class-map type control subscriber match-none NOT_IN_CRITICAL_AUTH
 match activated-service-template DefaultCriticalVoice_SRV_TEMPLATE
!
class-map type control subscriber match-none NOT_IN_CRITICAL_AUTH_CLOSED_MODE
 match activated-service-template DefaultCriticalAuthVlan_SRV_TEMPLATE
 match activated-service-template DefaultCriticalVoice_SRV_TEMPLATE
!
policy-map type control subscriber PMAP_DefaultWiredDot1xClosedAuth_1X_MAB
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using dot1x retries 2 retry-time 0 priority 10
 event authentication-failure match-first
  5 class DOT1X_FAILED do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  10 class AAA_SVR_DOWN_UNAUTHD_HOST do-until-failure
   10 activate service-template DefaultCriticalAuthVlan_SRV_TEMPLATE
   20 activate service-template DefaultCriticalVoice_SRV_TEMPLATE
   30 authorize
   40 pause reauthentication
  20 class AAA_SVR_DOWN_AUTHD_HOST do-until-failure
   10 pause reauthentication
   20 authorize
  30 class DOT1X_NO_RESP do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  40 class MAB_FAILED do-until-failure
   10 terminate mab
   20 authentication-restart 60
  50 class DOT1X_TIMEOUT do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  60 class always do-until-failure
   10 terminate dot1x
   20 terminate mab
   30 authentication-restart 60
 event aaa-available match-all
  10 class IN_CRITICAL_AUTH_CLOSED_MODE do-until-failure
   10 clear-session
  20 class NOT_IN_CRITICAL_AUTH_CLOSED_MODE do-until-failure
   10 resume reauthentication
 event agent-found match-all
```

```
  10 class always do-until-failure
   10 terminate mab
   20 authenticate using dot1x retries 2 retry-time 0 priority 10
 event inactivity-timeout match-all
  10 class always do-until-failure
   10 clear-session
 event authentication-success match-all
 event violation match-all
  10 class always do-until-failure
   10 restrict
 event authorization-failure match-all
  10 class AUTHC_SUCCESS-AUTHZ_FAIL do-until-failure
   10 authentication-restart 60
!
policy-map type control subscriber PMAP_DefaultWiredDot1xClosedAuth_MAB_1X
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using mab priority 20
 event authentication-failure match-first
  5 class DOT1X_FAILED do-until-failure
   10 terminate dot1x
   20 authentication-restart 60
  10 class AAA_SVR_DOWN_UNAUTHD_HOST do-until-failure
   10 activate service-template DefaultCriticalAuthVlan_SRV_TEMPLATE
   20 activate service-template DefaultCriticalVoice_SRV_TEMPLATE
   30 authorize
   40 pause reauthentication
  20 class AAA_SVR_DOWN_AUTHD_HOST do-until-failure
   10 pause reauthentication
   20 authorize
  30 class MAB_FAILED do-until-failure
   10 terminate mab
   20 authenticate using dot1x retries 2 retry-time 0 priority 10
  40 class DOT1X_NO_RESP do-until-failure
   10 terminate dot1x
   20 authentication-restart 60
  50 class DOT1X_TIMEOUT do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  60 class always do-until-failure
   10 terminate mab
   20 terminate dot1x
   30 authentication-restart 60
 event aaa-available match-all
  10 class IN_CRITICAL_AUTH_CLOSED_MODE do-until-failure
   10 clear-session
  20 class NOT_IN_CRITICAL_AUTH_CLOSED_MODE do-until-failure
   10 resume reauthentication
 event agent-found match-all
  10 class always do-until-failure
   10 terminate mab
   20 authenticate using dot1x retries 2 retry-time 0 priority 10
 event inactivity-timeout match-all
  10 class always do-until-failure
   10 clear-session
 event authentication-success match-all
 event violation match-all
  10 class always do-until-failure
   10 restrict
 event authorization-failure match-all
  10 class AUTHC_SUCCESS-AUTHZ_FAIL do-until-failure
   10 authentication-restart 60
!
policy-map type control subscriber PMAP_DefaultWiredDot1xLowImpactAuth_1X_MAB
```

```
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using dot1x retries 2 retry-time 0 priority 10
 event authentication-failure match-first
  5 class DOT1X_FAILED do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  10 class AAA_SVR_DOWN_UNAUTHD_HOST do-until-failure
   10 activate service-template DefaultCriticalAuthVlan_SRV_TEMPLATE
   20 activate service-template DefaultCriticalVoice_SRV_TEMPLATE
   25 activate service-template DefaultCriticalAccess_SRV_TEMPLATE
   30 authorize
   40 pause reauthentication
  20 class AAA_SVR_DOWN_AUTHD_HOST do-until-failure
   10 pause reauthentication
   20 authorize
  30 class DOT1X_NO_RESP do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  40 class MAB_FAILED do-until-failure
   10 terminate mab
   20 authentication-restart 60
  50 class DOT1X_TIMEOUT do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  60 class always do-until-failure
   10 terminate dot1x
   20 terminate mab
   30 authentication-restart 60
 event aaa-available match-all
  10 class IN_CRITICAL_AUTH do-until-failure
   10 clear-session
  20 class NOT_IN_CRITICAL_AUTH do-until-failure
   10 resume reauthentication
 event agent-found match-all
  10 class always do-until-failure
   10 terminate mab
   20 authenticate using dot1x retries 2 retry-time 0 priority 10
 event inactivity-timeout match-all
  10 class always do-until-failure
   10 clear-session
 event authentication-success match-all
 event violation match-all
  10 class always do-until-failure
   10 restrict
 event authorization-failure match-all
  10 class AUTHC_SUCCESS-AUTHZ_FAIL do-until-failure
   10 authentication-restart 60
!
policy-map type control subscriber PMAP_DefaultWiredDot1xLowImpactAuth_MAB_1X
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using mab priority 20
 event authentication-failure match-first
  5 class DOT1X_FAILED do-until-failure
   10 terminate dot1x
   20 authentication-restart 60
  10 class AAA_SVR_DOWN_UNAUTHD_HOST do-until-failure
   10 activate service-template DefaultCriticalAuthVlan_SRV_TEMPLATE
   20 activate service-template DefaultCriticalVoice_SRV_TEMPLATE
   25 activate service-template DefaultCriticalAccess_SRV_TEMPLATE
   30 authorize
   40 pause reauthentication
  20 class AAA_SVR_DOWN_AUTHD_HOST do-until-failure
```

```
   10 pause reauthentication
   20 authorize
  30 class MAB_FAILED do-until-failure
   10 terminate mab
   20 authenticate using dot1x retries 2 retry-time 0 priority 10
  40 class DOT1X_NO_RESP do-until-failure
   10 terminate dot1x
   20 authentication-restart 60
  50 class DOT1X_TIMEOUT do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  60 class always do-until-failure
   10 terminate mab
   20 terminate dot1x
   30 authentication-restart 60
 event aaa-available match-all
  10 class IN_CRITICAL_AUTH do-until-failure
   10 clear-session
  20 class NOT_IN_CRITICAL_AUTH do-until-failure
   10 resume reauthentication
 event agent-found match-all
  10 class always do-until-failure
   10 terminate mab
   20 authenticate using dot1x retries 2 retry-time 0 priority 10
 event inactivity-timeout match-all
  10 class always do-until-failure
   10 clear-session
 event authentication-success match-all
 event violation match-all
  10 class always do-until-failure
   10 restrict
 event authorization-failure match-all
  10 class AUTHC_SUCCESS-AUTHZ_FAIL do-until-failure
   10 authentication-restart 60
!
policy-map type control subscriber PMAP_DefaultWiredDot1xOpenAuth_1X_MAB
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using dot1x retries 2 retry-time 0 priority 10
 event authentication-failure match-first
  5 class DOT1X_FAILED do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  10 class AAA_SVR_DOWN_UNAUTHD_HOST do-until-failure
   10 activate service-template DefaultCriticalAuthVlan_SRV_TEMPLATE
   20 activate service-template DefaultCriticalVoice_SRV_TEMPLATE
   30 authorize
   40 pause reauthentication
  20 class AAA_SVR_DOWN_AUTHD_HOST do-until-failure
   10 pause reauthentication
   20 authorize
  30 class DOT1X_NO_RESP do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  40 class MAB_FAILED do-until-failure
   10 terminate mab
   20 authentication-restart 60
  50 class DOT1X_TIMEOUT do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  60 class always do-until-failure
   10 terminate dot1x
   20 terminate mab
   30 authentication-restart 60
```

```
 event aaa-available match-all
  10 class IN_CRITICAL_AUTH do-until-failure
   10 clear-session
  20 class NOT_IN_CRITICAL_AUTH do-until-failure
   10 resume reauthentication
 event agent-found match-all
  10 class always do-until-failure
   10 terminate mab
   20 authenticate using dot1x retries 2 retry-time 0 priority 10
 event inactivity-timeout match-all
  10 class always do-until-failure
   10 clear-session
 event authentication-success match-all
 event violation match-all
  10 class always do-until-failure
   10 restrict
 event authorization-failure match-all
  10 class AUTHC_SUCCESS-AUTHZ_FAIL do-until-failure
   10 authentication-restart 60
!
policy-map type control subscriber PMAP_DefaultWiredDot1xOpenAuth_MAB_1X
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using mab priority 20
 event authentication-failure match-first
  5 class DOT1X_FAILED do-until-failure
   10 terminate dot1x
   20 authentication-restart 60
  10 class AAA_SVR_DOWN_UNAUTHD_HOST do-until-failure
   10 activate service-template DefaultCriticalAuthVlan_SRV_TEMPLATE
   20 activate service-template DefaultCriticalVoice_SRV_TEMPLATE
   30 authorize
   40 pause reauthentication
  20 class AAA_SVR_DOWN_AUTHD_HOST do-until-failure
   10 pause reauthentication
   20 authorize
  30 class MAB_FAILED do-until-failure
   10 terminate mab
   20 authenticate using dot1x retries 2 retry-time 0 priority 10
  40 class DOT1X_NO_RESP do-until-failure
   10 terminate dot1x
   20 authentication-restart 60
  50 class DOT1X_TIMEOUT do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  60 class always do-until-failure
   10 terminate mab
   20 terminate dot1x
   30 authentication-restart 60
 event aaa-available match-all
  10 class IN_CRITICAL_AUTH do-until-failure
   10 clear-session
  20 class NOT_IN_CRITICAL_AUTH do-until-failure
   10 resume reauthentication
 event agent-found match-all
  10 class always do-until-failure
   10 terminate mab
   20 authenticate using dot1x retries 2 retry-time 0 priority 10
 event inactivity-timeout match-all
  10 class always do-until-failure
   10 clear-session
 event authentication-success match-all
 event violation match-all
  10 class always do-until-failure
```

```
   10 restrict
 event authorization-failure match-all
  10 class AUTHC_SUCCESS-AUTHZ_FAIL do-until-failure
   10 authentication-restart 60
!
!
template DefaultWiredDot1xClosedAuth
 dot1x pae authenticator
 dot1x timeout supp-timeout 7
 dot1x max-req 3
 switchport mode access
 switchport voice vlan 2046
 mab
 access-session closed
 access-session port-control auto
 authentication periodic
 authentication timer reauthenticate server
 service-policy type control subscriber PMAP_DefaultWiredDot1xClosedAuth_1X_MAB
!
template DefaultWiredDot1xLowImpactAuth
 dot1x pae authenticator
 dot1x timeout supp-timeout 7
 dot1x max-req 3
 switchport mode access
 switchport voice vlan 2046
 mab
 access-session port-control auto
 authentication periodic
 authentication timer reauthenticate server
 service-policy type control subscriber PMAP_DefaultWiredDot1xLowImpactAuth_1X_MAB
!
template DefaultWiredDot1xOpenAuth
 dot1x pae authenticator
 dot1x timeout supp-timeout 7
 dot1x max-req 3
 switchport mode access
 switchport voice vlan 2046
 mab
 access-session port-control auto
 authentication periodic
 authentication timer reauthenticate server
 service-policy type control subscriber PMAP_DefaultWiredDot1xOpenAuth_1X_MAB
!
!
ip access-list extended ACL_WEBAUTH_REDIRECT
 260 deny ip any host 172.16.2.1
 500 permit tcp any any eq www
 600 permit tcp any any eq 443
 700 permit tcp any any eq 8443
 800 deny udp any any eq domain
 900 deny udp any eq bootpc any eq bootps
ip access-list extended IPV4_CRITICAL_AUTH_ACL
 10 permit ip any any
ip access-list extended IPV4_PRE_AUTH_ACL
 10 permit udp any any eq bootps
 20 permit udp any any eq bootpc
 30 permit udp any any eq domain
 40 deny ip any any
!
!
ipv6 access-list IPV6_CRITICAL_AUTH_ACL
 sequence 10 permit ipv6 any any
!
ipv6 access-list IPV6_PRE_AUTH_ACL
```

```
 sequence 10 permit udp any any eq bootps
 sequence 20 permit udp any any eq bootpc
 sequence 30 permit udp any any eq domain
 sequence 40 deny ipv6 any any
!
Host onboarding interfaces
interface GigabitEthernet1/0/10
 switchport access vlan 50
 switchport mode access
 switchport voice vlan 51
 device-tracking attach-policy IPDT_POLICY
 load-interval 30
 access-session inherit disable interface-template-sticky
 access-session inherit disable autoconf
 dot1x timeout tx-period 7
 dot1x max-reauth-req 3
 no macro auto processing
 source template DefaultWiredDot1xClosedAuth
 spanning-tree portfast
 spanning-tree bpduguard enable
!
interface GigabitEthernet1/0/11
 switchport access vlan 50
 switchport mode access
 switchport voice vlan 51
 device-tracking attach-policy IPDT_POLICY
 load-interval 30
 access-session inherit disable interface-template-sticky
 access-session inherit disable autoconf
 dot1x timeout tx-period 7
 dot1x max-reauth-req 3
 no macro auto processing
 source template DefaultWiredDot1xOpenAuth
 spanning-tree portfast
 spanning-tree bpduguard enable
!
interface GigabitEthernet1/0/12
 switchport access vlan 50
 switchport mode access
 switchport voice vlan 51
 device-tracking attach-policy IPDT_POLICY
 ip access-group IPV4_PRE_AUTH_ACL in
 load-interval 30
 ipv6 traffic-filter IPV6_PRE_AUTH_ACL in
 access-session inherit disable interface-template-sticky
 access-session inherit disable autoconf
 dot1x timeout tx-period 7
 dot1x max-reauth-req 3
 no macro auto processing
 source template DefaultWiredDot1xLowImpactAuth
 spanning-tree portfast
 spanning-tree bpduguard enable
!
interface GigabitEthernet1/0/13
 switchport access vlan 50
 switchport mode access
 switchport voice vlan 51
 device-tracking attach-policy IPDT_POLICY
 load-interval 30
 access-session inherit disable interface-template-sticky
 access-session inherit disable autoconf
 cts manual
  policy static sgt 15
  no propagate sgt
```

```
 no macro auto processing
 spanning-tree portfast
 spanning-tree bpduguard enable
!
interface GigabitEthernet1/0/14
 device-tracking attach-policy IPDT_POLICY
!
```