

Supported RADIUS Attributes on the Wireless LAN Controller

Document ID: 96103

Introduction

Prerequisites

Requirements

Components Used

Conventions

Supported RADIUS Attributes on the Wireless LAN Controller

QoS-Level

ACL-Name

Interface-Name

VLAN-Tag

Tunnel Attributes

Syntax for the Configuration of WLC Attributes on RADIUS Servers

Cisco Airespace VSAs on Cisco Access Registrar

Cisco Airespace VSAs on Free Radius Sever

Cisco Airespace VSAs on the Microsoft IAS RADIUS Server

Cisco Airespace VSAs on Cisco Secure ACS Server

Verify and Troubleshoot

Related Information

Introduction

This document explains the list of supported RADIUS attributes on the Wireless LAN Controller (WLC) that are sent to the RADIUS server in the access-request, honored in access-accept, and sent in accounting requests. This also includes the vendor-specific attributes.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Wireless security methods
- RADIUS-based authentication

Components Used

This document is not restricted to specific software and hardware versions.

Conventions

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

Supported RADIUS Attributes on the Wireless LAN Controller

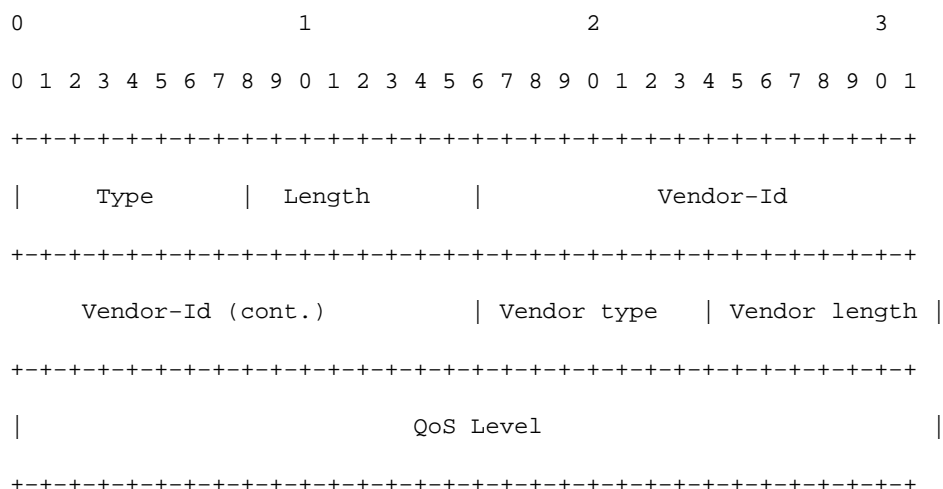
RADIUS attributes are used to define specific authentication, authorization, and accounting (AAA) elements in a user profile, which is stored on the RADIUS daemon. This section lists the RADIUS attributes currently supported on the Wireless LAN Controller.

- **Quality of Service** When present in a RADIUS Access Accept, the QoS-Level value overrides the QoS value specified in the WLAN profile.
- **ACL** When the Access Control List (ACL) attribute is present in the RADIUS Access Accept, the system applies the ACL-Name to the client station after it authenticates. This overrides any ACLs that are assigned to the interface.
- **VLAN** When a VLAN Interface-Name or VLAN-Tag is present in a RADIUS Access Accept, the system places the client on a specific interface.
- **WLAN ID** When the WLAN-ID attribute is present in the RADIUS Access Accept, the system applies the WLAN-ID (SSID) to the client station after it authenticates. The WLAN ID is sent by the WLC in all instances of authentication except IPsec. In case of web authentication, if the WLC receives a WLAN-ID attribute in the authentication response from the AAA server, and it does not match the ID of the WLAN, authentication is rejected. Other types of security methods do not do this.
- **DSCP Value** When present in a RADIUS Access Accept, the DSCP value overrides the DSCP value specified in the WLAN profile.
- **802.1p-Tag** When present in a RADIUS Access Accept, the 802.1p value overrides the default specified in the WLAN profile.

Note: The VLAN feature only supports MAC filtering, 802.1X, and Wi-Fi Protected Access (WPA). The VLAN feature does not support web authentication or IPsec. The operating system's local MAC filter database has been extended to include the interface name. This allows local MAC filters to specify which interface the client should be assigned. A separate RADIUS server can also be used, but the RADIUS server must be defined using the Security menus.

QoS-Level

The QoS-Level attribute indicates the Quality of Service level to be applied to the mobile client's traffic within the switching fabric, as well as over the air. This example shows a summary of the QoS-Level attribute format. The fields are transmitted from left to right.



"Type - 26 for Vendor-Specific

```

"Length - 10

"Vendor-Id - 14179

"Vendor type - 2

"Vendor length - 4

"Value - Three octets:

> - Bronze (Background)

~ - Silver (Best Effort)

™ - Gold (Video)

š - Platinum (Voice)

```

ACL-Name

The ACL-Name attribute indicates the ACL name to be applied to the client. A summary of the ACL-Name attribute format is shown here. The fields are transmitted from left to right.

```

0                1                2                3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|   Type   | Length |           Vendor-Id
+-----+-----+-----+-----+-----+-----+-----+-----+
| Vendor-Id (cont.) | Vendor type | Vendor length |
+-----+-----+-----+-----+-----+-----+-----+-----+
|           ACL Name...
+-----+-----+-----+-----+-----+-----+

```

```

"Type - 26 for Vendor-Specific

"Length - >7

"Vendor-Id - 14179

"Vendor type - 6

"Vendor length - >0

"Value - A string that includes the name of the ACL to use for the client

```

Interface-Name

The Interface-Name attribute indicates the VLAN interface a client is to be associated to. A summary of the Interface-Name attribute format is shown here. The fields are transmitted from left to right.

```

0                1                2                3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+

```

```

|      Type      |      Length      |      Vendor-Id
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
      Vendor-Id (cont.)      |      Vendor type  |      Vendor length  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      Interface Name...
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
"Type - 26 for Vendor-Specific
"Length - >7
"Vendor-Id - 14179
"Vendor type - 5
"Vendor length - >0

"Value - A string that includes the name of the interface
the client is to be assigned to.

```

Note: This attribute only works when MAC filtering is enabled or if 802.1X or WPA is used as the security policy.

VLAN-Tag

The VLAN-Tag attribute indicates the group ID for a particular tunneled session, and is also known as the Tunnel-Private-Group-ID attribute.

This attribute might be included in the Access-Request packet if the tunnel initiator can predetermine the group that results from a particular connection and should be included in the Access-Accept packet if this tunnel session is to be treated as if it belongs to a particular private group. Private groups can be used to associate a tunneled session with a particular group of users. For example, it can be used to facilitate routing of unregistered IP addresses through a particular interface. It should be included in Accounting-Request packets which contain Acct-Status-Type attributes with values of either Start or Stop and which pertain to a tunneled session.

A summary of the Tunnel-Private-Group-ID Attribute format is shown here. The fields are transmitted from left to right.

```

      0              1              2              3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      Type      |      Length      |      Tag      |      String...
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
"Type - 81 for Tunnel-Private-Group-ID.
"Length - > = 3

"Tag - The Tag field is one octet in length and is intended to
provide a means of grouping attributes in the same packet
which refer to the same tunnel. If the value of the Tag field is

```

greater than 0x00 and less than or equal to 0x1F, it should be interpreted as indicating which tunnel (of several alternatives) this attribute pertains. If the Tag field is greater than 0x1F, it should be interpreted as the first byte of the following String field.
"String - This field must be present. The group is represented by the String field. There is no restriction on the format of group IDs.

Tunnel Attributes

When any of the other RADIUS attributes (QoS-Level, ACL-Name, Interface-Name, or VLAN-Tag) are returned, the RFC 2868 tunnel attributes must also be returned.

RFC 2868 defines RADIUS tunnel attributes used for authentication and authorization, and RFC 2867 defines tunnel attributes used for accounting. Where the IEEE 802.1X Authenticator supports tunneling, a compulsory tunnel can be set up for the supplicant as a result of the authentication.

In particular, it might be desirable to allow a port to be placed into a particular VLAN, defined in IEEE 802.1Q, based on the result of the authentication. This can be used, for example, to allow a wireless host to remain on the same VLAN as it moves within a campus network.

The RADIUS server typically indicates the desired VLAN by including tunnel attributes within the Access-Accept. However, the IEEE 802.1X Authenticator might also provide a hint as to the VLAN to be assigned to the supplicant by including tunnel attributes within the Access-Request.

These tunnel attributes are used for the VLAN assignment:

- Tunnel-Type=VLAN (13)
- Tunnel-Medium-Type=802
- Tunnel-Private-Group-ID=VLANID

The VLANID is 12-bits, is a value between 1 and 4094, and is inclusive. Since the Tunnel-Private-Group-ID is of the type String as defined in RFC 2868, for use with IEEE 802.1X, the VLANID integer value is encoded as a string.

When tunnel attributes are sent, it is necessary to fill in the Tag field. This is noted in RFC 2868, section 3.1:

- The Tag field is one octet in length and is intended to provide a means of grouping attributes in the same packet that refer to the same tunnel. Valid values for this field are 0x01 through 0x1F (inclusive). If the Tag field is unused, it must be zero (0x00).
- For use with Tunnel-Client-Endpoint, Tunnel-Server-Endpoint, Tunnel-Private-Group-ID, Tunnel-Assignment-ID, Tunnel-Client-Auth-ID or Tunnel-Server-Auth-ID attributes (but not Tunnel-Type, Tunnel-Medium-Type, Tunnel-Password, or Tunnel-Preference), a tag field of greater than 0x1F is interpreted as the first octet of the next string field. For detailed information on the format refer to RFC 2868 section 3.1.
- Unless alternative tunnel types are provided, (for example, for IEEE 802.1X Authenticators that might support tunneling but not VLANs), it is only necessary for tunnel attributes to specify a single tunnel. As a result, where it is only desired to specify the VLANID, the Tag field should be set to zero (0x00) in all tunnel attributes. Where alternative tunnel types are provided, you should choose tag values between 0x01 and 0x1F.

Syntax for the Configuration of WLC Attributes on RADIUS Servers

Cisco Airespace VSAs on Cisco Access Registrar

Cisco CNS Access Registrar is a RADIUS-compliant, access policy server designed to support the delivery of dial, ISDN, and new services including DSL, cable with telco-return, wireless and Voice over IP. For detailed information on the Cisco Access Registrar refer to the Cisco Access Registrar Support Page.

This is the syntax that needs to be used on the Cisco Access Registrar to define the WLC attributes.

- **Defines Airespace RADIUS Attributes:**

```
Description = str:[0]
Name = str:[0]Airespace
Type = str:[0]SUB_ATTRIBUTES
VendorID = int32:[0]14179
VendorTypeSize = str:[0]8-bit
```

- **Defines WLAN-ID for the User:**

```
Description = str:[0]
Max = int32:[0]4294967295
Min = int32:[0]0
Name = str:[0]Airespace-WLAN-Id
SubAttribute = int32:[0]1
Type = str:[0]UINT32
```

- **Defines the QoS Level for a User:**

```
Description = str:[0]
Max = int32:[0]3
Min = int32:[0]0
Name = str:[0]Airespace-QoS-Level
SubAttribute = int32:[0]2
Type = str:[0]ENUM
0 = str:[0]Silver
1 = str:[0]Gold
2 = str:[0]Platinum
3 = str:[0]Bronze
```

- **Defines the DSCP Value of the Packets from a User:**

```
Description = str:[0]
Max = int32:[0]4294967295
Min = int32:[0]0
Name = str:[0]Airespace-DSCP
SubAttribute = int32:[0]3
Type = str:[0]UINT32
```

- **Defines the 802.1p Tag:**

```
Description = str:[0]
Max = int32:[0]4294967295
Min = int32:[0]0
Name = str:[0]Airespace-802.1P-Tag
SubAttribute = int32:[0]4
Type = str:[0]UINT32
```

- **Defines the Interface to Which the User is Mapped:**

```
Description = str:[0]
Max = int32:[0]253
Min = int32:[0]0
Name = str:[0]Airespace-Interface-Name
```

```
SubAttribute = int32:[0]5
Type = str:[0]STRING
```

- **Defines the ACL that is Applied:**

```
Description = str:[0]
Max = int32:[0]253
Min = int32:[0]0
Name = str:[0]Airespace-ACL-Name
SubAttribute = int32:[0]6
Type = str:[0]STRING
```

Cisco Airespace VSAs on Free Radius Sever

The Airespace dictionary file for the Free RADIUS server is available in the installation directory under the directory name **Share**. The filename is dictionary.airespace.

Note: The dictionary file might be different for earlier versions. The examples given in this document are from Free RADIUS version 1.1.6.

```
# -*- text -*-
#
#       As found on the net.
#
#       $Id: dictionary.airespace,v 1.3.2.1 2005/11/30 22:17:19 aland Exp $
#
VENDOR          Airespace          14179

BEGIN-VENDOR    Airespace
ATTRIBUTE       Airespace-Wlan-Id   1       integer
ATTRIBUTE       Airespace-QOS-Level 2       integer
ATTRIBUTE       Airespace-DSCP      3       integer
ATTRIBUTE       Airespace-8021p-Tag 4       integer
ATTRIBUTE       Airespace-Interface-Name 5       string
ATTRIBUTE       Airespace-ACL-Name   6       string

VALUE   Airespace-QOS-Level   Bronze   3
VALUE   Airespace-QOS-Level   Silver    0
VALUE   Airespace-QOS-Level   Gold       1
VALUE   Airespace-QOS-Level   Platinum  2

END-VENDOR Airespace
```

The vendor specific dictionary for Airespace products is included in the dictionary file available under the same directory. The file name is dictionary.

```
# -*- text -*-
#
# Version $Id: dictionary,v 1.93.2.5.2.10 2007/04/08 14:42:06 aland Exp $
#
#       DO NOT EDIT THE FILES IN THIS DIRECTORY
#
#
#       Use the main dictionary file (usually /etc/raddb/dictionary)
#       for local system attributes and $INCLUDEs.
#
#
#       This file contains dictionary translations for parsing
#       requests and generating responses. All transactions are
#       composed of Attribute/Value Pairs. The value of each attribute
#       is specified as one of 4 data types. Valid data types are:
#
```

```

#      text      - printable, generally UTF-8 encoded (subset of 'string')
#      string    - 0-253 octets
#      ipaddr    - 4 octets in network byte order
#      integer   - 32 bit value in big endian order (high byte first)
#      date      - 32 bit value in big endian order - seconds since
#                  00:00:00 GMT, Jan. 1, 1970
#      ifid      - 8 octets in network byte order
#      ipv6addr  - 16 octets in network byte order
#      ipv6prefix - 18 octets in network byte order
#
#      FreeRADIUS includes extended data types which are not defined
#      in the RFC's.  These data types are:
#
#      abinary - Ascend's binary filter format.
#      octets  - raw octets, printed and input as hex strings.
#                  e.g.: 0x123456789abcdef
#
#
#      Enumerated values are stored in the user file with dictionary
#      VALUE translations for easy administration.
#
#      Example:
#
#      ATTRIBUTE          VALUE
#      -----          -
#      Framed-Protocol = PPP
#      7                = 1      (integer encoding)
#
#
#      Include compatibility dictionary for older users file.  Move
#      this directive to the end of this file if you want to see the
#      old names in the logfiles, INSTEAD OF the new names.
#
$INCLUDE dictionary.compat

#
#      Include the RFC dictionaries next.
#
#      For a complete list of the standard attributes and values,
#      see:
#
#          http://www.iana.org/assignments/radius-types
#
$INCLUDE dictionary.rfc2865
$INCLUDE dictionary.rfc2866
$INCLUDE dictionary.rfc2867
$INCLUDE dictionary.rfc2868
$INCLUDE dictionary.rfc2869
$INCLUDE dictionary.rfc3162
$INCLUDE dictionary.rfc3576
$INCLUDE dictionary.rfc3580
$INCLUDE dictionary.rfc4372
$INCLUDE dictionary.rfc4675
$INCLUDE dictionary.rfc4679

#
#      Include vendor dictionaries after the standard ones.
#
$INCLUDE dictionary.3com
$INCLUDE dictionary.3gpp
$INCLUDE dictionary.3gpp2
$INCLUDE dictionary.acc
$INCLUDE dictionary.airespace
$INCLUDE dictionary.alcatel
$INCLUDE dictionary.alteon
$INCLUDE dictionary.alvarion

```

```

$INCLUDE dictionary.aruba
$INCLUDE dictionary.ascend
$INCLUDE dictionary.asn
$INCLUDE dictionary.bay
$INCLUDE dictionary.bintec
$INCLUDE dictionary.cablelabs
$INCLUDE dictionary.cabletron
$INCLUDE dictionary.cisco
#
#       The Cisco VPN300 dictionary is the same as the altiga one.
#       You shouldn't use both at the same time.
#
#$INCLUDE dictionary.cisco.vpn3000
$INCLUDE dictionary.cisco.vpn5000
$INCLUDE dictionary.cisco.bbsm

#
#       And finally the server internal attributes.
#
$INCLUDE dictionary.freeradius.internal

#
#       Miscellaneous attributes defined in weird places that
#       don't really belong anywhere else...
#
ATTRIBUTE          Originating-Line-Info          94          string

# As defined in draft-sterman-aaa-sip-00.txt
ATTRIBUTE          Digest-Response              206          string
ATTRIBUTE          Digest-Attributes            207          octets      #

#
#       Integer Translations
#
VALUE  Service-Type          Voice          12
VALUE  Service-Type          Fax            13
VALUE  Service-Type          Modem-Relay   14
VALUE  Service-Type          IAPP-Register 15
VALUE  Service-Type          IAPP-AP-Check 16

VALUE  Framed-Protocol       GPRS-PDP-Context 7

VALUE  NAS-Port-Type         Wireless-CDMA2000 22
VALUE  NAS-Port-Type         Wireless-UMTS     23
VALUE  NAS-Port-Type         Wireless-1X-EV   24
VALUE  NAS-Port-Type         IAPP              25

VALUE  Framed-Protocol       PPTP             9

```

Cisco Airespace VSAs on the Microsoft IAS RADIUS Server

For information on how to configure a Microsoft Internet Authentication Service (MS IAS) server to support Cisco Airespace Vendor Specific Attributes (VSAs) read [Cisco Airespace VSAs on MS IAS Radius Server Configuration Example](#)

Cisco Airespace VSAs on Cisco Secure ACS Server

The Cisco Secure Access Control Server Release 4.0 Solution Engine, supports many Remote Access Dial-In User Service (RADIUS) attributes that include Cisco Airespace Attributes.

ACS cannot offer partial support of IETF. Hence, when you add a Cisco Airespace device (into the Network Configuration), it automatically enables all IETF attributes. This table gives the Cisco Airespace attributes supported by Cisco ACS.

Number	Name	Description	Type of Value	Inbound/Outbound	Multiple
1	Aire-WLAN-Id	Name of the user being authenticated.	Integer	Outbound	No
2	Aire-QoS-Level	Enumerations: 3: Bronze 0: Silver 1: Gold 2: Platinum	Integer	Outbound	No
3	Aire-DSCP	—	Integer	Outbound	No
4	Aire-802.1P-Tag	—	Integer	Outbound	No
5	Aire-Interface-Name	—	String	Outbound	No
6	Aire-ACL-Name	—	String	Outbound	No

Cisco Airespace devices support some IETF attributes for 802.1x identity networking:

- Tunnel-Type (64)
- Tunnel-Medium-Type (65)
- Tunnel-Private-Group-Id (81)

In order to configure a specific attribute to be sent for a user, you must ensure that:

- In the Network Configuration section, you must configure the AAA client entry that correspond to the access device that grants network access to the user to use a variety of RADIUS that supports the attribute that you want sent to the AAA client.
- In the Interface Configuration section, you must enable the attribute so that it appears on user or user group profile pages. You can enable attributes on the page that correspond to the RADIUS variety that supports the attribute. For example, IETF RADIUS Session-Timeout attribute (27) appears on the RADIUS (IETF) page.

Note: By default, per-user RADIUS attributes are not enabled (they do not appear in the Interface Configuration page). Before you can enable attributes on a per-user basis, you must enable the Per-user TACACS+/RADIUS Attributes option on the Advanced Options page in the Interface Configuration section. After enabling per-user attributes, a user column appears as disabled in the Interface Configuration page for that attribute.

- In the profile that you use to control authorizations for the user in the user or group edit pages or Shared RADIUS Authorization Component page you must enable the attribute. When you enable this attribute, it causes ACS to send the attribute to the AAA client in the access-accept message. In the options that are associated with the attribute, you can determine the value of the attribute that is sent to the AAA client.

Refer to the RADIUS Attributes section of User Guide for Cisco Secure ACS Solution Engine 4.0 for more information.

Verify and Troubleshoot

When the user connects to the WLAN with a user ID and password, the WLC passes the credentials to the RADIUS server which authenticates the user against the conditions and the user profile configured. If the user authentication is successful, the RADIUS server returns a RADIUS accept request which also contains the RADIUS attributes configured for that user. In this example, the QoS policy of the user is returned.

You can issue the **debug aaa all enable** command in order to see the sequence of events that occur during authentication. Here is sample output:

```
(Cisco Controller) >debug aaa all enable
Wed Apr 18 18:14:24 2007: User admin authenticated
Wed Apr 18 18:14:24 2007: 28:1f:00:00:00:00 Returning AAA Error 'Success' (0) for
mobile 28:1f:00:00:00:00
Wed Apr 18 18:14:24 2007: AuthorizationResponse: 0xbadff97c
Wed Apr 18 18:14:24 2007:      structureSize.....70
Wed Apr 18 18:14:24 2007:      resultCode.....0
Wed Apr 18 18:14:24 2007:      protocolUsed.....0x00000008
Wed Apr 18 18:14:24 2007:      proxyState.....
28:1F:00:00:00:00-00:00
Wed Apr 18 18:14:24 2007:      Packet contains 2 AVPs:
Wed Apr 18 18:14:24 2007:      AVP[01] Service-Type.....
0x00000006 (6) (4 bytes)
Wed Apr 18 18:14:24 2007:      AVP[02] Airespace / WLAN-Identifier....
0x00000000 (0) (4 bytes)
Wed Apr 18 18:14:24 2007: User admin authenticated
Wed Apr 18 18:14:24 2007: 29:1f:00:00:00:00 Returning AAA Error 'Success'
(0) for mobile 29:1f:00:00:00:00
Wed Apr 18 18:14:24 2007: AuthorizationResponse: 0xbadff97c
Wed Apr 18 18:14:24 2007:      structureSize.....70
Wed Apr 18 18:14:24 2007:      resultCode.....0
Wed Apr 18 18:14:24 2007:      protocolUsed.....0x00000008
Wed Apr 18 18:14:24 2007:      proxyState.....
29:1F:00:00:00:00-00:00
Wed Apr 18 18:14:24 2007:      Packet contains 2 AVPs:
Wed Apr 18 18:14:24 2007:      AVP[01] Service-Type.....
0x00000006 (6) (4 bytes)
Wed Apr 18 18:14:24 2007:      AVP[02] Airespace / WLAN-Identifier....
0x00000000 (0) (4 bytes)
Wed Apr 18 18:15:08 2007: Unable to find requested user entry for User-VLAN10
Wed Apr 18 18:15:08 2007: AuthenticationRequest: 0xa64c8bc
Wed Apr 18 18:15:08 2007:      Callback.....0x8250c40
Wed Apr 18 18:15:08 2007:      protocolType.....0x00000001
Wed Apr 18 18:15:08 2007:      proxyState.....
00:40:96:AC:E6:57-00:00
Wed Apr 18 18:15:08 2007:      Packet contains 8 AVPs (not shown)
Wed Apr 18 18:15:08 2007: 00:40:96:ac:e6:57 Successful transmission
of Authentication Packet
(id 26) to 172.16.1.1:1812, proxy state
00:40:96:ac:e6:57-96:ac
Wed Apr 18 18:15:08 2007: 00000000: 01 1a 00 68 00 00 00 00 00 00 00 00 00 00 00 00
...h.....
Wed Apr 18 18:15:08 2007: 00000010: 00 00 00 00 01 0d 55 73 65 72 2d 56 4c 41 4e 31
.....User-VLAN1
Wed Apr 18 18:15:08 2007: 00000020: 30 02 12 fa 32 57 ba 2a ba 57 38 11 bc 9a 5d 59
0...2W.*.W8...Y
Wed Apr 18 18:15:08 2007: 00000030: ed ca 23 06 06 00 00 00 01 04 06 ac 10 01 1e 20
..#.....
Wed Apr 18 18:15:08 2007: 00000040: 06 57 4c 43 32 1a 0c 00 00 37 63 01 06 00 00 00
.WLC2...7c....
Wed Apr 18 18:15:08 2007: 00000050: 01 1f 0a 32 30 2e 30 2e 30 2e 31 1e 0d 31 37 32
...20.0.0.1..172
```

```

Wed Apr 18 18:15:08 2007: 00000060: 2e 31 36 2e 31 2e 33 30 .16.1.30
Wed Apr 18 18:15:08 2007: 00000000: 02 1a 00 46 3f cf 1b cc e4 ea 41 3e 28 7e cc bc
    ...F?...A>(~...
Wed Apr 18 18:15:08 2007: 00000010: 00 e1 61 ae 1a 0c 00 00 37 63 02 06 00 00 00 03
    ..a.....7c.....
Wed Apr 18 18:15:08 2007: 00000020: 06 06 00 00 00 01 19 20 37 d0 03 e6 00 00 01 37
    .....7.....7
Wed Apr 18 18:15:08 2007: 00000030: 00 01 ac 10 01 01 01 c7 7a 8b 35 20 31 80 00 00
    .....z.5.1...
Wed Apr 18 18:15:08 2007: 00000040: 00 00 00 00 00 1b .....
Wed Apr 18 18:15:08 2007: ****Enter processIncomingMessages: response code=2
Wed Apr 18 18:15:08 2007: ****Enter processRadiusResponse: response code=2
Wed Apr 18 18:15:08 2007: 00:40:96:ac:e6:57 Access-Accept received
    from RADIUS server
    172.16.1.1 for mobile 00:40:96:ac:e6:57 receiveId = 0
Wed Apr 18 18:15:08 2007: AuthorizationResponse: 0x9802520
Wed Apr 18 18:15:08 2007: structureSize.....114
Wed Apr 18 18:15:08 2007: resultCode.....0
Wed Apr 18 18:15:08 2007: protocolUsed.....0x00000001
Wed Apr 18 18:15:08 2007: proxyState.....
    00:40:96:AC:E6:57-00:00
Wed Apr 18 18:15:08 2007: Packet contains 3 AVPs:
Wed Apr 18 18:15:08 2007: AVP[01] Airespace / QOS-Level.....
    0x00000003 (3) (4 bytes)
Wed Apr 18 18:15:08 2007: AVP[02] Service-Type.....
    0x00000001 (1) (4 bytes)
Wed Apr 18 18:15:08 2007: AVP[03] Class.....
    DATA (30 bytes)
Wed Apr 18 18:15:08 2007: 00:40:96:ac:e6:57 Applying new AAA override for station
00:40:96:ac:e6:57
Wed Apr 18 18:15:08 2007: 00:40:96:ac:e6:57 Override values for
station 00:40:96:ac:e6:57
    source: 48, valid bits: 0x3
    qosLevel: 3, dscp: 0xffffffff, dot1pTag: 0xffffffff, sessionTimeout: -1
    dataAvgC: -1, rTAvgC: -1, dataBurstC: -1, rTimeBurstC: -1
    vlanIfName: '', aclName: '
Wed Apr 18 18:15:12 2007: AccountingMessage Accounting Start: 0xa64c8bc
Wed Apr 18 18:15:12 2007: Packet contains 13 AVPs:
Wed Apr 18 18:15:12 2007: AVP[01] User-Name.....
    User-VLAN10 (11 bytes)
Wed Apr 18 18:15:12 2007: AVP[02] Nas-Port.....
    0x00000001 (1) (4 bytes)
Wed Apr 18 18:15:12 2007: AVP[03] Nas-Ip-Address.....
    0xac10011e (-1408237282) (4 bytes)
Wed Apr 18 18:15:12 2007: AVP[04] NAS-Identifier.....
    0x574c4332 (1464615730) (4 bytes)
Wed Apr 18 18:15:12 2007: AVP[05] Airespace / WLAN-Identifier.....
    0x00000001 (1) (4 bytes)
Wed Apr 18 18:15:12 2007: AVP[06] Acct-Session-Id.....
    4626602c/00:40:96:ac:e6:57/16 (29 bytes)
Wed Apr 18 18:15:12 2007: AVP[07] Acct-Authentic.....
    0x00000001 (1) (4 bytes)
Wed Apr 18 18:15:12 2007: AVP[08] Tunnel-Type.....
    0x0000000d (13) (4 bytes)
Wed Apr 18 18:15:12 2007: AVP[09] Tunnel-Medium-Type.....
    0x00000006 (6) (4 bytes)
Wed Apr 18 18:15:12 2007: AVP[10] Tunnel-Group-Id.....
    0x3230 (12848) (2 bytes)
Wed Apr 18 18:15:12 2007: AVP[11] Acct-Status-Type.....
    0x00000001 (1) (4 bytes)
Wed Apr 18 18:15:12 2007: AVP[12] Calling-Station-Id.....
    20.0.0.1 (8 bytes)
Wed Apr 18 18:15:12 2007: AVP[13] Called-Station-Id.....
    172.16.1.30 (11 bytes)

```

This user shows that the user is authenticated. Then, AAA override values are returned with the RADIUS accept message. In this case, you see that the QoS attribute is returned along with the RADIUS accept message. Therefore, the user is given the QoS policy of Bronze which overrides the default QoS value set for that SSID.

Related Information

- [Cisco Airespace VSAs on MS IAS RADIUS Server Configuration Example](#)
 - [Cisco Wireless LAN Controller Configuration Guide, Release 4.1](#)
 - [Wireless Support Page](#)
 - [Technical Support & Documentation – Cisco Systems](#)
-

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2008 – 2009 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Jan 21, 2008

Document ID: 96103
