Implementing MAC Authentication Bypass

This chapter describes the implementation of MAC Authentication Bypass (MAB).

IEEE 802.1X authentication configuration on the router helps to prevent unauthorized end devices from gaining access to the network. However, not all end devices support 802.1X. Hence, we introduce port controlling functionality on these routers using MAC authentication bypass (MAB)—a feature that grants network access to devices based on their MAC addresses, regardless of their 802.1X capability or credentials.

For details of commands related to MAB, see the *802.1X and Port Control Commands* chapter in the *System Security Command Reference for Cisco 8000 Series Routers*.

- MAC Authentication Bypass, on page 1

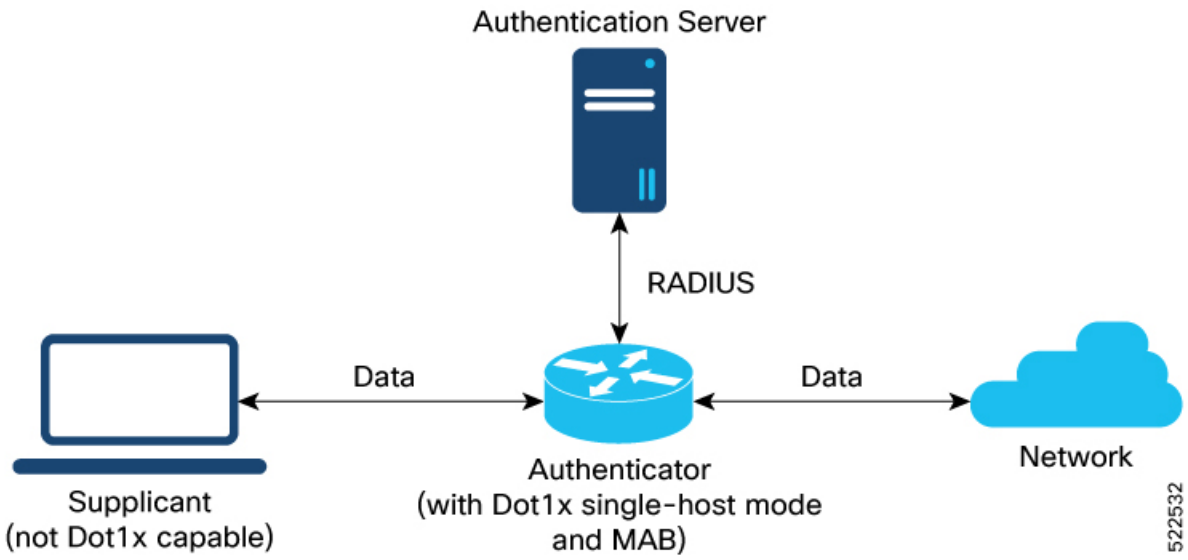
MAC Authentication Bypass

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
Multi-auth MAC Authentication Bypass	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM

Feature Name	Release Information	Feature Description
Multi-auth MAC Authentication Bypass	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: Q100, Q200]) (*select variants only) You can enhance network flexibility by enabling multiple hosts on a single port using MAC Authentication Bypass (MAB). The router now supports up to two clients per port by expanding its MAC learning capability from one to two. It authenticates each MAC address individually, allowing multi-domain authentication and enabling independent management of two endpoints. This feature simplifies network management and increases the connectivity options for devices per port. *This feature is supported on: • 8201-SYS • 8101-32FH
MAC Authentication Bypass	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM

Feature Name	Release Information	Feature Description
MAC Authentication Bypass	Release 7.3.4 Release 7.5.2	Based on the MAC address of the end device or the client connected to the router port, this feature enables port control functionality for your router. This functionality provides controlled access to network services for end devices that do not support other authentication methods such as IEEE 802.1X port-based authentication. The MAB support is only for the **single-host** mode. This feature introduces these commands and options: • **mab** option in the **dot1x profile** command • **mab-retry-time** option in the **authenticator** command • **clear mab** • **show mab**



With MAC authentication bypass (MAB) functionality, the router (authenticator) uses the MAC address of the end device or the client (also called as supplicant) as an authenticating parameter for providing network access. With MAB enabled, when the router receives an incoming data packet from the client that is connected to the router port, it learns the source MAC address and sends it to the external RADIUS server (authentication server) for authentication. The RADIUS authentication server maintains a database of MAC addresses for

devices that require access to the network. Based on the authentication result, the router allows or drops the data packets from that client. If the RADIUS server returns a success (*Access-Accept*) message, it indicates that the MAC address is authenticated and the client is authorized to send traffic through that port. The router then programs that MAC address on the port to which the client is connected. The router allows the traffic from the client to be forwarded to the network. Similarly, if the RADIUS server returns a failure (*Access-Reject*) message, it indicates that the MAC address is unauthenticated. And hence the router drops further data packets from that client. Thus, the MAB feature brings in port control functionality for Cisco 8000 Series Routers and provides end devices a controlled access to network services.

Starting with Cisco IOS-XR Release 24.4.1, MAC Authentication Bypass (MAB) can now have multiple hosts by allowing MAC addresses on a single port, each authenticated separately. The router achieves this functionality by increasing the maximum limit on MAC learning capability from 1 to 2 clients. With this new ability, when **multi-auth** mode is configured under MAB, the router continues MAC -learning on a port after authenticating a client using MAB, until the second client authentication is begun. With this you can use MAB for multi-domain authentication by allowing two endpoints to be authenticated and managed separately on the same port.

Authentication Failure Scenarios with MAB

This table lists various authentication failure scenarios and the expected feature behavior with MAB:

Table 2: Authentication Failure Scenarios with MAB

Authentication Failure Scenarios	Expected MAB Feature Behavior
RADIUS server rejects the authentication request	• The router deletes the client programming on the port (if that client was already authenticated). • Router retries the authentication process twice with the RADIUS server at an interval of 60 seconds, by default. You can configure this interval using the **authenticator timer mab-retry-time** command. • If the server still does not authorize the client, then the router clears the client session and its programming on the port. • The router puts the port back in MAC learning mode to relearn a new MAC address. The router:

Authentication Failure Scenarios	Expected MAB Feature Behavior
RADIUS server is not reachable during authentication process	With **server dead action auth-retry** command configured: • The router retains the programing of the client that was already authenticated. Else, the router deletes it. • Router retries the authentication process with the RADIUS server at an interval of 60 seconds until the server becomes available. You can configure this interval using the **authenticator timer mab-retry-time** command. • The router does not attempt to learn any new MAC address on the port. • To clear the client session and its programming on the router, you must use the **clear mab session** command. • The router puts the port back in MAC learning mode to relearn a new MAC address. Similarly, for an unauthenticated client, if the authentication does not happen after the retries, the router deletes the client context and puts the port back in MAC learning mode. Without **server dead action auth-retry** command configuration: • The router deletes the programming of the client that was already authenticated and retries authentication (as mentioned earlier). • If the client is still not authenticated, the router automatically clears the client session. • The router puts the port back in MAC learning mode to relearn a new MAC address.

Restrictions for MAB

The restrictions apply to the MAB feature:

- With MAB, user authentication can only be done using a remote AAA server; not using the local AAA server on the router.
- MAB feature works only as a standalone feature; not as a fallback mechanism for any other type of authentication failures.
- Until Cisco IOS-XR Release 24.4.1, MAB supports only a single end device on each port. Hence, you must configure the authenticator (the router) to be in **single-host** mode. Starting with Cisco IOS-XR Release 24.4.1, you can configure the authenticator (the router) to be in **single-host** or **multi-auth** mode.
- The Centralized Systems does not support MAB.

Configure MAC Authentication Bypass

Prerequisites

- Configure the remote RADIUS server (using the **radius-server** command), and authentication method with the RADIUS server (using the **aaa authentication dot1x** command) in
- Configure the 802.1X profile (using the **dot1x profile** command in XR Config mode)
- Configure the authenticator (using the **authenticator** command in dot1x profile configuration sub mode) with respective parameters such as:
  - Re-authentication time—**reauth-time**
  - Host mode—as **single-host** or **multi-auth**
  - Retry action for server-unreachable scenarios—**auth-retry** or **auth-fail**

See the *MACSec Using EAP-TLS Authentication* chapter for these configuration details.

See *Running Configuration* section for examples.

To configure MAB, use the **mab** command in dot1x profile configuration sub mode.

Configuration Example for MAB

Enable MAB:

```
Router#configure
Router(config)#dot1x profile test_mab
Router(dot1xx-test_mab)#mab
Router(dot1xx-test_mab)#commit
```

Configure the authenticator retry time for MAB clients:

```
Router#configure
Router(config)#dot1x profile test_mab
Router(dot1xx-test_mab)#authenticator
Router(dot1xx-test_mab-auth)#timer mab-retry-time 60
Router(dot1xx-test_mab-auth)#commit
```

Attach the dot1x profile to the corresponding interface or port on the router.

```
Router(config)#interface GigabitEthernet0/0/0/0
Router(config-intf)#dot1x profile test_mab
Router(config-intf)#commit
```

Running Configuration

```
Router# show running-configuration

!
radius-server host <ip-address> auth-port <auth-port-num> acct-port <acct-port-num>
 key 7 <key>
!
aaa authentication dot1x default group radius
interface GigabitEthernet0/0/0/0
```

```
 dot1x profile test_mab
!

dot1x profile test_mab
 mab
 authenticator
  timer reauth-time 60
  timer mab-retry-time 60
  host-mode single-host
  server dead action auth-retry
 !
!
end
```

Verify MAB Configuration

You can use these **show** commands to verify your MAB configuration:

- To check the MAB summary:

```
Router#show mab summary
Fri Apr  1 16:37:32.340 IST

NODE: node0_0_CPU0
==============================================================================
   Interface-Name        Client          Status
==============================================================================
     Gi0/0/0/0         1122.3344.5566   Authorized
Router#
```

- To verify the detailed MAB status:

```
Router#show mab detail
Fri Apr  1 16:37:37.140 IST

NODE: node0_0_CPU0

MAB info for GigabitEthernet0/0/0/0
------------------------------------------------------------
InterfaceName       : Gi0/0/0/0
InterfaceHandle     : 0x00000060
HostMode            : single-host
PortControl         : Enabled
PuntState           : Stop Success
PuntSummary         : Punt disabled
Client:
  MAC Address       : 1122.3344.5566
  Status            : Authorized
  SM State          : Terminate
  ReauthTimeout     : 60s, Remaining 0 day(s), 00:00:46
  RetryTimeout      : 60s, timer not started yet
  AuthMethod        : PAP (remote)
  LastAuthTime      : 2022 Apr 01 16:37:23.634
  ProgrammingStatus : Add Success
Router#
```

- To verify the MAB interface summary:

```
Router#show mab interface gigabitEthernet 0/0/0/0
Fri Apr  1 16:38:27.715 IST
==============================================================================
   Interface-Name        Client          Status
```

```
=====================================================================
     Gi0/0/0/0         1122.3344.5566   Authorized
```

- To verify the MAB interface details:

```
Router#show mab interface gigabitEthernet 0/0/0/0 detail
Fri Apr  1 16:38:31.543 IST
MAB info for GigabitEthernet0/0/0/0
-------------------------------------------------------------------
InterfaceName       : Gi0/0/0/0
InterfaceHandle     : 0x00000060
HostMode            : single-host
PortControl         : Enabled
PuntState           : Stop Success
PuntSummary         : Punt disabled
Client:
  MAC Address       : 1122.3344.5566
  Status            : Authorized
  SM State          : Terminate
  ReauthTimeout     : 60s, Remaining 0 day(s), 00:00:51
  RetryTimeout      : 60s, timer not started yet
  AuthMethod        : PAP (remote)
  LastAuthTime      : 2022 Apr 01 16:38:23.640
  ProgrammingStatus : Add Success
Router#
```

- To verify the MAB interface statistics:

```
Router#show mab statistics interface gigabitEthernet 0/0/0/0
Fri Apr  1 16:41:23.011 IST
InterfaceName         : GigabitEthernet0/0/0/0
-------------------------------------------------------------------
MAC Learning:
  RxTotal             : 0
  RxNoSrcMac          : 0
  RxNoIdb             : 0
Port Control:
  EnableSuccess       : 1
  EnableFail          : 0
  UpdateSuccess       : 0
  UpdateFail          : 0
  PuntStartSuccess    : 0
  PuntStartFail       : 0
  PuntStopSuccess     : 1
  PuntStopFail        : 0
  AddClientSuccess    : 1
  AddClientFail       : 0
  RemoveClientSuccess : 0
  RemoveClientFail    : 0
Client                :
  MAC Address         : 1122.3344.5566
  Authentication:
    Success           : 1406
    Fail              : 0
    Timeout           : 0
    AAA Unreachable   : 0
Router#
```

System Logs for MAB

The router displays the following system logs on the console in various MAB scenarios:

- When the dot1x profile is applied on the port, with MAB feature enabled:

  Success case:

```
%L2-DOT1X-5-PORT_CONTROL_ENABLE_SUCCESS : Hu0/0/1/0 : Port Control Enabled with
Single-Host mode
```

  Failure case:

```
%L2-DOT1X-5-PORT_CONTROL_ENABLE_FAILURE : Hu0/0/1/0 : Failed to enable port-control
```

- When the dot1x profile is removed from the interface:

  Success case:

```
%L2-DOT1X-5-PORT_CONTROL_DISABLE_SUCCESS : Hu0/0/1/0 : Port Control Disabled
```

  Failure case:

```
%L2-DOT1X-5-PORT_CONTROL_DISABLE_FAILURE : Hu0/0/1/0 : Failed to disable port-control
```

- As part of MAB client authentication process:

  Success case:

```
%L2-DOT1X-5-MAB_AUTH_SUCCESS : Hu0/0/1/0 : Authentication successful for client
<mac-address>
%L2-DOT1X-5-PORT_CONTROL_ADD_CLIENT_SUCCESS : Hu0/0/1/0 : Port Access Enabled For Client
 <mac-address>
```

  Failure case:

```
%L2-DOT1X-5-MAB_AUTH_FAIL    : Hu0/0/1/0 : Authentication failed for client <mac-address>
%L2-DOT1X-5-PORT_CONTROL_REMOVE_CLIENT_SUCCESS : Hu0/0/1/0 : Port Access Disabled For
Client <mac-address>
```

- When the authentication server is unreachable:

```
%L2-DOT1X-5-MAB_AAA_UNREACHABLE : Hu0/0/1/0 : AAA server unreachable for client
027E.15F2.CAE7, Retrying Authentication
```