Debug Authentications

Contents

[Introduction](#)
[Prerequisites](#)
[Requirements](#)
[Components Used](#)
[Conventions](#)
[Capture Debugs](#)
[EAP](#)
[MAC Authentication](#)
[WPA](#)
[Administrative/HTTP Authentication](#)
[Related Information](#)

Introduction

Wireless communication uses authentication in many ways. The most common authentication type is Extensible Authentication Protocol (EAP) in different types and forms. Other authentication types include MAC address authentication and administrative authentication. This document describes how to debug and interpret the output from debug authentications. The information from these debugs is invaluable when you troubleshoot wireless installations.

Note: The portions of this document that refer to non-Cisco products are based on the experience of the author, not on formal training. They are intended for your convenience and not as technical support. For authoritative technical support on non-Cisco products, contact the technical support for that product.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Authentication as it relates to wireless networks
- Cisco IOS® software command-line interface (CLI)
- RADIUS server configuration

Components Used

The information in this document is based on these software and hardware versions:

- Cisco IOS software-based wireless products of any model and version
- Hilgraeve HyperTerminal

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Capture Debugs

If you cannot capture and analyze debug information, the information is useless. The easiest way to capture this data is with a screen-capture function that is built into the Telnet or communications application.

This example describes how to capture output with the Hilgraeve HyperTerminal application. Most Microsoft Windows operating systems include HyperTerminal, but you can apply the concepts to any terminal emulation application. For more complete information on the application, refer to Hilgraeve .

Complete these steps in order to configure HyperTerminal to communicate with your access point (AP) or bridge:

1. In order to open HyperTerminal, choose **Start > Programs > System Tools > Communications > HyperTerminal**.**Figure 1 – HyperTerminal Launch**
2. When HyperTerminal opens, complete these steps:Enter a name for the connection.Choose an icon.Click **OK**.
3. For Telnet connections, complete these steps:From the Connect Using drop-down menu, choose **TCP/IP**.Enter the IP address of the device where you want to run the debugs.Click **OK**.**Figure 2 – Telnet Connection**
4. For console connections, complete these steps:From the Connect Using drop-down menu, choose the COM port where the console cable is connected.Click **OK**.The property sheet for the connection appears.Set the speed for the connection to the console port.In order to restore the default port settings, click **Restore Defaults**.**Note:** Most Cisco products follow the default port settings.The default port settings are:Bits per second—9600Data bits—8Parity—NoneStop bits—1Flow control—None**Figure 3 – COM1 Properties** At this point, the Telnet or console connection establishes, and you are prompted for a user name and password.**Note:** Cisco Aironet equipment assigns both a default user name and password of *Cisco* (case sensitive).
5. In order to run debugs, complete these steps:Issue the **enable** command in order to enter privileged mode.Enter the enable password.**Note:** Remember that the default password for Aironet equipment is *Cisco* (case sensitive).**Note:** In order to see the output of debugs from a Telnet session, use the **terminal monitor** or **term mon** command in order to turn on the terminal monitor.**Figure 4 – Connected Telnet Session**
6. After you establish a connection, complete these steps in order to collect a screen capture:Choose **Capture Text** from the Transfer menu.**Figure 5 – Save a Screen Capture**When a dialog box opens that prompts you for a file name for the output, enter a file

name.

7. Complete these steps in order to disable the screen wrap:**Note:** You can read the debugs more easily when you disable the screen wrap.From the HyperTerminal menu, choose **File**.Choose **Properties**.On the connection property sheet, click the **Settings** tab.Click **ASCII Setup**.Uncheck **Wrap lines that exceed terminal width**.In order to close the ASCII Settings, click **OK**.In order to close the connection property sheet, click **OK**.**Figure 6 – ASCII Settings**

Now that you can capture any screen output to a text file, the debugs that you run depend on what is negotiated. The next sections of this document describe the type of negotiated connection provided by the debugs.

EAP

These debugs are the most helpful for EAP authentications:

- **debug radius authentication**—The outputs of this debug start with this word: `RADIUS`.
- **debug dot11 aaa authenticator process**—The outputs of this debug start with this text: `dot11_auth_dot1x_`.
- **debug dot11 aaa authenticator state-machine**—The outputs of this debug start with this text: `dot11_auth_dot1x_run_rfsm`.

These debugs show:

- What is reported during the RADIUS portions of an authentication dialog
- The actions that are taken during that authentication dialog
- The various states through which the authentication dialog transitions

This example shows a successful Light EAP (LEAP) authentication:

Successful EAP Authentication Example

```
Apr  8 17:45:48.208: dot11_auth_dot1x_start: in the
dot11_auth_dot1x_start
Apr  8 17:45:48.208:
dot11_auth_dot1x_send_id_req_to_client:
   sending identity request for 0002.8aa6.304f Apr 8
17:45:48.208: dot11_auth_dot1x_send_id_req_to_client:
Started timer client_timeout 30 seconds Apr 8
17:45:48.210: dot11_auth_parse_client_pak: Received
EAPOL packet from 0002.8aa6.304f Apr 8 17:45:48.210:
dot11_auth_dot1x_run_rfsm: Executing
Action(CLIENT_WAIT,EAP_START) for 0002.8aa6.304f
Apr  8 17:45:48.210:
dot11_auth_dot1x_send_id_req_to_client:
    sending identity request for 0002.8aa6.304f Apr 8
17:45:48.210: dot11_auth_dot1x_send_id_req_to_client:
Started timer client_timeout 30 seconds Apr 8
17:45:48.212: dot11_auth_parse_client_pak: Received
EAPOL packet from 0002.8aa6.304f Apr 8 17:45:48.212:
dot11_auth_parse_client_pak: id is not matching req-
id:1resp-id:2, waiting for response Apr 8 17:45:48.213:
dot11_auth_parse_client_pak: Received EAPOL packet from
0002.8aa6.304f Apr 8 17:45:48.213:
dot11_auth_dot1x_run_rfsm: Executing
Action(CLIENT_WAIT,CLIENT_REPLY) for 0002.8aa6.304f
Apr  8 17:45:48.214:
```

```
dot11_auth_dot1x_send_response_to_server:
    Sending client 0002.8aa6.304f data to server Apr 8
17:45:48.214: dot11_auth_dot1x_send_response_to_server:
tarted timer server_timeout 60 seconds Apr 8
17:45:48.214: RADIUS: AAA Unsupported [248] 14 Apr 8
17:45:48.214: RADIUS: 6C 61 62 61 70 31 32 30 30 69 70
31 [labap1200ip1] Apr 8 17:45:48.215: RADIUS: AAA
Unsupported [150] 2 Apr 8 17:45:48.215:
RADIUS(0000001C): Storing nasport 17 in rad_db Apr 8
17:45:48.215: RADIUS(0000001C): Config NAS IP:
10.0.0.102 Apr 8 17:45:48.215: RADIUS/ENCODE(0000001C):
acct_session_id: 28 Apr 8 17:45:48.216:
RADIUS(0000001C): Config NAS IP: 10.0.0.102 Apr 8
17:45:48.216: RADIUS(0000001C): sending Apr 8
17:45:48.216: RADIUS(0000001C): Send Access-Request to
10.0.0.3:1645 id 21645/93, len 139 Apr 8 17:45:48.216:
RADIUS: authenticator 92 26 A8 31 ED 60 6A 88 - 84 8C 80
B2 B8 26 4C 04 Apr 8 17:45:48.216: RADIUS: User-Name [1]
9 "aironet" Apr 8 17:45:48.216: RADIUS: Framed-MTU [12]
6 1400 Apr 8 17:45:48.217: RADIUS: Called-Station-Id
[30] 16 "0005.9a39.0374" Apr 8 17:45:48.217: RADIUS:
Calling-Station-Id [31] 16 "0002.8aa6.304f" Apr 8
17:45:48.217: RADIUS: Service-Type [6] 6 Login [1] Apr 8
17:45:48.217: RADIUS: Message-Authenticato[80] 18 * Apr
8 17:45:48.217: RADIUS: EAP-Message [79] 14 Apr 8
17:45:48.218: RADIUS: 02 02 00 0C 01 61 69 72 6F 6E 65
74 [?????aironet] Apr 8 17:45:48.218: RADIUS: NAS-Port-
Type [61] 6 802.11 wireless [19] Apr 8 17:45:48.218:
RADIUS: NAS-Port [5] 6 17 Apr 8 17:45:48.218: RADIUS:
NAS-IP-Address [4] 6 10.0.0.102 Apr 8 17:45:48.218:
RADIUS: Nas-Identifier [32] 16 "labap1200ip102" Apr 8
17:45:48.224: RADIUS: Received from id 21645/93
10.0.0.3:1645, Access-Challenge, len 69 Apr 8
17:45:48.224: RADIUS: authenticator C8 6D 9B B3 67 60 44
29 - CC AB 39 DE 00 A9 A8 CA Apr 8 17:45:48.224: RADIUS:
EAP-Message [79] 25 Apr 8 17:45:48.224: RADIUS: 01 43 00
17 11 01 00 08 63 BB E7 8C 0F AC EB 9A
[?C??????c???????] Apr 8 17:45:48.225: RADIUS: 61 69 72
6F 6E 65 74 [aironet] Apr 8 17:45:48.225: RADIUS:
Session-Timeout [27] 6 20 Apr 8 17:45:48.225: RADIUS:
Message-Authenticato[80] 18 * Apr 8 17:45:48.226:
RADIUS(0000001C): Received from id 21645/93 Apr 8
17:45:48.226: RADIUS/DECODE: EAP-Message fragments, 23,
total 23 bytes Apr 8 17:45:48.226:
dot11_auth_dot1x_parse_aaa_resp: Received server
response: GET_CHALLENGE_RESPONSE Apr 8 17:45:48.226:
dot11_auth_dot1x_parse_aaa_resp: found eap pak in server
response Apr 8 17:45:48.226:
dot11_auth_dot1x_parse_aaa_resp: found session timeout
20 sec Apr 8 17:45:48.227: dot11_auth_dot1x_run_rfsm:
Executing Action(SERVER_WAIT,SERVER_REPLY) for
    0002.8aa6.304f
Apr  8 17:45:48.227:
dot11_auth_dot1x_send_response_to_client:
    Forwarding server message to client 0002.8aa6.304f
Apr 8 17:45:48.227:
dot11_auth_dot1x_send_response_to_client: Started timer
client_timeout 20 seconds Apr 8 17:45:48.232:
dot11_auth_parse_client_pak: Received EAPOL packet from
0002.8aa6.304f Apr 8 17:45:48.232:
dot11_auth_dot1x_run_rfsm: Executing Action
(CLIENT_WAIT,CLIENT_REPLY) for 0002.8aa6.304f
Apr  8 17:45:48.232:
dot11_auth_dot1x_send_response_to_server:
```

Sending client 0002.8aa6.304f data to server Apr 8 17:45:48.232: dot11_auth_dot1x_send_response_to_server: Started timer server_timeout 60 seconds Apr 8 17:45:48.233: RADIUS: AAA Unsupported [248] 14 Apr 8 17:45:48.234: RADIUS: 6C 61 62 61 70 31 32 30 30 69 70 31 [labap1200ip1] Apr 8 17:45:48.234: RADIUS: AAA Unsupported [150] 2 Apr 8 17:45:48.234: RADIUS(0000001C): Using existing nas_port 17 Apr 8 17:45:48.234: RADIUS(0000001C): Config NAS IP: 10.0.0.102 Apr 8 17:45:48.234: RADIUS/ENCODE(0000001C): acct_session_id: 28 Apr 8 17:45:48.234: RADIUS(0000001C): Config NAS IP: 10.0.0.102 Apr 8 17:45:48.234: RADIUS(0000001C): sending Apr 8 17:45:48.234: RADIUS(0000001C): Send Access-Request to 10.0.0.3:1645 id 21645/94, len 166 Apr 8 17:45:48.235: RADIUS: authenticator 93 B5 CC B6 41 97 A0 85 - 1B 4D 13 0F 6A EE D4 11 Apr 8 17:45:48.235: RADIUS: User-Name [1] 9 "aironet" Apr 8 17:45:48.235: RADIUS: Framed-MTU [12] 6 1400 Apr 8 17:45:48.236: RADIUS: Called-Station-Id [30] 16 "0005.9a39.0374" Apr 8 17:45:48.236: RADIUS: Calling-Station-Id [31] 16 "0002.8aa6.304f" Apr 8 17:45:48.236: RADIUS: Service-Type [6] 6 Login [1] Apr 8 17:45:48.236: RADIUS: Message-Authenticato[80] 18 * Apr 8 17:45:48.236: RADIUS: EAP-Message [79] 41 Apr 8 17:45:48.236: RADIUS: 02 43 00 27 11 01 00 18 30 9F 55 AF 05 03 71 7D [?C?'????0?U???q}] Apr 8 17:45:48.236: RADIUS: 25 41 1B B0 F4 A9 7C EE F5 51 24 9A FC 6D 51 6D [?A????|??Q$??mQm] Apr 8 17:45:48.237: RADIUS: 61 69 72 6F 6E 65 74 [aironet] Apr 8 17:45:48.237: RADIUS: NAS-Port-Type [61] 6 802.11 wireless [19] Apr 8 17:45:48.237: RADIUS: NAS-Port [5] 6 17 Apr 8 17:45:48.238: RADIUS: NAS-IP-Address [4] 6 10.0.0.102 Apr 8 17:45:48.238: RADIUS: Nas-Identifier [32] 16 "labap1200ip102" Apr 8 17:45:48.242: RADIUS: Received from id 21645/94 10.0.0.3:1645, Access-Challenge, len 50 Apr 8 17:45:48.243: RADIUS: authenticator 59 2D EE 24 CF B2 87 AF - 86 D0 C9 00 79 BE 6E 1E Apr 8 17:45:48.243: RADIUS: EAP-Message [79] 6 Apr 8 17:45:48.243: RADIUS: 03 43 00 04 [?C??] Apr 8 17:45:48.244: RADIUS: Session-Timeout [27] 6 20 Apr 8 17:45:48.244: RADIUS: Message-Authenticato[80] 18 * Apr 8 17:45:48.244: RADIUS(0000001C): Received from id 21645/94 Apr 8 17:45:48.244: RADIUS/DECODE: EAP-Message fragments, 4, total 4 bytes Apr 8 17:45:48.244: dot11_auth_dot1x_parse_aaa_resp: *Received server response: GET_CHALLENGE_RESPONSE* Apr 8 17:45:48.245: dot11_auth_dot1x_parse_aaa_resp: found eap pak in server response Apr 8 17:45:48.245: dot11_auth_dot1x_parse_aaa_resp: found session timeout 20 sec Apr 8 17:45:48.245: dot11_auth_dot1x_run_rfsm: Executing Action(**SERVER_WAIT,SERVER_REPLY**) for 0002.8aa6.304f Apr 8 17:45:48.245: dot11_auth_dot1x_send_response_to_client: *Forwarding server message to client 0002.8aa6.304f* Apr 8 17:45:48.246: dot11_auth_dot1x_send_response_to_client: Started timer client_timeout 20 seconds Apr 8 17:45:48.249: dot11_auth_parse_client_pak: Received EAPOL packet from 0002.8aa6.304f Apr 8 17:45:48.250: dot11_auth_dot1x_run_rfsm: Executing Action(**CLIENT_WAIT,CLIENT_REPLY**) for 0002.8aa6.304f Apr 8 17:45:48.250: dot11_auth_dot1x_send_response_to_server:

Sending client 0002.8aa6.304f data to server Apr 8 17:45:48.250: dot11_auth_dot1x_send_response_to_server: Started timer server_timeout 60 seconds Apr 8 17:45:48.250: RADIUS: AAA Unsupported [248] 14 Apr 8 17:45:48.251: RADIUS: 6C 61 62 61 70 31 32 30 30 69 70 31 [labap1200ip1] Apr 8 17:45:48.251: RADIUS: AAA Unsupported [150] 2 Apr 8 17:45:48.251: RADIUS(0000001C): Using existing nas_port 17 Apr 8 17:45:48.252: RADIUS(0000001C): Config NAS IP: 10.0.0.102 Apr 8 17:45:48.252: RADIUS/ENCODE(0000001C): acct_session_id: 28 Apr 8 17:45:48.252: RADIUS(0000001C): Config NAS IP: 10.0.0.102 Apr 8 17:45:48.252: RADIUS(0000001C): sending Apr 8 17:45:48.252: RADIUS(0000001C): Send Access-Request to 10.0.0.3:1645 id 21645/95, len 150 Apr 8 17:45:48.252: RADIUS: authenticator 39 1C A5 EF 86 9E BA D1 - 50 FD 58 80 A8 8A BC 2A Apr 8 17:45:48.253: RADIUS: User-Name [1] 9 "aironet" Apr 8 17:45:48.253: RADIUS: Framed-MTU [12] 6 1400 Apr 8 17:45:48.253: RADIUS: Called-Station-Id [30] 16 "0005.9a39.0374" Apr 8 17:45:48.253: RADIUS: Calling-Station-Id [31] 16 "0002.8aa6.304f" Apr 8 17:45:48.254: RADIUS: Service-Type [6] 6 Login [1] Apr 8 17:45:48.254: RADIUS: Message-Authenticato[80] 18 * Apr 8 17:45:48.254: RADIUS: EAP-Message [79] 25 Apr 8 17:45:48.254: RADIUS: 01 43 00 17 11 01 00 08 50 9A 67 2E 7D 26 75 AA [?C??????P?g.}&u?] Apr 8 17:45:48.254: RADIUS: 61 69 72 6F 6E 65 74 [aironet] Apr 8 17:45:48.254: RADIUS: NAS-Port-Type [61] 6 802.11 wireless [19] Apr 8 17:45:48.254: RADIUS: NAS-Port [5] 6 17 Apr 8 17:45:48.255: RADIUS: NAS-IP-Address [4] 6 10.0.0.102 Apr 8 17:45:48.255: RADIUS: Nas-Identifier [32] 16 "labap1200ip102" Apr 8 17:45:48.260: RADIUS: Received from id 21645/95 10.0.0.3:1645, Access-Accept, len 206 Apr 8 17:45:48.260: RADIUS: authenticator 39 13 3C ED FC 02 68 63 - 24 13 1B 46 CF 93 B8 E3 Apr 8 17:45:48.260: RADIUS: Framed-IP-Address [8] 6 255.255.255.255 Apr 8 17:45:48.261: RADIUS: EAP-Message [79] 41 Apr 8 17:45:48.261: RADIUS: 02 00 00 27 11 01 00 18 FA 53 D0 29 6C 9D 66 8E [???'?????S?)l?f?] Apr 8 17:45:48.262: RADIUS: C4 A3 CD 54 08 8C 35 7C 74 0C 6A EF D4 6D 30 A4 [???T??5|t?j??m0?] Apr 8 17:45:48.262: RADIUS: 61 69 72 6F 6E 65 74 [aironet] Apr 8 17:45:48.262: RADIUS: Vendor, Cisco [26] 59 Apr 8 17:45:48.262: RADIUS: Cisco AVpair [1] 53 "leap:session-key=G:3asil;mwerAEJNYH-JxI," Apr 8 17:45:48.262: RADIUS: Vendor, Cisco [26] 31 Apr 8 17:45:48.262: RADIUS: Cisco AVpair [1] 25 "auth-algo-type=eap-leap" Apr 8 17:45:48.262: RADIUS: Class [25] 31 Apr 8 17:45:48.263: RADIUS: 43 49 53 43 4F 41 43 53 3A 30 30 30 30 31 64 36 [CISCOACS:00001d6] Apr 8 17:45:48.263: RADIUS: 33 2F 30 61 30 30 30 30 36 36 2F 31 37 [3/0a000066/17] Apr 8 17:45:48.263: RADIUS: Message-Authenticato[80] 18 * Apr 8 17:45:48.264: RADIUS(0000001C): Received from id 21645/95 Apr 8 17:45:48.264: RADIUS/DECODE: EAP-Message fragments, 39, total 39 bytes Apr 8 17:45:48.264: found leap session key Apr 8 17:45:48.265: dot11_auth_dot1x_parse_aaa_resp: *Received server response: PASS* Apr 8 17:45:48.265: dot11_auth_dot1x_parse_aaa_resp: *found eap pak in server response* Apr 8 17:45:48.265: dot11_auth_dot1x_parse_aaa_resp: *found leap session key in server response* Apr 8 17:45:48.265: dot11_auth_dot1x_parse_aaa_resp: leap session key length 16 Apr 8 17:45:48.266: dot11_auth_dot1x_run_rfsm:

```
Executing Action(SERVER_WAIT,SERVER_PASS) for
0002.8aa6.304f
Apr  8 17:45:48.266:
dot11_auth_dot1x_send_response_to_client:
    Forwarding server message to client 0002.8aa6.304f
Apr 8 17:45:48.266:
dot11_auth_dot1x_send_response_to_client: Started timer
client_timeout 20 seconds Apr 8 17:45:48.266: %DOT11-6-
ASSOC: Interface Dot11Radio0, Station RKIBBE-W2K4
0002.8aa6.304f Associated KEY_MGMT[NONE]
```

Notice the flow in the **state-machine** debugs. There is a progression through several states:

1. EAP_START
2. CLIENT_WAIT
3. CLIENT_REPLY
4. SERVER_WAIT
5. SERVER_REPLY**Note:** As the two negotiate, there can be several iterations of CLIENT_WAIT and CLIENT_REPLY, as well as SERVER_WAIT and SERVER_REPLY.
6. SERVER_PASS

The **process** debug shows each individual step through each state. The **radius** debugs show the actual conversation between the authentication server and the client. The easiest way to work with EAP debugs is to watch the progression of state machine messages through each state.

When something fails in the negotiation, the **state-machine** debugs show why the process stopped. Watch for messages similar to these examples:

- **CLIENT TIMEOUT** —This state indicates that the client did not respond within an appropriate amount of time. This failure to respond can occur due to one of these reasons:There is a problem with the client software.The EAP client timeout value (from the EAP Authentication subtab under Advanced Security) has expired.Some EAPs, particularly Protected EAP (PEAP), take longer than 30 seconds to complete authentication. Set this timer to a higher value (between 90 and 120 seconds).This is an example of a CLIENT TIMEOUT attempt:**Note:** Watch for any system error messages that are similar to this message:
  ```
  %DOT11-4-MAXRETRIES: Packet to client xxxx.xxxx.xxxx reached
    max retries, removing the client
  ```
  Note: Such error messages can indicate a radio frequency (RF) problem.
- **Shared secret mismatch between the AP and the RADIUS server**—In this example log, the RADIUS server does not accept the authentication request from the AP. The AP continues to send the request to the RADIUS server, but the RADIUS server rejects the request because the shared secret is mismatched.In order to resolve this problem, be sure to check that the shared secret on the AP is the same one that is used in the RADIUS server.
- **server_timeout** —This state indicates that the authentication server did not respond in an appropriate amount of time. This failure to respond occurs because of a problem on the server. Verify that these situations are true:The AP has IP connectivity to the authentication server.**Note:** You can use the **ping** command in order to verify connectivity.The authentication and accounting port numbers are correct for the server.**Note:** You can check the port numbers from the Server Manager tab.The authentication service is running and functional.This is an example of a server_timeout attempt:
- **SERVER_FAIL** —This state indicates that the server gave an unsuccessful authentication response based on the user credentials. The RADIUS debug that precedes this failure shows the user name that was presented to the authentication server. Be sure to check the Failed Attempts log in the authentication server for additional details on why the server denied the

client access.This is an example of a SERVER_FAIL attempt:

- **No Response from Client**—In this example, the radius server sends a pass message to the AP which the AP forwards on and then it associates the client. Eventually the client does not respond to the AP. Therefore, the AP deauthenticates it after it reaches the maximum retries.The AP forwards a get challenge response from the radius to the client. The client does not respond and reaches max retries which causes EAP to fail and the AP to deauthenticate the client.Radius sends a pass message to the AP, the AP forwards the pass message to the client, and the client does not respond. The AP deauthenticates it after it reaches the maximum retries. The client then attempts a new Identity request to the AP, but the AP rejects this request because the client has already reached the maximum retries.

The **process** and/or **radius** debugs that immediately *precede* the state machine message show the details of the failure.

For more information on how to configure EAP, refer to EAP Authentication with RADIUS Server.

MAC Authentication

These debugs are the most helpful for MAC authentication:

- **debug radius authentication**—When an external authentication server is used, the outputs of this debug start with this word: RADIUS.
- **debug dot11 aaa authenticator mac-authen**—The outputs of this debug start with this text: dot11_auth_dot1x_.

These debugs show:

- What is reported during the RADIUS portions of an authentication dialog
- The comparison between the MAC address that is given and the one that is authenticated against

When an external RADIUS server is used with MAC address authentication, the RADIUS debugs apply. The result of this conjunction is a display of the actual conversation between the authentication server and the client.

When a list of MAC addresses is built locally to the device as a user name and password database, only the **mac-authen** debugs show outputs. As the address match or mismatch is determined, these outputs display.

Note: Always enter any alphabetic characters in a MAC address in lowercase.

This examples shows a successful MAC authentication against a local database:

Successful MAC Authentication Example

```
Apr  8 19:02:00.109: dot11_auth_mac_start: method_list:
mac_methods
Apr  8 19:02:00.109: dot11_auth_mac_start: method_index:
0x4500000B, req: 0xA7626C
Apr  8 19:02:00.109: dot11_auth_mac_start: client-
>unique_id: 0x28
Apr  8 19:02:00.110: dot11_mac_process_reply: AAA reply
for 0002.8aa6.304f PASSED
Apr  8 19:02:00.145: %DOT11-6-ASSOC: Interface
```

```
Dot11Radio0, Station RKIBBE-W2K4
    0002.8aa6.304f Associated KEY_MGMT[NONE]
```

This examples shows a failed MAC authentication against a local database:

Failed MAC Authentication Example

```
Apr  8 19:01:22.336: dot11_auth_mac_start: method_list:
mac_methods
Apr  8 19:01:22.336: dot11_auth_mac_start: method_index:
0x4500000B,
    req: 0xA7626C
Apr  8 19:01:22.336: dot11_auth_mac_start: client-
>unique_id: 0x27
Apr  8 19:01:22.337: dot11_mac_process_reply:
    AAA reply for 0002.8aa6.304f FAILED
Apr  8 19:01:22.337: %DOT11-7-AUTH_FAILED:
    Station 0002.8aa6.304f Authentication failed
```

When a MAC address authentication fails, check for the accuracy of the characters that are entered in the MAC address. Be sure that you have entered any alphabetic characters in a MAC address in lowercase.

For more information on how to configure MAC authentication, refer to Configuring Authentication Types (Cisco IOS Software Configuration Guide for Cisco Aironet Access Points, 12.2(13)JA).

WPA

Although Wi-Fi Protected Access (WPA) is not an authentication type, it is a negotiated protocol.

- WPA negotiates between the AP and the client card.
- WPA key management negotiates after a client is successfully authenticated by an authentication server.
- WPA negotiates both a Pairwise Transient Key (PTK) and a Groupwise Transient Key (GTK) in a four-way handshake.

Note: Because WPA requires that the underlying EAP be successful, verify that clients can successfully authenticate with that EAP before you engage WPA.

These debugs are the most helpful for WPA negotiations:

- **debug dot11 aaa authenticator process**—The outputs of this debug start with this text: `dot11_auth_dot1x_`.
- **debug dot11 aaa authenticator state-machine**—The outputs of this debug start with this text: `dot11_auth_dot1x_run_rfsm`.

Relative to the other authentications in this document, WPA debugs are simple to read and analyze. A PTK message should be sent and an appropriate reply received. Next, a GTK message should be sent and another appropriate response received.

If the PTK or GTK messages are not sent, the configuration or software level on the AP can be at fault. If the PTK or GTK responses from the client are not received, check the configuration or software level on the WPA supplicant of the client card.

Successful WPA Negotiation Example

```
labap1200ip102#
Apr  7 16:29:57.908: dot11_dot1x_build_ptk_handshake:
    building PTK msg 1 for 0030.6527.f74a
Apr  7 16:29:59.190: dot11_dot1x_verify_ptk_handshake:
    verifying PTK msg 2 from 0030.6527.f74a
Apr  7 16:29:59.191: dot11_dot1x_verify_eapol_header:
Warning:
    Invalid key info (exp=0x381, act=0x109
Apr  7 16:29:59.191: dot11_dot1x_verify_eapol_header:
Warning:
    Invalid key len (exp=0x20, act=0x0)
Apr  7 16:29:59.192: dot11_dot1x_build_ptk_handshake:
    building PTK msg 3 for 0030.6527.f74a
Apr  7 16:29:59.783: dot11_dot1x_verify_ptk_handshake:
    verifying PTK msg 4 from 0030.6527.f74a
Apr  7 16:29:59.783: dot11_dot1x_verify_eapol_header:
Warning:
    Invalid key info (exp=0x381, act=0x109
Apr  7 16:29:59.783: dot11_dot1x_verify_eapol_header:
Warning:
    Invalid key len (exp=0x20, act=0x0)
Apr  7 16:29:59.788: dot11_dot1x_build_gtk_handshake:
    building GTK msg 1 for 0030.6527.f74a
Apr  7 16:29:59.788: dot11_dot1x_build_gtk_handshake:
    dot11_dot1x_get_multicast_key len 32 index 1
Apr  7 16:29:59.788: dot11_dot1x_hex_dump: GTK:
    27 CA 88 7D 03 D9 C4 61 FD 4B BE 71 EC F7 43 B5 82
93 57 83
Apr  7 16:30:01.633: dot11_dot1x_verify_gtk_handshake:
    verifying GTK msg 2 from 0030.6527.f74a
Apr  7 16:30:01.633: dot11_dot1x_verify_eapol_header:
    Warning: Invalid key info (exp=0x391, act=0x301
Apr  7 16:30:01.633: dot11_dot1x_verify_eapol_header:
Warning:
    Invalid key len (exp=0x20, act=0x0)
Apr  7 16:30:01.633: %DOT11-6-ASSOC: Interface
Dot11Radio0,
    Station   0030.6527.f74a Associated KEY_MGMT[WPA]
labap1200ip102#
```

For more information on how to configure WPA, refer to WPA Configuration Overview.

Administrative/HTTP Authentication

You can restrict administrative access to the device to users who are listed in either a local user name and password database or to an external authentication server. Administrative access is supported with both RADIUS and TACACS+.

These debugs are the most helpful for administrative authentication:

- **debug radius authentication** or **debug tacacs authentication**—The outputs of this debug start with one of these words: RADIUS or TACACS.
- **debug aaa authentication**—The outputs of this debugs start with this text: AAA/AUTHEN.
- **debug aaa authorization**—The outputs of this debugs start with this text: AAA/AUTHOR.

These debugs show:

- What is reported during the RADIUS or TACACS portions of an authentication dialog
- The actual negotiations for authentication and authorization between the device and the authentication server

This example shows a successful administrative authentication when the `Service-Type RADIUS` attribute is set to `Administrative`:

Successful Administrative Authentication Example with Service-Type Attribute

```
Apr 13 19:43:08.030: AAA: parse name=tty2 idb type=-1
tty=-1
Apr 13 19:43:08.030: AAA: name=tty2 flags=0x11 type=5
shelf=0 slot=0
    adapter=0 port=2 channel=0
Apr 13 19:43:08.031: AAA/MEMORY: create_user (0xA1BB6C)
user='NULL' ruser='NULL'
    ds0=0 port='tty2' rem_addr='10.0.0.25'
authen_type=ASCII service=LOGINN
Apr 13 19:43:08.031: AAA/AUTHEN/START (3200017540):
port='tty2'
    list='' action=LOGIN service=LOGIN
Apr 13 19:43:08.031: AAA/AUTHEN/START (3200017540):
using "default" list
Apr 13 19:43:08.031: AAA/AUTHEN/START (3200017540):
    Method=tac_admin (tacacs+) Apr 13 19:43:08.032:
TAC+: send AUTHEN/START packet ver=192 id=3200017540 Apr
13 19:43:08.032: AAA/AUTHEN(3200017540): Status=ERROR
Apr 13 19:43:08.032: AAA/AUTHEN/START (3200017540):
Method=rad_admin (radius) Apr 13 19:43:08.032:
AAA/AUTHEN(3200017540): Status=GETUSER Apr 13
19:43:08.032: AAA/AUTHEN/CONT (3200017540):
continue_login (user='(undef)') Apr 13 19:43:08.032:
AAA/AUTHEN(3200017540): Status=GETUSER Apr 13
19:43:08.032: AAA/AUTHEN(3200017540): Method=rad_admin
(radius) Apr 13 19:43:08.032: AAA/AUTHEN(3200017540):
Status=GETPASS Apr 13 19:43:08.033: AAA/AUTHEN/CONT
(3200017540): continue_login (user='aironet') Apr 13
19:43:08.033: AAA/AUTHEN(3200017540): Status=GETPASS Apr
13 19:43:08.033: AAA/AUTHEN(3200017540):
Method=rad_admin (radius) Apr 13 19:43:08.033: RADIUS:
Pick NAS IP for u=0xA1BB6C tableid=0 cfg_addr=10.0.0.102
best_addr=0.0.0.0 Apr 13 19:43:08.033: RADIUS: ustruct
sharecount=1 Apr 13 19:43:08.034: Radius:
radius_port_info() success=1 radius_nas_port=1 Apr 13
19:43:08.034: RADIUS(00000000): Send Access-Request to
10.0.0.3:1645 id 21646/48, len 76 Apr 13 19:43:08.034:
RADIUS: authenticator 91 A0 98 87 C1 FC F2 E7 - E7 E4 57
DF 20 D0 82 27 Apr 13 19:43:08.034: RADIUS: NAS-IP-
Address [4] 6 10.0.0.102 Apr 13 19:43:08.034: RADIUS:
NAS-Port [5] 6 2 Apr 13 19:43:08.035: RADIUS: NAS-Port-
Type [61] 6 Virtual [5] Apr 13 19:43:08.035: RADIUS:
User-Name [1] 9 "aironet" Apr 13 19:43:08.035: RADIUS:
Calling-Station-Id [31] 11 "10.0.0.25" Apr 13
19:43:08.035: RADIUS: User-Password [2] 18 * Apr 13
19:43:08.042: RADIUS: Received from id 21646/48
10.0.0.3:1645, Access-Accept, len 62 Apr 13
19:43:08.042: RADIUS: authenticator C9 32 E7 8F 97 5F E6
4C - 6B 90 71 EE ED 2C 2B 2B Apr 13 19:43:08.042:
RADIUS:  Service-Type        [6]   6
    Administrative             [6]
Apr 13 19:43:08.042: RADIUS:  Framed-IP-Address   [8]
```

```
6   255.255.255.255
Apr 13 19:43:08.042: RADIUS:  Class               [25]
30
Apr 13 19:43:08.043: RADIUS:   43 49 53 43 4F 41 43 53
3A 30 30 30 30 33 36 36
    [CISCOACS:0000366]
Apr 13 19:43:08.043: RADIUS:   39 2F 30 61 30 30 30 30
36 36 2F 32
    [9/0a000066/2]
Apr 13 19:43:08.044: RADIUS: saved authorization data
for user A1BB6C at B0C260
Apr 13 19:43:08.044: AAA/AUTHEN(3200017540): Status=PASS
Apr 13 19:43:08.044: tty2 AAA/AUTHOR/HTTP(1763745147):
Port='tty2' list='' service=EXEC Apr 13 19:43:08.044:
AAA/AUTHOR/HTTP: tty2(1763745147) user='aironet' Apr 13
19:43:08.044: tty2 AAA/AUTHOR/HTTP(1763745147): send AV
service=shell Apr 13 19:43:08.044: tty2
AAA/AUTHOR/HTTP(1763745147): send AV cmd* Apr 13
19:43:08.045: tty2 AAA/AUTHOR/HTTP(1763745147): found
list "default" Apr 13 19:43:08.045: tty2
AAA/AUTHOR/HTTP(1763745147): Method=tac_admin (tacacs+)
Apr 13 19:43:08.045: AAA/AUTHOR/TAC+: (1763745147):
user=aironet Apr 13 19:43:08.045: AAA/AUTHOR/TAC+:
(1763745147): send AV service=shell Apr 13 19:43:08.045:
AAA/AUTHOR/TAC+: (1763745147): send AV cmd* Apr 13
19:43:08.046: AAA/AUTHOR (1763745147): Post
authorization status = ERROR Apr 13 19:43:08.046: tty2
AAA/AUTHOR/HTTP(1763745147): Method=rad_admin (radius)
Apr 13 19:43:08.046: AAA/AUTHOR (1763745147): Post
authorization status = PASS_ADD Apr 13 19:43:08.443:
AAA/MEMORY: free_user (0xA1BB6C) user='aironet'
ruser='NULL' port='tty2' rem_addr='10.0.0.25'
authen_type=ASCII service=LOGIN
```

This example shows a successful administrative authentication when you use vendor-specific attributes in order to send a "priv-level" statement:

Successful Administrative Authentication Example with Vendor-Specific Attribute

```
Apr 13 19:38:04.699: RADIUS: cisco AVPair ""shell:priv-
lvl=15""
    not applied for shell
Apr 13 19:38:04.699: AAA/AUTHOR (380584213): Post
authorization status
    = PASS_ADD
Apr 13 19:38:04.802: AAA/MEMORY: free_user (0xAA0E38)
user='aironet'
    ruser='NULL' port='tty3' rem_addr='10.0.0.25'
authen_type=ASCII
    service=LOGIN
Apr 13 19:38:04.901: AAA: parse name=tty3 idb type=-1
tty=-1
Apr 13 19:38:04.901: AAA: name=tty3 flags=0x11 type=5
shelf=0 slot=0
    adapter=0  port=3 channel=0
Apr 13 19:38:04.902: AAA/MEMORY: create_user (0xAA23BC)
user='NULL'
    ruser='NULL' ds0=0 port='tty3' rem_addr='10.0.0.25'
    authen_type=ASCII service=LOGIN
Apr 13 19:38:04.902: AAA/AUTHEN/START (1346300140):
port='tty3' list=''
```

```
    action=LOGIN service=LOGIN
Apr 13 19:38:04.902: AAA/AUTHEN/START (1346300140):
using "default" list
Apr 13 19:38:04.902: AAA/AUTHEN/START (1346300140):
Method=tac_admin (tacacs+) Apr 13 19:38:04.902: TAC+:
send AUTHEN/START packet ver=192 id=1346300140 Apr 13
19:38:04.902: AAA/AUTHEN(1346300140): Status=ERROR Apr
13 19:38:04.902: AAA/AUTHEN/START (1346300140):
Method=rad_admin (radius) Apr 13 19:38:04.902:
AAA/AUTHEN(1346300140): Status=GETUSER Apr 13
19:38:04.903: AAA/AUTHEN/CONT (1346300140):
continue_login (user='(undef)') Apr 13 19:38:04.903:
AAA/AUTHEN(1346300140): Status=GETUSER Apr 13
19:38:04.903: AAA/AUTHEN(1346300140): Method=rad_admin
(radius) Apr 13 19:38:04.904: AAA/AUTHEN(1346300140):
Status=GETPASS Apr 13 19:38:04.904: AAA/AUTHEN/CONT
(1346300140): continue_login (user='aironet') Apr 13
19:38:04.904: AAA/AUTHEN(1346300140): Status=GETPASS Apr
13 19:38:04.904: AAA/AUTHEN(1346300140):
Method=rad_admin (radius) Apr 13 19:38:04.904: RADIUS:
Pick NAS IP for u=0xAA23BC tableid=0 cfg_addr=10.0.0.102
best_addr=0.0.0.0 Apr 13 19:38:04.904: RADIUS: ustruct
sharecount=1 Apr 13 19:38:04.904: Radius:
radius_port_info() success=1 radius_nas_port=1 Apr 13
19:38:04.925: RADIUS(00000000): Send Access-Request to
10.0.0.3:1645 id 21646/3, len 76 Apr 13 19:38:04.926:
RADIUS: authenticator 0C DD 2B B7 CA 5E 7C B9 - 46 90 FD
7A FD 56 3F 07 Apr 13 19:38:04.926: RADIUS: NAS-IP-
Address [4] 6 10.0.0.102 Apr 13 19:38:04.926: RADIUS:
NAS-Port [5] 6 3 Apr 13 19:38:04.926: RADIUS: NAS-Port-
Type [61] 6 Virtual [5] Apr 13 19:38:04.926: RADIUS:
User-Name [1] 9 "aironet" Apr 13 19:38:04.926: RADIUS:
Calling-Station-Id [31] 11 "10.0.0.25" Apr 13
19:38:04.926: RADIUS: User-Password [2] 18 * Apr 13
19:38:04.932: RADIUS: Received from id 21646/3
10.0.0.3:1645, Access-Accept, len 89 Apr 13
19:38:04.933: RADIUS: authenticator FA A4 31 49 51 87 9D
CA - 9D F7 B3 9B EF C2 8B 7E Apr 13 19:38:04.933:
RADIUS: Vendor, Cisco [26] 27 Apr 13 19:38:04.933:
RADIUS:   Cisco AVpair     [1]   21  ""shell:priv-
lvl=15""
Apr 13 19:38:04.934: RADIUS:  Service-Type        [6]
6   Login       [1]
Apr 13 19:38:04.934: RADIUS:  Framed-IP-Address   [8]
6   255.255.255.255
Apr 13 19:38:04.934: RADIUS:  Class               [25]
30
Apr 13 19:38:04.934: RADIUS:   43 49 53 43 4F 41 43 53
3A 30 30 30 30 33 36 33
    [CISCOACS:0000363]
Apr 13 19:38:04.934: RADIUS:   61 2F 30 61 30 30 30 30
36 36 2F 33
    [a/0a000066/3]
Apr 13 19:38:05.634: AAA/AUTHOR (3854191802): Post
authorization status = PASS_ADD Apr 13 19:38:05.917:
AAA/MEMORY: free_user (0xA9D054) user='aironet'
ruser='NULL' port='tty2' rem_addr='10.0.0.25'
authen_type=ASCII service=LOGIN priv=0
```

The most common problem with administrative authentication is the failure to configure the authentication server to send the appropriate privilege-level or administrative service-type attributes. This example attempt failed administrative authentication because no privilege-level attributes or administrative service-type attributes were sent:

Without Vendor-Specific or Service-Type Attributes

```
Apr 13 20:02:59.516: tty3 AAA/AUTHOR/HTTP(2007927065):
Port='tty3'
    list='' service=EXEC
Apr 13 20:02:59.516: AAA/AUTHOR/HTTP: tty3(2007927065)
user='aironet'
Apr 13 20:02:59.516: tty3 AAA/AUTHOR/HTTP(2007927065):
send AV service=shell
Apr 13 20:02:59.516: tty3 AAA/AUTHOR/HTTP(2007927065):
send AV cmd*
Apr 13 20:02:59.516: tty3 AAA/AUTHOR/HTTP(2007927065):
found list "default"
Apr 13 20:02:59.516: tty3 AAA/AUTHOR/HTTP(2007927065):
Method=tac_admin (tacacs+)
Apr 13 20:02:59.516: AAA/AUTHOR/TAC+: (2007927065):
user=aironet
Apr 13 20:02:59.516: AAA/AUTHOR/TAC+: (2007927065): send
AV service=shell
Apr 13 20:02:59.516: AAA/AUTHOR/TAC+: (2007927065): send
AV cmd*
Apr 13 20:02:59.516: AAA/AUTHOR (2007927065): Post
authorization status = ERROR
Apr 13 20:02:59.517: tty3 AAA/AUTHOR/HTTP(2007927065):
Method=rad_admin (radius)
Apr 13 20:02:59.517: AAA/AUTHOR (2007927065): Post
authorization status = PASS_ADD
Apr 13 20:02:59.561: AAA/MEMORY: free_user (0xA756E8)
user='aironet'
    ruser='NULL' port='tty2' rem_addr='10.0.0.25'
authen_type=ASCII
    service=LOGIN priv=0  vrf= (id=0)
Apr 13 20:02:59.620: AAA/MEMORY: free_user (0x9E5B04)
user='aironet'
    ruser='NULL'  port='tty3' rem_addr='10.0.0.25'
authen_type=ASCII
    service=LOGIN priv=0 vrf= (id=0)
Apr 13 20:03:04.501: AAA: parse name=tty2 idb type=-1
tty=-1
Apr 13 20:03:04.501: AAA: name=tty2 flags=0x11 type=5
shelf=0 slot=0 adapter=0
    port=2 channel=0
Apr 13 20:03:04.502: AAA/MEMORY: create_user (0xA9C7A4)
user='NULL'
    ruser='NULL' ds0=0 port='tty2' rem_addr='10.0.0.25'
authen_type=ASCII
    service=LOGIN priv=0
Apr 13 20:03:04.502: AAA/AUTHEN/START (377202642):
port='tty2' list=''
    action=LOGIN service=LOGIN
Apr 13 20:03:04.502: AAA/AUTHEN/START (377202642): using
"default" list
Apr 13 20:03:04.503: AAA/AUTHEN/START (377202642):
Method=tac_admin (tacacs+)
Apr 13 20:03:04.503: TAC+: send AUTHEN/START packet
ver=192 id=377202642
Apr 13 20:03:04.503: AAA/AUTHEN(377202642): Status=ERROR
Apr 13 20:03:04.503: AAA/AUTHEN/START (377202642):
Method=rad_admin (radius)
Apr 13 20:03:04.503: AAA/AUTHEN(377202642):
Status=GETUSER
Apr 13 20:03:04.503: AAA/AUTHEN/CONT (377202642):
continue_login (user='(undef)')
```

```
Apr 13 20:03:04.503: AAA/AUTHEN(377202642):
Status=GETUSER
Apr 13 20:03:04.503: AAA/AUTHEN(377202642):
Method=rad_admin (radius)
Apr 13 20:03:04.503: AAA/AUTHEN(377202642):
Status=GETPASS
Apr 13 20:03:04.504: AAA/AUTHEN/CONT (377202642):
continue_login (user='aironet')
Apr 13 20:03:04.504: AAA/AUTHEN(377202642):
Status=GETPASS
Apr 13 20:03:04.504: AAA/AUTHEN(377202642):
Method=rad_admin (radius)
Apr 13 20:03:04.504: RADIUS: Pick NAS IP for u=0xA9C7A4
tableid=0
    cfg_addr=10.0.0.102 best_addr=0.0.0.0
Apr 13 20:03:04.505: RADIUS: ustruct sharecount=1
Apr 13 20:03:04.505: Radius: radius_port_info()
success=1 radius_nas_port=1
Apr 13 20:03:04.505: RADIUS(00000000): Send Access-
Request to 10.0.0.3:1645
    id 21646/59, len 76
Apr 13 20:03:04.505: RADIUS:  authenticator 0F BD 81 17
8F C5 1C B4
    - 84 1C 66 4D CF D4 96 03
Apr 13 20:03:04.505: RADIUS:  NAS-IP-Address      [4]
6   10.0.0.102
Apr 13 20:03:04.506: RADIUS:  NAS-Port            [5]
6   2
Apr 13 20:03:04.506: RADIUS:  NAS-Port-Type       [61]
6   Virtual           [5]
Apr 13 20:03:04.506: RADIUS:  User-Name           [1]
9   "aironet"
Apr 13 20:03:04.506: RADIUS:  Calling-Station-Id  [31]
11  "10.0.0.25"
Apr 13 20:03:04.507: RADIUS:  User-Password       [2]
18  *
Apr 13 20:03:04.513: RADIUS: Received from id 21646/59
10.0.0.3:1645,
    Access-Accept, len 56
Apr 13 20:03:04.513: RADIUS:  authenticator BB F0 18 78
33 D0 DE D3
    - 8B E9 E0 EE 2A 33 92 B5
Apr 13 20:03:04.513: RADIUS:  Framed-IP-Address   [8]
6   255.255.255.255
Apr 13 20:03:04.513: RADIUS:  Class               [25]
30
Apr 13 20:03:04.514: RADIUS:   43 49 53 43 4F 41 43 53
3A 30 30 30 30 33 36 38
    [CISCOACS:0000368]
Apr 13 20:03:04.514: RADIUS:   33 2F 30 61 30 30 30 30
36 36 2F 32
    [3/0a000066/2]
Apr 13 20:03:04.515: RADIUS: saved authorization data
for user A9C7A4 at A9C99C
Apr 13 20:03:04.515: AAA/AUTHEN(377202642): Status=PASS
Apr 13 20:03:04.515: tty2 AAA/AUTHOR/HTTP(2202245138):
Port='tty2' list=''
    service=EXEC
Apr 13 20:03:04.515: AAA/AUTHOR/HTTP: tty2(2202245138)
user='aironet'
Apr 13 20:03:04.515: tty2 AAA/AUTHOR/HTTP(2202245138):
send AV service=shell
Apr 13 20:03:04.515: tty2 AAA/AUTHOR/HTTP(2202245138):
send AV cmd*
```

```
Apr 13 20:03:04.515: tty2 AAA/AUTHOR/HTTP(2202245138):
found list "default"
Apr 13 20:03:04.516: tty2 AAA/AUTHOR/HTTP(2202245138):
Method=tac_admin (tacacs+)
Apr 13 20:03:04.516: AAA/AUTHOR/TAC+: (2202245138):
user=aironet
Apr 13 20:03:04.516: AAA/AUTHOR/TAC+: (2202245138): send
AV service=shell
Apr 13 20:03:04.516: AAA/AUTHOR/TAC+: (2202245138): send
AV cmd*
Apr 13 20:03:04.517: AAA/AUTHOR (2202245138): Post
authorization status = ERROR
Apr 13 20:03:04.517: tty2 AAA/AUTHOR/HTTP(2202245138):
Method=rad_admin (radius)
Apr 13 20:03:04.517: AAA/AUTHOR (2202245138): Post
authorization status
    = PASS_ADD
Apr 13 20:03:04.619: AAA/MEMORY: free_user (0xA9C7A4)
user='aironet'
    ruser='NULL' port='tty2' rem_addr='10.0.0.25'
authen_type=ASCII
    service=LOGIN priv=0 vrf=
```

For more information on how to configure administrative authentication, refer to Administering the Access Point (Cisco IOS Software Configuration Guide for Cisco Aironet Access Points, 12.2(13)JA).

For more information on how to configure administrative privilege to users on the authentication server, refer to Sample Configuration: Local Authentication for HTTP Server Users. Check the section that matches the authentication protocol that you use.

Related Information

- **Cisco IOS Software Configuration Guide for Cisco Aironet Access Points, 12.2(13)JA**
- **EAP Authentication with RADIUS Server**
- **LEAP Authentication with Local RADIUS Server**
- **FAQ on Cisco Aironet Wireless Security**
- **Wireless Domain Services AP as an AAA Server Configuration Example**
- **Technical Support & Documentation - Cisco Systems**