Network Working Group
Request for Comments: 2866
Category: Informational
Obsoletes: 2139

C. Rigney
Livingston

June 2000

RADIUS Accounting

Status of this Memo

Copyright Notice

Abstract

This document describes a protocol for carrying accounting information between a Network Access Server and a shared Accounting Server.

Implementation Note

This memo documents the RADIUS Accounting protocol. The early deployment of RADIUS Accounting was done using UDP port number 1646, which conflicts with the "sa-msg-port" service. The officially assigned port number for RADIUS Accounting is 1813.

Table of Contents

1. Introduction

Managing dispersed serial line and modem pools for large numbers of users can create the need for significant administrative support. Since modem pools are by definition a link to the outside world, they require careful attention to security, authorization and accounting. This can be best achieved by managing a single "database" of users, which allows for authentication (verifying user name and password) as well as configuration information detailing the type of service to deliver to the user (for example, SLIP, PPP, telnet, rlogin).

The RADIUS (Remote Authentication Dial In User Service) document [2] specifies the RADIUS protocol used for Authentication and Authorization. This memo extends the use of the RADIUS protocol to cover delivery of accounting information from the Network Access Server (NAS) to a RADIUS accounting server.

This document obsoletes RFC 2139 [1]. A summary of the changes between this document and RFC 2139 is available in the "Change Log" appendix.

Key features of RADIUS Accounting are:

Client/Server Model

A Network Access Server (NAS) operates as a client of the RADIUS accounting server. The client is responsible for passing user accounting information to a designated RADIUS accounting server.

The RADIUS accounting server is responsible for receiving the accounting request and returning a response to the client indicating that it has successfully received the request.

The RADIUS accounting server can act as a proxy client to other kinds of accounting servers.

Network Security

Transactions between the client and RADIUS accounting server are authenticated through the use of a shared secret, which is never sent over the network.

Extensible Protocol

All transactions are comprised of variable length Attribute-Length-Value 3-tuples. New attribute values can be added without disturbing existing implementations of the protocol.

1.1. Specification of Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [3]. These key words mean the same thing whether capitalized or not.

1.2. Terminology

This document uses the following terms:

service The NAS provides a service to the dial-in user, such as PPP or Telnet.

session Each service provided by the NAS to a dial-in user constitutes a session, with the beginning of the session defined as the point where service is first provided and the end of the session defined as the point where service is ended. A user may have multiple sessions in parallel or series if the NAS supports that, with each session generating a separate start and stop accounting record with its own Acct-Session-Id.

silently discard
This means the implementation discards the packet without further processing. The implementation SHOULD provide the capability of logging the error, including the contents of the silently discarded packet, and SHOULD record the event in a statistics counter.

2. Operation

When a client is configured to use RADIUS Accounting, at the start of service delivery it will generate an Accounting Start packet describing the type of service being delivered and the user it is being delivered to, and will send that to the RADIUS Accounting server, which will send back an acknowledgement that the packet has been received. At the end of service delivery the client will generate an Accounting Stop packet describing the type of service that was delivered and optionally statistics such as elapsed time, input and output octets, or input and output packets. It will send that to the RADIUS Accounting server, which will send back an acknowledgement that the packet has been received.

The Accounting-Request (whether for Start or Stop) is submitted to the RADIUS accounting server via the network. It is recommended that the client continue attempting to send the Accounting-Request packet until it receives an acknowledgement, using some form of backoff. If no response is returned within a length of time, the request is re-sent a number of times. The client can also forward requests to an alternate server or servers in the event that the primary server is down or unreachable. An alternate server can be used either after a number of tries to the primary server fail, or in a round-robin fashion. Retry and fallback algorithms are the topic of current research and are not specified in detail in this document.

The RADIUS accounting server MAY make requests of other servers in order to satisfy the request, in which case it acts as a client.

If the RADIUS accounting server is unable to successfully record the accounting packet it MUST NOT send an Accounting-Response acknowledgment to the client.

2.1. Proxy

See the "RADIUS" RFC [2] for information on Proxy RADIUS. Proxy Accounting RADIUS works the same way, as illustrated by the following example.

1. The NAS sends an accounting-request to the forwarding server.

2. The forwarding server logs the accounting-request (if desired), adds its Proxy-State (if desired) after any other Proxy-State attributes, updates the Request Authenticator, and forwards the request to the remote server.

3. The remote server logs the accounting-request (if desired), copies all Proxy-State attributes in order and unmodified from the request to the response packet, and sends the accounting-response to the forwarding server.

4. The forwarding server strips the last Proxy-State (if it added one in step 2), updates the Response Authenticator and sends the accounting-response to the NAS.

A forwarding server MUST not modify existing Proxy-State or Class attributes present in the packet.

A forwarding server may either perform its forwarding function in a pass through manner, where it sends retransmissions on as soon as it gets them, or it may take responsibility for retransmissions, for example in cases where the network link between forwarding and remote server has very different characteristics than the link between NAS and forwarding server.

Extreme care should be used when implementing a proxy server that takes responsibility for retransmissions so that its retransmission policy is robust and scalable.

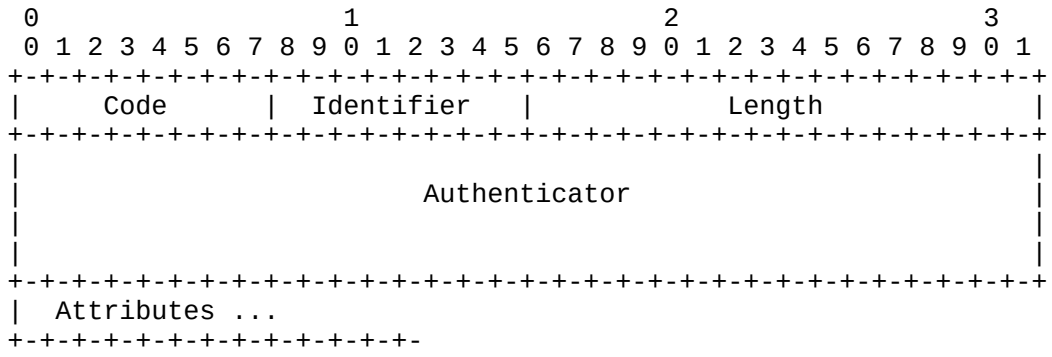
3. Packet Format

Exactly one RADIUS Accounting packet is encapsulated in the UDP Data field [4], where the UDP Destination Port field indicates 1813 (decimal).

When a reply is generated, the source and destination ports are reversed.

This memo documents the RADIUS Accounting protocol. The early deployment of RADIUS Accounting was done using UDP port number 1646, which conflicts with the "sa-msg-port" service. The officially assigned port number for RADIUS Accounting is 1813.

A summary of the RADIUS data format is shown below. The fields are transmitted from left to right.



Code

The Code field is one octet, and identifies the type of RADIUS packet. When a packet is received with an invalid Code field, it is silently discarded.

RADIUS Accounting Codes (decimal) are assigned as follows:

4	Accounting-Request
5	Accounting-Response

Identifier

The Identifier field is one octet, and aids in matching requests and replies. The RADIUS server can detect a duplicate request if it has the same client source IP address and source UDP port and Identifier within a short span of time.

Length

The Length field is two octets. It indicates the length of the packet including the Code, Identifier, Length, Authenticator and Attribute fields. Octets outside the range of the Length field MUST be treated as padding and ignored on reception. If the packet is shorter than the Length field indicates, it MUST be silently discarded. The minimum length is 20 and maximum length is 4095.

Authenticator

The Authenticator field is sixteen (16) octets. The most significant octet is transmitted first. This value is used to authenticate the messages between the client and RADIUS accounting server.

Request Authenticator

In Accounting-Request Packets, the Authenticator value is a 16 octet MD5 [5] checksum, called the Request Authenticator.

The NAS and RADIUS accounting server share a secret. The Request Authenticator field in Accounting-Request packets contains a one-way MD5 hash calculated over a stream of octets consisting of the Code + Identifier + Length + 16 zero octets + request attributes + shared secret (where + indicates concatenation). The 16 octet MD5 hash value is stored in the Authenticator field of the Accounting-Request packet.

Note that the Request Authenticator of an Accounting-Request can not be done the same way as the Request Authenticator of a RADIUS Access-Request, because there is no User-Password attribute in an Accounting-Request.

Response Authenticator

The Authenticator field in an Accounting-Response packet is called the Response Authenticator, and contains a one-way MD5 hash calculated over a stream of octets consisting of the Accounting-Response Code, Identifier, Length, the Request Authenticator field from the Accounting-Request packet being replied to, and the response attributes if any, followed by the shared secret. The resulting 16 octet MD5 hash value is stored in the Authenticator field of the Accounting-Response packet.

Attributes

Attributes may have multiple instances, in such a case the order of attributes of the same type SHOULD be preserved. The order of attributes of different types is not required to be preserved.

4. Packet Types

The RADIUS packet type is determined by the Code field in the first octet of the packet.

4.1. Accounting-Request

Description

Accounting-Request packets are sent from a client (typically a Network Access Server or its proxy) to a RADIUS accounting server, and convey information used to provide accounting for a service provided to a user. The client transmits a RADIUS packet with the Code field set to 4 (Accounting-Request).

Upon receipt of an Accounting-Request, the server MUST transmit an Accounting-Response reply if it successfully records the accounting packet, and MUST NOT transmit any reply if it fails to record the accounting packet.

Any attribute valid in a RADIUS Access-Request or Access-Accept packet is valid in a RADIUS Accounting-Request packet, except that the following attributes MUST NOT be present in an Accounting-Request: User-Password, CHAP-Password, Reply-Message, State. Either NAS-IP-Address or NAS-Identifier MUST be present in a RADIUS Accounting-Request. It SHOULD contain a NAS-Port or NAS-Port-Type attribute or both unless the service does not involve a port or the NAS does not distinguish among its ports.

If the Accounting-Request packet includes a Framed-IP-Address, that attribute MUST contain the IP address of the user. If the Access-Accept used the special values for Framed-IP-Address telling the NAS to assign or negotiate an IP address for the user, the Framed-IP-Address (if any) in the Accounting-Request MUST contain the actual IP address assigned or negotiated.

A summary of the Accounting-Request packet format is shown below.

The fields are transmitted from left to right.

Code

4 for Accounting-Request.

Identifier

The Identifier field MUST be changed whenever the content of the Attributes field changes, and whenever a valid reply has been received for a previous request. For retransmissions where the contents are identical, the Identifier MUST remain unchanged.

Note that if Acct-Delay-Time is included in the attributes of an Accounting-Request then the Acct-Delay-Time value will be updated when the packet is retransmitted, changing the content of the Attributes field and requiring a new Identifier and Request Authenticator.

Request Authenticator

The Request Authenticator of an Accounting-Request contains a 16-octet MD5 hash value calculated according to the method described in "Request Authenticator" above.

Attributes

The Attributes field is variable in length, and contains a list of Attributes.

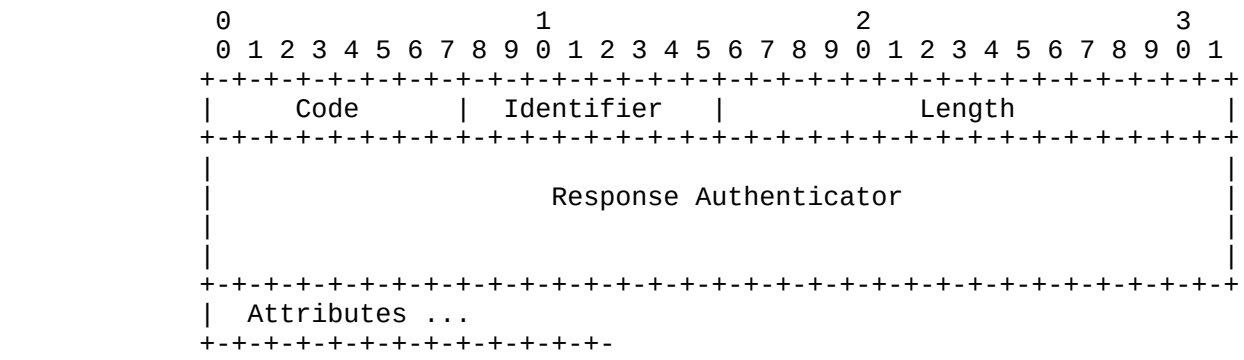
4.2. Accounting-Response

Description

Accounting-Response packets are sent by the RADIUS accounting server to the client to acknowledge that the Accounting-Request has been received and recorded successfully. If the Accounting-Request was recorded successfully then the RADIUS accounting server MUST transmit a packet with the Code field set to 5 (Accounting-Response). On reception of an Accounting-Response by the client, the Identifier field is matched with a pending Accounting-Request. The Response Authenticator field MUST contain the correct response for the pending Accounting-Request. Invalid packets are silently discarded.

A RADIUS Accounting-Response is not required to have any attributes in it.

A summary of the Accounting-Response packet format is shown below. The fields are transmitted from left to right.



Code

5 for Accounting-Response.

Identifier

The Identifier field is a copy of the Identifier field of the Accounting-Request which caused this Accounting-Response.

Response Authenticator

The Response Authenticator of an Accounting-Response contains a 16-octet MD5 hash value calculated according to the method described in "Response Authenticator" above.

Attributes

The Attributes field is variable in length, and contains a list of zero or more Attributes.

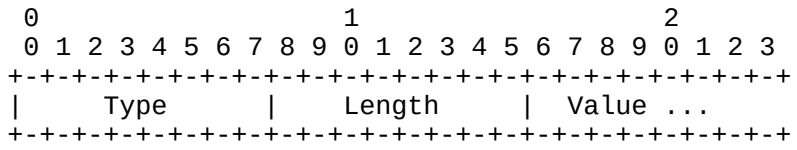
5. Attributes

RADIUS Attributes carry the specific authentication, authorization and accounting details for the request and response.

Some attributes MAY be included more than once. The effect of this is attribute specific, and is specified in each attribute description.

The end of the list of attributes is indicated by the Length of the RADIUS packet.

A summary of the attribute format is shown below. The fields are transmitted from left to right.



Type

The Type field is one octet. Up-to-date values of the RADIUS Type field are specified in the most recent "Assigned Numbers" RFC [6]. Values 192-223 are reserved for experimental use, values 224-240 are reserved for implementation-specific use, and values 241-255 are reserved and should not be used. This specification concerns the following values:

1-39	(refer to RADIUS document [2])
40	Acct-Status-Type
41	Acct-Delay-Time
42	Acct-Input-Octets
43	Acct-Output-Octets
44	Acct-Session-Id
45	Acct-Authentic
46	Acct-Session-Time
47	Acct-Input-Packets
48	Acct-Output-Packets
49	Acct-Terminate-Cause
50	Acct-Multi-Session-Id
51	Acct-Link-Count
60+	(refer to RADIUS document [2])

Length

The Length field is one octet, and indicates the length of this attribute including the Type, Length and Value fields. If an attribute is received in an Accounting-Request with an invalid Length, the entire request MUST be silently discarded.

Value

The Value field is zero or more octets and contains information specific to the attribute. The format and length of the Value field is determined by the Type and Length fields.

Note that none of the types in RADIUS terminate with a NUL (hex 00). In particular, types "text" and "string" in RADIUS do not terminate with a NUL (hex 00). The Attribute has a length field and does not use a terminator. Text contains UTF-8 encoded 10646

[7] characters and String contains 8-bit binary data. Servers and servers and clients MUST be able to deal with embedded nulls. RADIUS implementers using C are cautioned not to use strcpy() when handling strings.

The format of the value field is one of five data types. Note that type "text" is a subset of type "string."

text 1-253 octets containing UTF-8 encoded 10646 [7] characters. Text of length zero (0) MUST NOT be sent; omit the entire attribute instead.

string 1-253 octets containing binary data (values 0 through 255 decimal, inclusive). Strings of length zero (0) MUST NOT be sent; omit the entire attribute instead.

address 32 bit value, most significant octet first.

integer 32 bit unsigned value, most significant octet first.

time 32 bit unsigned value, most significant octet first -- seconds since 00:00:00 UTC, January 1, 1970. The standard Attributes do not use this data type but it is presented here for possible use in future attributes.

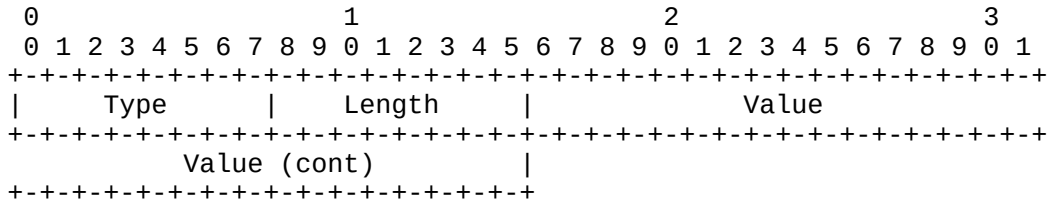
5.1. Acct-Status-Type

Description

This attribute indicates whether this Accounting-Request marks the beginning of the user service (Start) or the end (Stop).

It MAY be used by the client to mark the start of accounting (for example, upon booting) by specifying Accounting-On and to mark the end of accounting (for example, just before a scheduled reboot) by specifying Accounting-Off.

A summary of the Acct-Status-Type attribute format is shown below. The fields are transmitted from left to right.



Type

40 for Acct-Status-Type.

Length

6

Value

The Value field is four octets.

1	Start
2	Stop
3	Interim-Update
7	Accounting-On
8	Accounting-Off
9-14	Reserved for Tunnel Accounting
15	Reserved for Failed

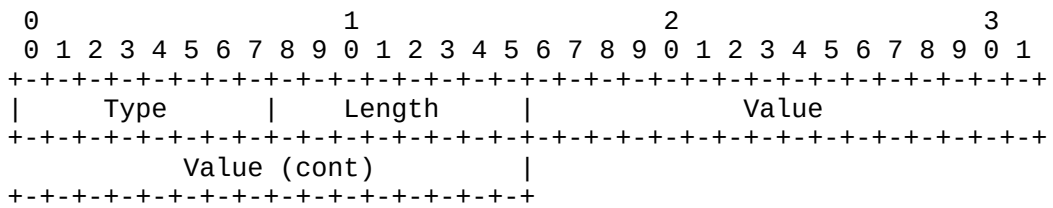
5.2. Acct-Delay-Time

Description

This attribute indicates how many seconds the client has been trying to send this record for, and can be subtracted from the time of arrival on the server to find the approximate time of the event generating this Accounting-Request. (Network transit time is ignored.)

Note that changing the Acct-Delay-Time causes the Identifier to change; see the discussion under Identifier above.

A summary of the Acct-Delay-Time attribute format is shown below. The fields are transmitted from left to right.



Type

41 for Acct-Delay-Time.

Length

6

Value

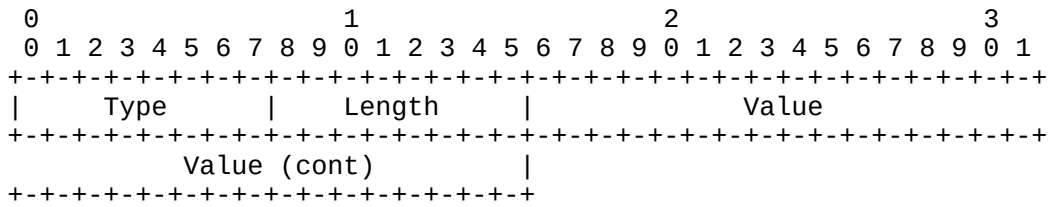
The Value field is four octets.

5.3. Acct-Input-Octets

Description

This attribute indicates how many octets have been received from the port over the course of this service being provided, and can only be present in Accounting-Request records where the Acct-Status-Type is set to Stop.

A summary of the Acct-Input-Octets attribute format is shown below. The fields are transmitted from left to right.



Type

42 for Acct-Input-Octets.

Length

6

Value

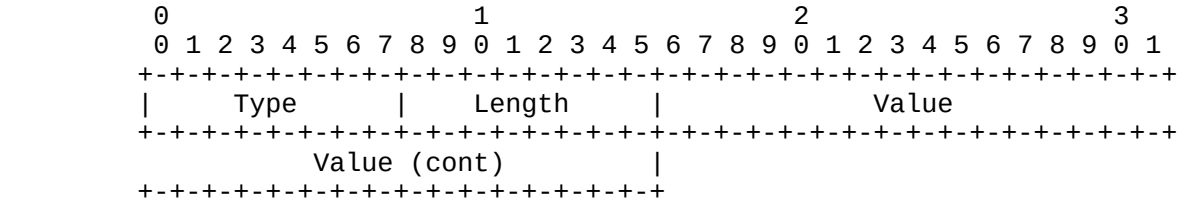
The Value field is four octets.

5.4. Acct-Output-Octets

Description

This attribute indicates how many octets have been sent to the port in the course of delivering this service, and can only be present in Accounting-Request records where the Acct-Status-Type is set to Stop.

A summary of the Acct-Output-Octets attribute format is shown below. The fields are transmitted from left to right.



Type

43 for Acct-Output-Octets.

Length

6

Value

The Value field is four octets.

5.5. Acct-Session-Id

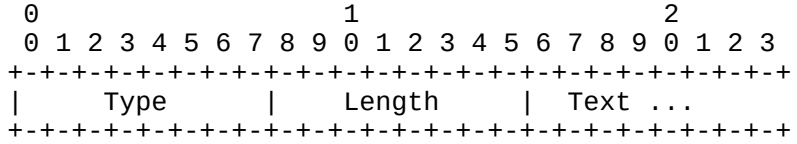
Description

This attribute is a unique Accounting ID to make it easy to match start and stop records in a log file. The start and stop records for a given session MUST have the same Acct-Session-Id. An Accounting-Request packet MUST have an Acct-Session-Id. An Access-Request packet MAY have an Acct-Session-Id; if it does, then the NAS MUST use the same Acct-Session-Id in the Accounting-Request packets for that session.

The Acct-Session-Id SHOULD contain UTF-8 encoded 10646 [7] characters.

For example, one implementation uses a string with an 8-digit upper case hexadecimal number, the first two digits increment on each reboot (wrapping every 256 reboots) and the next 6 digits counting from 0 for the first person logging in after a reboot up to 2^24-1, about 16 million. Other encodings are possible.

A summary of the Acct-Session-Id attribute format is shown below. The fields are transmitted from left to right.



Type

44 for Acct-Session-Id.

Length

>= 3

String

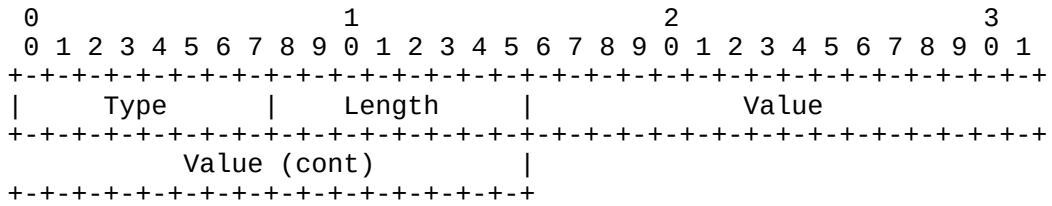
The String field SHOULD be a string of UTF-8 encoded 10646 [7] characters.

5.6. Acct-Authentic

Description

This attribute MAY be included in an Accounting-Request to indicate how the user was authenticated, whether by RADIUS, the NAS itself, or another remote authentication protocol. Users who are delivered service without being authenticated SHOULD NOT generate Accounting records.

A summary of the Acct-Authentic attribute format is shown below. The fields are transmitted from left to right.



Type

45 for Acct-Authentic.

Length

6

Value

The Value field is four octets.

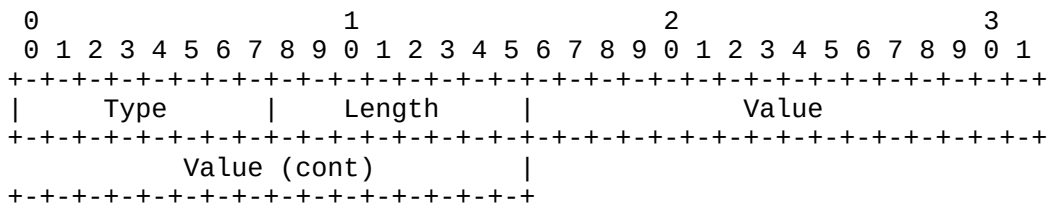
1	RADIUS
2	Local
3	Remote

5.7. Acct-Session-Time

Description

This attribute indicates how many seconds the user has received service for, and can only be present in Accounting-Request records where the Acct-Status-Type is set to Stop.

A summary of the Acct-Session-Time attribute format is shown below. The fields are transmitted from left to right.



Type

46 for Acct-Session-Time.

Length

6

Value

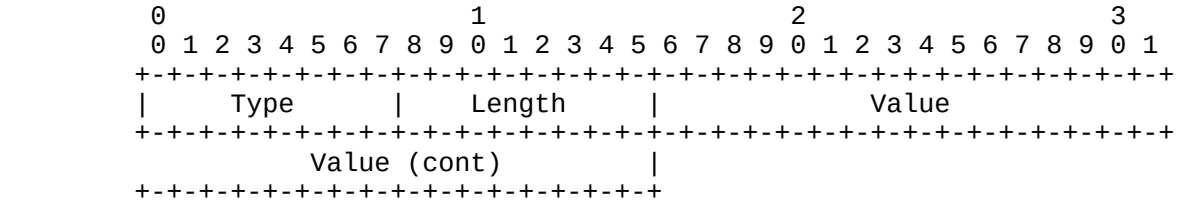
The Value field is four octets.

5.8. Acct-Input-Packets

Description

This attribute indicates how many packets have been received from the port over the course of this service being provided to a Framed User, and can only be present in Accounting-Request records where the Acct-Status-Type is set to Stop.

A summary of the Acct-Input-packets attribute format is shown below. The fields are transmitted from left to right.



Type

47 for Acct-Input-Packets.

Length

6

Value

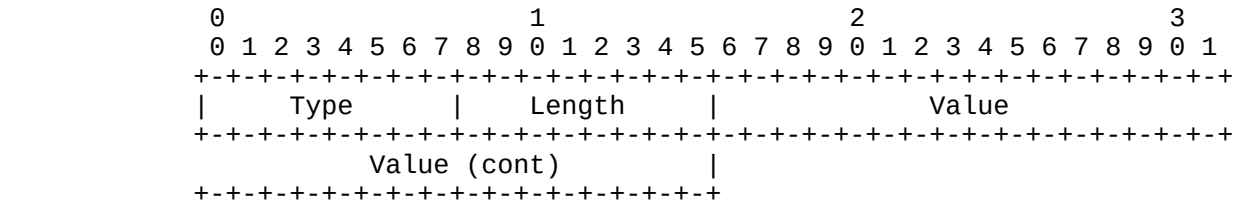
The Value field is four octets.

5.9. Acct-Output-Packets

Description

This attribute indicates how many packets have been sent to the port in the course of delivering this service to a Framed User, and can only be present in Accounting-Request records where the Acct-Status-Type is set to Stop.

A summary of the Acct-Output-Packets attribute format is shown below. The fields are transmitted from left to right.



Type

48 for Acct-Output-Packets.

Length

6

Value

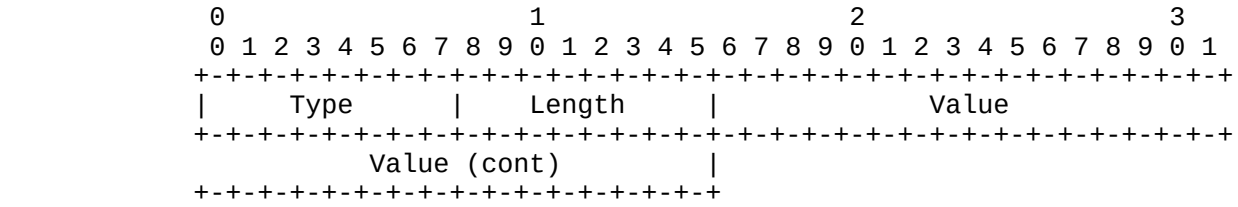
The Value field is four octets.

5.10. Acct-Terminate-Cause

Description

This attribute indicates how the session was terminated, and can only be present in Accounting-Request records where the Acct-Status-Type is set to Stop.

A summary of the Acct-Terminate-Cause attribute format is shown below. The fields are transmitted from left to right.



Type

49 for Acct-Terminate-Cause

Length

6

Value

The Value field is four octets, containing an integer specifying the cause of session termination, as follows:

1 User Request
2 Lost Carrier
3 Lost Service
4 Idle Timeout
5 Session Timeout
6 Admin Reset
7 Admin Reboot
8 Port Error
9 NAS Error
10 NAS Request
11 NAS Reboot
12 Port Unneeded
13 Port Preempted
14 Port Suspended
15 Service Unavailable
16 Callback
17 User Error
18 Host Request

The termination causes are as follows:

User Request	User requested termination of service, for example with LCP Terminate or by logging out.
Lost Carrier	DCD was dropped on the port.
Lost Service	Service can no longer be provided; for example, user's connection to a host was interrupted.
Idle Timeout	Idle timer expired.
Session Timeout	Maximum session length timer expired.
Admin Reset	Administrator reset the port or session.

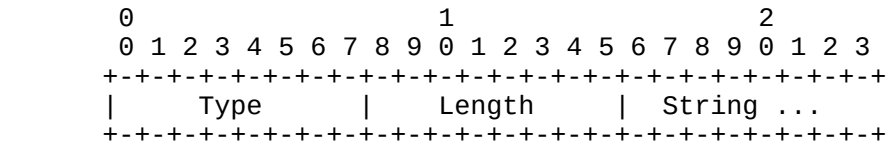
Admin Reboot	Administrator is ending service on the NAS, for example prior to rebooting the NAS.
Port Error	NAS detected an error on the port which required ending the session.
NAS Error	NAS detected some error (other than on the port) which required ending the session.
NAS Request	NAS ended session for a non-error reason not otherwise listed here.
NAS Reboot	The NAS ended the session in order to reboot non-administratively ("crash").
Port Unneeded	NAS ended session because resource usage fell below low-water mark (for example, if a bandwidth-on-demand algorithm decided that the port was no longer needed).
Port Preempted	NAS ended session in order to allocate the port to a higher priority use.
Port Suspended	NAS ended session to suspend a virtual session.
Service Unavailable	NAS was unable to provide requested service.
Callback	NAS is terminating current session in order to perform callback for a new session.
User Error	Input from user is in error, causing termination of session.
Host Request	Login Host terminated session normally.

5.11. Acct-Multi-Session-Id

Description

This attribute is a unique Accounting ID to make it easy to link together multiple related sessions in a log file. Each session linked together would have a unique Acct-Session-Id but the same Acct-Multi-Session-Id. It is strongly recommended that the Acct-Multi-Session-Id contain UTF-8 encoded 10646 [7] characters.

A summary of the Acct-Session-Id attribute format is shown below. The fields are transmitted from left to right.



Type

50 for Acct-Multi-Session-Id.

Length

>= 3

String

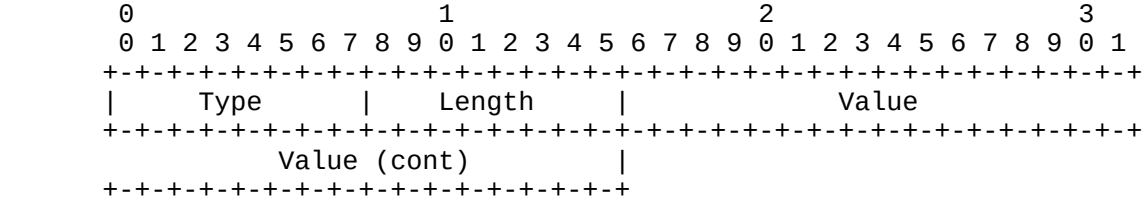
The String field SHOULD contain UTF-8 encoded 10646 [7] characters.

5.12. Acct-Link-Count

Description

This attribute gives the count of links which are known to have been in a given multilink session at the time the accounting record is generated. The NAS MAY include the Acct-Link-Count attribute in any Accounting-Request which might have multiple links.

A summary of the Acct-Link-Count attribute format is show below. The fields are transmitted from left to right.



Type

51 for Acct-Link-Count.

Length

6

Value

The Value field is four octets, and contains the number of links seen so far in this Multilink Session.

It may be used to make it easier for an accounting server to know when it has all the records for a given Multilink session. When the number of Accounting-Requests received with Acct-Status-Type = Stop and the same Acct-Multi-Session-Id and unique Acct-Session-Id's equals the largest value of Acct-Link-Count seen in those Accounting-Requests, all Stop Accounting-Requests for that Multilink Session have been received.

An example showing 8 Accounting-Requests should make things clearer. For clarity only the relevant attributes are shown, but additional attributes containing accounting information will also be present in the Accounting-Request.

Multi-Session-Id	Session-Id	Status-Type	Link-Count
"10"	"10"	Start	1
"10"	"11"	Start	2
"10"	"11"	Stop	2
"10"	"12"	Start	3
"10"	"13"	Start	4
"10"	"12"	Stop	4
"10"	"13"	Stop	4
"10"	"10"	Stop	4

5.13. Table of Attributes

The following table provides a guide to which attributes may be found in Accounting-Request packets. No attributes should be found in Accounting-Response packets except Proxy-State and possibly Vendor-Specific.

#	Attribute
0-1	User-Name
0	User-Password
0	CHAP-Password

0-1	NAS-IP-Address [Note 1]
0-1	NAS-Port
0-1	Service-Type
0-1	Framed-Protocol
0-1	Framed-IP-Address
0-1	Framed-IP-Netmask
0-1	Framed-Routing
0+	Filter-Id
0-1	Framed-MTU
0+	Framed-Compression
0+	Login-IP-Host
0-1	Login-Service
0-1	Login-TCP-Port
0	Reply-Message
0-1	Callback-Number
0-1	Callback-Id
0+	Framed-Route
0-1	Framed-IPX-Network
0	State
0+	Class
0+	Vendor-Specific
0-1	Session-Timeout
0-1	Idle-Timeout
0-1	Termination-Action
0-1	Called-Station-Id
0-1	Calling-Station-Id
0-1	NAS-Identifier [Note 1]
0+	Proxy-State
0-1	Login-LAT-Service
0-1	Login-LAT-Node
0-1	Login-LAT-Group
0-1	Framed-AppleTalk-Link
0-1	Framed-AppleTalk-Network
0-1	Framed-AppleTalk-Zone
1	Acct-Status-Type
0-1	Acct-Delay-Time
0-1	Acct-Input-Octets
0-1	Acct-Output-Octets
1	Acct-Session-Id
0-1	Acct-Authentic
0-1	Acct-Session-Time
0-1	Acct-Input-Packets
0-1	Acct-Output-Packets
0-1	Acct-Terminate-Cause
0+	Acct-Multi-Session-Id
0+	Acct-Link-Count
0	CHAP-Challenge

0-1	NAS-Port-Type
0-1	Port-Limit
0-1	Login-LAT-Port

[Note 1] An Accounting-Request MUST contain either a NAS-IP-Address or a NAS-Identifier (or both).

The following table defines the above table entries.

0	This attribute MUST NOT be present
0+	Zero or more instances of this attribute MAY be present.
0-1	Zero or one instance of this attribute MAY be present.
1	Exactly one instance of this attribute MUST be present.

6. IANA Considerations

The Packet Type Codes, Attribute Types, and Attribute Values defined in this document are registered by the Internet Assigned Numbers Authority (IANA) from the RADIUS name spaces as described in the "IANA Considerations" section of RFC 2865 [2], in accordance with BCP 26 [8].

7. Security Considerations

Security issues are discussed in sections concerning the authenticator included in accounting requests and responses, using a shared secret which is never sent over the network.

8. Change Log

US-ASCII replaced by UTF-8.

Added notes on Proxy.

Framed-IP-Address should contain the actual IP address of the user.

If Acct-Session-ID was sent in an access-request, it must be used in the accounting-request for that session.

New values added to Acct-Status-Type.

Added an IANA Considerations section.

Updated references.

Text strings identified as a subset of string, to clarify use of UTF-8.

9. References

[1] Rigney, C., "RADIUS Accounting", RFC 2139, April 1997.

[2] Rigney, C., Willens, S., Rubens, A. and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

[3] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March, 1997.

[4] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[5] Rivest, R. and S. Dusse, "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[6] Reynolds, J. and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October 1994.

[7] Yergeau, F., "UTF-8, a transformation format of ISO 10646", RFC 2279, January 1998.

[8] Alvestrand, H. and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

10. Acknowledgements

RADIUS and RADIUS Accounting were originally developed by Steve Willens of Livingston Enterprises for their PortMaster series of Network Access Servers.

11. Chair’s Address

The RADIUS working group can be contacted via the current chair:

Carl Rigney
Livingston Enterprises
4464 Willow Road
Pleasanton, California 94588

Phone: +1 925 737 2100
EMail: cdr@telemancy.com

12. Author's Address

Questions about this memo can also be directed to:

Carl Rigney
Livingston Enterprises
4464 Willow Road
Pleasanton, California 94588

EMail: cdr@telemancy.com

13. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.