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 NCS 5500 Series Routers and Cisco NCS 540 and NCS 560 Series Routers*.

- MAC Authentication Bypass, on page 2

MAC Authentication Bypass

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
MAC Authentication Bypass	Release 24.3.1	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 feature introduces these changes: **CLI:** • New **mab** option for the **dot1x profile** command • New **mab-retry-time** option for the **authenticator** command • **clear mab** • **show mab**

MAC authentication bypass (MAB) is a port control feature in which the router (authenticator) uses the MAC address of the end device or the client (also called as supplicant) as an authenticating parameter to provide network access.

802.1X (Dot1x) is one of the most widely used port-based authentication method to allow controlled access to the end devices connected to the port. However, not all clients support 802.1X. We would still need to allow them into the network even without 802.1X authentication. The MAB feature intends to provide this controlled access to such devices based on their MAC addresses.

Restrictions for MAC Authentication Bypass

These restrictions apply to the MAB feature:

- With MAB, you can perform user authentication using a remote AAA server only; 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.

- MAB supports only a single end device on each port.

  Hence, you must configure the authenticator (the router) to be in **single-host** mode.

Authentication Failure Scenarios of MAB

This table lists the various authentication failure scenarios and the expected behavior of MAB feature.

Table 2: Authentication Failure Scenarios and Expected Feature Behavior of MAB

If...	And...	Then…
the RADIUS server rejects the authentication request	—	the router • deletes the client programming on the port, if that client was already authenticated • retries the authentication process twice with the RADIUS server at an interval (configurable using the **authenticator timer mab-retry-time** command) of 60 seconds, by default • clears the client session and its programming on the port (if the server still does not authorize the client), and • puts the port back in MAC learning mode to relearn a new MAC address.
the client is unauthenticated	authentication does not happen after the retries	the router • deletes the client context, and • puts the port back in MAC learning mode to relearn a new MAC address.
the RADIUS server is not reachable during authentication process	**server dead action auth-retry** command is configured	the router • retains the programming of the client that was already authenticated • retries the authentication process with the RADIUS server at an interval (configurable using the **authenticator timer mab-retry-time** command) of 60 seconds until the server becomes available • does not attempt to learn any new MAC address on the port, and • the router puts the port back in MAC learning mode to relearn a new MAC address. To clear the client session and its programming on the router, use the **clear mab session** command.

If...	And...	Then...
the RADIUS server is not reachable during authentication process	**server dead action auth-retry** command is not configured	the router • deletes the programming of the client that was already authenticated and retries authentication • automatically clears the client session, if the client is still not authenticated, and • puts the port back in MAC learning mode to relearn a new MAC address.

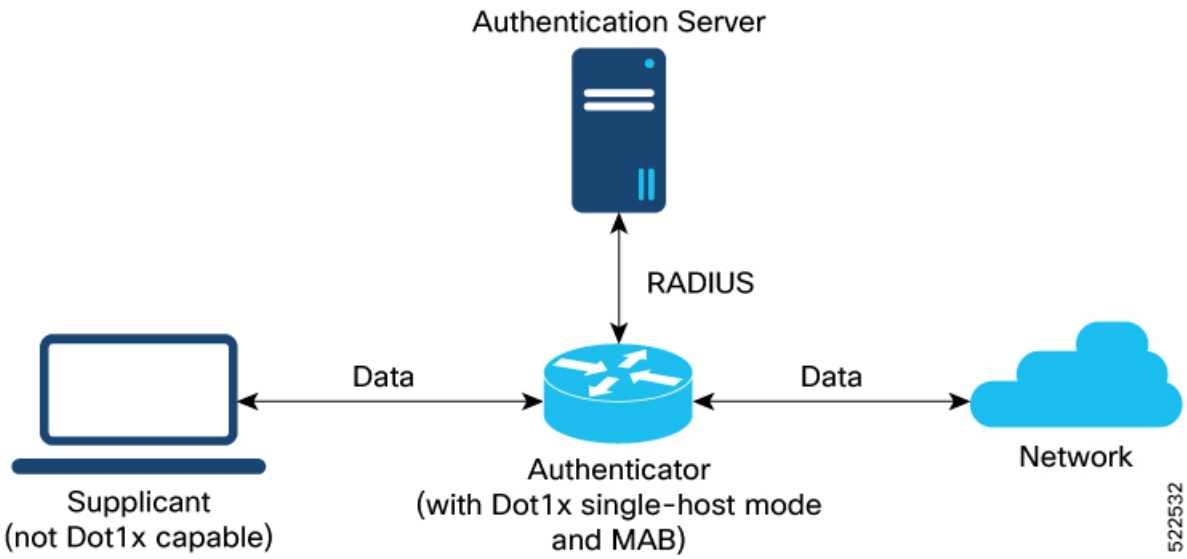
How MAC Authentication Bypass Works

Summary

These are the key components of MAC authentication bypass:

- Supplicant: The client or end device without dot1x support.
- Authenticator: The router that tries to authenticate the host device running the supplicant with the authentication server.
- Authentication Server: The server that provides the authenticator the RADIUS reply (**Access-Accept** or **Access-Reject** message), which allows or denies network access to the end device.

Workflow



These stages describe how MAC authentication bypass process works.

1. The router receives an incoming data packet from the client that is connected to the router port.
2. The router

a. learns the source MAC address, and

b. 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.

3. Based on the authentication result, the router allows or drops the data packets from that client.

If the RADIUS server…	Then…	And…
returns a success (**Access-Accept**) message	• it indicates that the MAC address is authenticated • the client is authorized to send traffic through that port	the router allows the traffic from the client to be forwarded to the network.
returns a failure (**Access-Reject**) message	it indicates that the MAC address is unauthenticated	the router drops further data packets from that client.

Result

Thus, the MAB feature brings in port control functionality for Cisco IOS XR routers and provides end devices a controlled access to network services.

Configure MAC Authentication Bypass

Before you begin

- 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) by specifying these parameters:
  - Re-authentication time—using **reauth-time** option
  - Host mode—using **single-host** option
  - Retry action for server-unreachable scenarios—using **auth-retry** or **auth-fail** option

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

See *Running Configuration* section for examples.

Follow these steps to configure MAC authentication bypass feature.

Procedure

Step 1 Enable MAB on the dot1x profile.

Example:

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

Step 2 Configure the authenticator retry time for MAB clients.

Example:

```
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
```

Step 3 Attach the dot1x profile to the corresponding interface or port on the router.

Example:

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

Step 4 Verify the running configuration on the router.

Example:

```
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 MAC Authentication Bypass

Follow these steps to verify MAC authentication bypass feature.

Procedure

Step 1 Check the MAB summary.

Example:

```
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#
```

The *Status* field shows as **Authorized**.

Step 2 Verify the detailed status of MAB.

Example:

```
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#
```

The *PortControl* field shows as **Enabled**.

Step 3 Verify the MAB interface summary.

Example:

```
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
```

The *Status* field shows as **Authorized**.

Step 4 Verify the MAB interface details.

Example:

```
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#
```

The *PortControl* field shows as **Enabled**.

Step 5 Verify the MAB interface statistics.

Example:

```
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#
```

The *EnableSuccess* field under *Port Control* shows as **1**.

System Logs for MAC Authentication Bypass

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

- When you apply dot1x profile 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 you remove dot1x profile 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
  ```