Oracle® Communications Network Charging and Control

RADIUS Control Agent Protocol Implementation Conformance Statement

Release 15.2

January 2026

Copyright

Contents

Chapter 1

Compliance to RFC 2865 (Remote Authentication Dial In User Service (RADIUS)) 1

Chapter 2

Compliance to RFC 2866 (RADIUS Accounting) 59

Chapter 3

Compliance to RFC 2869 (RADIUS Extensions) 93

Chapter 4

Chapter 5

Chapter 6

About This Document

Purpose

To define the compliance of RCA to various RADIUS standards documents.

Audience

This document is written to provide information to third parties communicating with RCA via the RADIUS protocol.

Document Conventions

Conformance indication

RCA is an implementation of a RADIUS server, for the purposes of providing credit control. It does not perform password authentication or IP address allocation within the Oracle usage.

This document states RCA compliance to the RFCs and other standards documents listed in the Related documents section.

Mostly, the compliance is stated by reproducing the relevant document and then annotating it where Oracle either does not conform or uses the standard in a different way as:

NCC Implementation Notes

The note text provides the RCA non-conformance details, and details of the standards relevant to a RADIUS server that are not supported by RCA.

Chapter 1

Compliance to RFC 2865 (Remote Authentication Dial In User Service (RADIUS))

Overview

Introduction

This chapter identifies the compliance of Oracle to RFC 2865 - Remote Authentication Dial In User Service (RADIUS).

In this chapter

This chapter contains the following topics.

Remote Authentication Dial In User Service (RADIUS)

RADIUS

Network Working Group	C. Rigney
Request for Comments: 2865	S. Willens
Obsoletes: 2138	Livingston
Category: Standards Track	A. Rubens
	Merit
	W. Simpson
	Daydreamer
	June 2000

Remote Authentication Dial In User Service (RADIUS)

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

IESG Note:

This protocol is widely implemented and used. Experience has shown that it can suffer degraded performance and lost data when used in large scale systems, in part because it does not include provisions for congestion control. Readers of this document may find it beneficial to track the progress of the IETF's AAA working group, which may develop a successor protocol that better addresses the scaling and congestion control issues.

Abstract

This document describes a protocol for carrying authentication, authorization, and configuration information between a Network Access Server which desires to authenticate its links and a shared Authentication Server.

NCC Implementation Note

This memo documents the RADIUS protocol. The early deployment of RADIUS was done using UDP port number 1645, which conflicts with the "datametrics" service. The officially assigned port number for RADIUS is 1812. Therefore, in RCA, the default port for core Radius messages is 1812 but any other port number can be configured instead.

Table of Contents

1. Introduction

1. Introduction

This document obsoletes RFC 2138 [1]. A summary of the changes between this document and RFC 2138 is available in the "Change Log" appendix.

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).

NCC Implementation Note

RCA does not have a database of users or consult any database of users. It is assumed that authentication is done by some sort of proxy before Access_request messages are sent to RCA. RCAs role is purely credit control, not authentication.

Key features of RADIUS are:

Client/Server Model

A Network Access Server (NAS) operates as a client of RADIUS. The client is responsible for passing user information to designated RADIUS servers, and then acting on the response which is returned.

RADIUS servers are responsible for receiving user connection requests, authenticating the user, and then returning all configuration information necessary for the client to deliver service to the user.

NCC Implementation Note

The following is not supported by RCA: A RADIUS server can act as a proxy client to other RADIUS servers or other kinds of authentication servers.

Network Security

Transactions between the client and RADIUS server are authenticated through the use of a shared secret, which is never sent over the network. In addition, any user passwords are sent encrypted between the client and RADIUS server, to eliminate the possibility that someone snooping on an unsecure network could determine a user's password.

Flexible Authentication Mechanisms

(Not supported)

NCC Implementation Note

The following is not supported by RCA:

- The RADIUS server can support a variety of methods to authenticate a user. When it is provided with the user name and original password given by the user, it can support PPP PAP or CHAP, UNIX login, and other authentication mechanisms.

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 BCP 14 [2]. These key words mean the same thing whether capitalised or not.

An implementation is not compliant if it fails to satisfy one or more of the must or must not requirements for the protocols it implements.

An implementation that satisfies all the must, must not, should and should not requirements for its protocols is said to be "unconditionally compliant"; one that satisfies all the must and must not requirements but not all the should or should not requirements for its protocols is said to be "conditionally compliant".

A NAS that does not implement a given service MUST NOT implement the RADIUS attributes for that service. For example, a NAS that is unable to offer ARAP service MUST NOT implement the RADIUS attributes for ARAP. A NAS MUST treat a RADIUS access-accept authorising an unavailable service as an access-reject instead.

1.2. Terminology

This document frequently 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.
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

2. Operation

When a client is configured to use RADIUS, any user of the client presents authentication information to the client. This might be with a customisable login prompt, where the user is expected to enter their username and password. Alternatively, the user might use a link framing protocol such as the Point-to-Point Protocol (PPP), which has authentication packets which carry this information.

Once the client has obtained such information, it may choose to authenticate using RADIUS. To do so, the client creates an "Access- Request" containing such Attributes as the user's name, the user's password, the ID of the client and the Port ID which the user is accessing. When a password is present, it is hidden using a method based on the RSA Message Digest Algorithm MD5 [3].

The Access-Request is submitted to the RADIUS server via the network.

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.

NCC Implementation Notes:

RCA does not validate the sending client – it accepts clients with any shared secret. If such validation is done it must be done by a proxy.

RCA does not support the following for RADIUS:

- Once the RADIUS server receives the request, it validates the sending client. A request from a client for which the RADIUS server does not have a shared secret MUST be silently discarded.
- If the client is valid, the RADIUS server consults a database of users to find the user whose name matches the request. The user entry in the database contains a list of requirements which must be met to allow access for the user. This always includes verification of the password, but can also specify the client(s) or port(s) to which the user is allowed access.
- The RADIUS server MAY make requests of other servers in order to satisfy the request, in which case it acts as a client.

If any Proxy-State attributes were present in the Access-Request, they MUST be copied unmodified and in order into the response packet.

NCC Implementation Notes:

In 3GPP2 mode, RCA cannot deal with Proxy-State attribute.

In parameterised mode, RCA can deal with Proxy-State attributes to a limited extent. RCA can do one of the following:

- Copy a single proxy-State attribute of any type from the Access-Request to the response packet.
- Copy multiple Proxy-State attribute which are ASCII strings from the Access-Request to the response packet, provided each ASCII string starts with a unique string which distinguished that Proxy-State attribute from all the others. In this case, the order of the attributes id dependent on the order specified in configuration, and not on the order of the attributes in the Access-Request.

Other Attributes can be placed before, after, or even between the Proxy-State attributes.

If any condition is not met, the RADIUS server sends an "Access- Reject" response indicating that this user request is invalid. If desired, the server MAY include a text message in the Access-Reject which MAY be displayed by the client to the user. No other Attributes (except Proxy-State) are permitted in an Access-Reject.

NCC Implementation Note

The RCA does not support the following for RADIUS:

- If all conditions are met and the RADIUS server wishes to issue a challenge to which the user must respond, the RADIUS server sends an "Access-Challenge" response. It MAY include a text message to be displayed by the client to the user prompting for a response to the challenge, and MAY include a State attribute.

If the client receives an Access-Challenge and supports challenge/response it MAY display the text message, if any, to the user, and then prompt the user for a response. The client then re- submits its original Access-Request with a new request ID, with the User-Password Attribute replaced by the response (encrypted), and including the State Attribute from the Access-Challenge, if any. Only 0 or 1 instances of the State Attribute SHOULD be present in a request. The server can respond to this new Access- Request with either an Access-Accept, an Access-Reject, or another Access-Challenge.

If all conditions are met, the list of configuration values for the user are placed into an "Access-Accept" response. These values include the type of service (for example: SLIP, PPP, Login User) and all necessary values to deliver the desired service. For SLIP and PPP, this may include values such as IP address, subnet mask, MTU, desired compression, and desired packet filter identifiers. For character mode users, this may include values such as desired protocol and host.

2.1. Challenge/Response

NCC Implementation Notes:

The RCA does not support the following:

- In challenge/response authentication, the user is given an unpredictable number and challenged to encrypt it and give back the result. Authorised users are equipped with special devices such as smart cards or software that facilitate calculation of the correct response with ease. Unauthorised users, lacking the appropriate device or software and lacking knowledge of the secret key necessary to emulate such a device or software, can only guess at the response.
- The Access-Challenge packet typically contains a Reply-Message including a challenge to be displayed to the user, such as a numeric value unlikely ever to be repeated. Typically this is obtained from an external server that knows what type of authenticator is in the possession of the authorised user and can therefore choose a random or non-repeating pseudorandom number of an appropriate radix and length.

- The user then enters the challenge into his device (or software) and it calculates a response, which the user enters into the client which forwards it to the RADIUS server via a second Access-Request. If the response matches the expected response the RADIUS server replies with an Access-Accept, otherwise an Access-Reject.

Example: The NAS sends an Access-Request packet to the RADIUS Server with NAS-Identifier, NAS-Port, User-Name, User-Password (which may just be a fixed string like "challenge" or ignored). The server sends back an Access-Challenge packet with State and a Reply-Message along the lines of "Challenge 12345678, enter your response at the prompt" which the NAS displays. The NAS prompts for the response and sends a NEW Access-Request to the server (with a new ID) with NAS- Identifier, NAS-Port, User-Name, User-Password (the response just entered by the user, encrypted), and the same State Attribute that came with the Access-Challenge. The server then sends back either an Access-Accept or Access-Reject based on whether the response matches the required value, or it can even send another Access-Challenge.

2.2. Interoperation with PAP and CHAP

For PAP, the NAS takes the PAP ID and password and sends them in an Access-Request packet as the User-Name and User-Password. The NAS MAY include the Attributes Service-Type = Framed-User and Framed-Protocol = PPP as a hint to the RADIUS server that PPP service is expected.

For CHAP, the NAS generates a random challenge (preferably 16octets) and sends it to the user, who returns a CHAP response along with a CHAP ID and CHAP username. The NAS then sends an Access-Request packet to the RADIUS server with the CHAP username as the User-Name and with the CHAP ID and CHAP response as the CHAP-Password (Attribute 3). The random challenge can either be included in the CHAP-Challenge attribute or, if it is 16 octets long, it can be placed in the Request Authenticator field of the Access-Request packet. The NAS MAY include the Attributes Service-Type = Framed- User and Framed-Protocol = PPP as a hint to the RADIUS server that PPP service is expected.

NCC Implementation Notes:

The RCA does not support the following:

- The RADIUS server looks up a password based on the User-Name, encrypts the challenge using MD5 on the CHAP ID octet, that password, and the CHAP challenge (from the CHAP-Challenge attribute if present, otherwise from the Request Authenticator), and compares that result to the CHAP-Password. If they match, the server sends back an Access-Accept, otherwise it sends back an Access-Reject.
- If the RADIUS server is unable to perform the requested authentication it MUST return an Access-Reject. For example, CHAP requires that the user's password be available in cleartext to the server so that it can encrypt the CHAP challenge and compare that to the CHAP response. If the password is not available in cleartext to the RADIUS server then the server MUST send an Access-Reject to the client.

2.3. Proxy

With proxy RADIUS, one RADIUS server receives an authentication (or accounting) request from a RADIUS client (such as a NAS), forwards the request to a remote RADIUS server, receives the reply from the remote server, and sends that reply to the client, possibly with changes to reflect local administrative policy. A common use for proxy RADIUS is roaming. Roaming permits two or more administrative entities to allow each other's users to dial in to either entity's network for service.

The NAS sends its RADIUS access-request to the "forwarding server" which forwards it to the "remote server". The remote server sends a response (Access-Accept, Access-Reject, or Access-Challenge) back to the forwarding server, which sends it back to the NAS. The User-Name attribute MAY contain a Network Access Identifier [8] for RADIUS Proxy operations. The choice of which server receives the forwarded request SHOULD be based on the authentication "realm". The authentication realm MAY be the realm part of a Network Access Identifier (a "named realm"). Alternatively, the choice of which server receives the forwarded request MAY be based on whatever other criteria the forwarding server is configured to use, such as Called- Station-Id (a "numbered realm").

NCC Implementation Note:

RCA does not act as a forwarding server.

A RADIUS server can function as both a forwarding server and a remote server, serving as a forwarding server for some realms and a remote server for other realms. One forwarding server can act as a forwarder for any number of remote servers. A remote server can have any number of servers forwarding to it and can provide authentication for any number of realms. One forwarding server can forward to another forwarding server to create a chain of proxies, although care must be taken to avoid introducing loops.

The following scenario illustrates a proxy RADIUS communication between a NAS and the forwarding and remote RADIUS servers:

1. A NAS sends its access-request to the forwarding server.
2. The forwarding server forwards the access-request to the remote server.
3. The remote server sends an access-accept, access-reject or access-challenge back to the forwarding server. For this example, an access-accept is sent.
4. The forwarding server sends the access-accept to the NAS.

The forwarding server MUST treat any Proxy-State attributes already in the packet as opaque data. Its operation MUST NOT depend on the content of Proxy-State attributes added by previous servers.

If there are any Proxy-State attributes in the request received from the client, the forwarding server MUST include those Proxy-State attributes in its reply to the client. The forwarding server MAY include the Proxy-State attributes in the access-request when it forwards the request, or MAY omit them in the forwarded request. If the forwarding server omits the Proxy-State attributes in the forwarded access-request, it MUST attach them to the response before sending it to the client.

We now examine each step in more detail.

1. A NAS sends its access-request to the forwarding server. The forwarding server decrypts the User-Password, if present, using the shared secret it knows for the NAS. If a CHAP-Password attribute is present in the packet and no CHAP-Challenge attribute is present, the forwarding server MUST leave the Request- Authenticator untouched or copy it to a CHAP-Challenge attribute.

   " The forwarding server MAY add one Proxy-State attribute to the packet. (It MUST NOT add more than one.) If it adds a Proxy- State, the Proxy-State MUST appear after any other Proxy-States in the packet. The forwarding server MUST NOT modify any other Proxy-States that were in the packet (it may choose not to forward them, but it MUST NOT change their contents). The forwarding server MUST NOT change the order of any attributes of the same type, including Proxy-State.
2. The forwarding server encrypts the User-Password, if present, using the secret it shares with the remote server, sets the Identifier as needed, and forwards the access-request to the remote server.
3. The remote server (if the final destination) verifies the user using User-Password, CHAP-Password, or such method as future extensions may dictate, and returns an access-accept, access- reject or access-challenge back to the forwarding server. For this example, an access-accept is sent. The remote server MUST copy all Proxy-State attributes (and only the Proxy-State attributes) in order from the access-request to the response packet, without modifying them.

4 The forwarding server verifies the Response Authenticator using the secret it shares with the remote server, and silently discards the packet if it fails verification. If the packet passes verification, the forwarding server removes the last Proxy-State (if it attached one), signs the Response Authenticator using the secret it shares with the NAS, restores the Identifier to match the one in the original request by the NAS, and sends the access- accept to the NAS.

A forwarding server MAY need to modify attributes to enforce local policy. Such policy is outside the scope of this document, with the following restrictions. A forwarding server MUST not modify existing Proxy-State, State, or Class attributes present in the packet.

Implementers of forwarding servers should consider carefully which values it is willing to accept for Service-Type. Careful consideration must be given to the effects of passing along Service- Types of NAS-Prompt or Administrative in a proxied Access-Accept, and implementers may wish to provide mechanisms to block those or other service types, or other attributes. Such mechanisms are outside the scope of this document.

2.4. Why UDP?

A frequently asked question is why RADIUS uses UDP instead of TCP as a transport protocol. UDP was chosen for strictly technical reasons.

There are a number of issues which must be understood. RADIUS is a transaction based protocol which has several interesting characteristics:

1 If the request to a primary Authentication server fails, a secondary server must be queried.

To meet this requirement, a copy of the request must be kept above the transport layer to allow for alternate transmission. This means that retransmission timers are still required.

2 The timing requirements of this particular protocol are significantly different than TCP provides.

At one extreme, RADIUS does not require a "responsive" detection of lost data. The user is willing to wait several seconds for the authentication to complete. The generally aggressive TCP retransmission (based on average round trip time) is not required, nor is the acknowledgement overhead of TCP.

At the other extreme, the user is not willing to wait several minutes for authentication. Therefore the reliable delivery of TCP data two minutes later is not useful. The faster use of an alternate server allows the user to gain access before giving up.

3 The stateless nature of this protocol simplifies the use of UDP.

Clients and servers come and go. Systems are rebooted, or are power cycled independently. Generally this does not cause a problem and with creative timeouts and detection of lost TCP connections, code can be written to handle anomalous events. UDP however completely eliminates any of this special handling. Each client and server can open their UDP transport just once and leave it open through all types of failure events on the network.

4 UDP simplifies the server implementation.

In the earliest implementations of RADIUS, the server was single threaded. This means that a single request was received, processed, and returned. This was found to be unmanageable in environments where the back-end security mechanism took real time (1 or more seconds). The server request queue would fill and in environments where hundreds of people were being authenticated every minute, the request turn-around time increased to longer than users were willing to wait (this was especially severe when a specific lookup in a database or over DNS took 30 or more seconds). The obvious solution was to make the server multi- threaded. Achieving this was simple with UDP. Separate processes were spawned to serve each request and these processes could respond directly to the client NAS with a simple UDP packet to the original transport of the client.

It's not all a panacea. As noted, using UDP requires one thing which is built into TCP: with UDP we must artificially manage retransmission timers to the same server, although they don't require the same attention to timing provided by TCP. This one penalty is a small price to pay for the advantages of UDP in this protocol.

Without TCP we would still probably be using tin cans connected by string.   But for this particular protocol, UDP is a better choice.

2.5. Retransmission Hints

NCC Implementation Notes:

The following is not implemented for RCA:

- If the RADIUS server and alternate RADIUS server share the same shared secret, it is OK to retransmit the packet to the alternate RADIUS server with the same ID and Request Authenticator, because the content of the attributes haven't changed.

RCA does not support failover to another server mod-transaction. Therefore, once the Access-Request has been sent to a particular server, all subsequent messages from the client must be sent to the same server.

If you want to use a new Request Authenticator when sending to the alternate server, you may.

If you change the contents of the User-Password attribute (or any other attribute), you need a new Request Authenticator and therefore a new ID.

If the NAS is retransmitting a RADIUS request to the same server as before, and the attributes haven't changed, you MUST use the same Request Authenticator, ID, and source port.   If any attributes have changed, you MUST use a new Request Authenticator and ID.

A NAS MAY use the same ID across all servers, or MAY keep track of IDs separately for each server, it is up to the implementer.   If a NAS needs more than 256 IDs for outstanding requests, it MAY use additional source ports to send requests from, and keep track of IDs for each source port.   This allows up to 16 million or so outstanding requests at one time to a single server.

NCC Implementation Note:

RCA can handle approximately 100 thousand outstanding requests, dependent on hardware constraints and configuration.   This is unlikely to be a limiting factor on a real system.

2.6. Keep-Alives Considered Harmful

Some implementers have adopted the practice of sending test RADIUS requests to see if a server is alive.   This practice is strongly discouraged, since it adds to load and harms scalability without providing any additional useful information.   Since a RADIUS request is contained in a single datagram, in the time it would take you to send a ping you could just send the RADIUS request, and getting a reply tells you that the RADIUS server is up.   If you do not have a RADIUS request to send, it does not matter if the server is up or not, because you are not using it.

If you want to monitor your RADIUS server, use SNMP.   That's what SNMP is for.

3. Packet Format

3. Packet Format

Exactly one RADIUS packet is encapsulated in the UDP Data field [4], where the UDP Destination Port field indicates 1812 (decimal).

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

This memo documents the RADIUS protocol.   The early deployment of RADIUS was done using UDP port number 1645, which conflicts with the "datametrics" service. The officially assigned port number for RADIUS is 1812.

NCC Implementation Note:

In RCA, the default port for core Radius messages is 1812 but any other port number can be configured instead.

A summary of the RADIUS data format is shown below.   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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         Authenticator                         |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Attributes ...
+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Field	Description
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 Codes (decimal) are assigned as follows: 1 Access-Request 2 Access-Accept 3 Access-Reject 4 Accounting-Request 5 Accounting-Response 255 Reserved **NCC Implementation Note:** The following RADUIS codes are not supported: 11 Access-Challenge 12 Status-Server (experimental) 13 Status-Client (experimental) Codes 4 and 5 are covered in the RADIUS Accounting document [5]. Codes 12 and 13 are reserved for possible use, but are not further mentioned here.
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 4096.

Field	Description
Authenticator	The Authenticator field is sixteen (16) octets. The most significant octet is transmitted first. This value is used to authenticate the reply from the RADIUS server, and is used in the password hiding algorithm.
Request Authenticator	In Access-Request Packets, the Authenticator value is a 16 octet random number, called the Request Authenticator. The value SHOULD be unpredictable and unique over the lifetime of a secret (the password shared between the client and the RADIUS server), since repetition of a request value in conjunction with the same secret would permit an attacker to reply with a previously intercepted response. Since it is expected that the same secret MAY be used to authenticate with servers in disparate geographic regions, the Request Authenticator field SHOULD exhibit global and temporal uniqueness. The Request Authenticator value in an Access-Request packet SHOULD also be unpredictable, lest an attacker trick a server into responding to a predicted future request, and then use the response to masquerade as that server to a future Access- Request.
	Although protocols such as RADIUS are incapable of protecting against theft of an authenticated session via realtime active wiretapping attacks, generation of unique unpredictable requests can protect against a wide range of active attacks against authentication. The NAS and RADIUS server share a secret. That shared secret followed by the Request Authenticator is put through a one-way MD5 hash to create a 16 octet digest value which is stored with the password entered by the user, and the xored result placed in the User-Password attribute in the Access-Request packet. See the entry for User-Password in the section on Attributes for a more detailed description.
Response Authenticator	The value of the Authenticator field in Access-Accept, Access-Reject, and Access-Challenge packets is called the Response Authenticator, and contains a one-way MD5 hash calculated over a stream of octets consisting of: the RADIUS packet, beginning with the Code field, including the Identifier, the Length, the Request Authenticator field from the Access-Request packet, and the response Attributes, followed by the shared secret. That is, ResponseAuth = MD5(Code+ID+Length+RequestAuth+Attributes+Secret) where + denotes concatenation.

Field	Description
Administrative Note	The secret (password shared between the client and the RADIUS server) SHOULD be at least as large and unguessable as a well-chosen password. It is preferred that the secret be at least 16 octets. This is to ensure a sufficiently large range for the secret to provide protection against exhaustive search attacks. The secret MUST NOT be empty (length 0) since this would allow packets to be trivially forged. A RADIUS server MUST use the source IP address of the RADIUS UDP packet to decide which shared secret to use, so that RADIUS requests can be proxied. When using a forwarding proxy, the proxy must be able to alter the packet as it passes through in each direction - when the proxy forwards the request, the proxy MAY add a Proxy-State Attribute, and when the proxy forwards a response, it MUST remove its Proxy-State Attribute if it added one. Proxy-State is always added or removed after any other Proxy-States, but no other assumptions regarding its location within the list of attributes can be made. Since Access-Accept and Access-Reject replies are authenticated on the entire packet contents, the stripping of the Proxy-State attribute invalidates the signature in the packet - so the proxy has to re-sign it. Further details of RADIUS proxy implementation are outside the scope of this document.

4. Packet Types

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

4.1. Access-Request

Description

Access-Request packets are sent to a RADIUS server, and convey information used to determine whether a user is allowed access to a specific NAS, and any special services requested for that user. An implementation wishing to authenticate a user MUST transmit a RADIUS packet with the Code field set to 1 (Access-Request).

Upon receipt of an Access-Request from a valid client, an appropriate reply MUST be transmitted.

An Access-Request SHOULD contain a User-Name attribute. It MUST contain either a NAS-IP-Address attribute or a NAS-Identifier attribute (or both).

An Access-Request MUST contain either a User-Password or a CHAP- Password or a State. An Access-Request MUST NOT contain both a User-Password and a CHAP-Password. If future extensions allow other kinds of authentication information to be conveyed, the attribute for that can be used in an Access-Request instead of User-Password or CHAP-Password.

An Access-Request SHOULD contain a NAS-Port or NAS-Port-Type attribute or both unless the type of access being requested does not involve a port or the NAS does not distinguish among its ports.

An Access-Request MAY contain additional attributes as a hint to the server, but the server is not required to honor the hint.

When a User-Password is present, it is hidden using a method based on the RSA Message Digest Algorithm MD5 [3].

A summary of the Access-Request packet format is shown below. 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                     Request Authenticator                     |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Attributes ...
   +-+-+-+-+-+-+-+-+-+-+-+-+-
```

Code 1 for Access-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, the Identifier MUST remain unchanged.

Request Authenticator The Request Authenticator value MUST be changed each time a new Identifier is used.

Attributes The Attribute field is variable in length, and contains the list of Attributes that are required for the type of service, as well as any desired optional Attributes.

4.2. Access-Accept

Description

Access-Accept packets are sent by the RADIUS server, and provide specific configuration information necessary to begin delivery of service to the user. If all Attribute values received in an Access-Request are acceptable then the RADIUS implementation MUST transmit a packet with the Code field set to 2 (Access-Accept).

On reception of an Access-Accept, the Identifier field is matched with a pending Access-Request. The Response Authenticator field MUST contain the correct response for the pending Access-Request.

Invalid packets are silently discarded.

A summary of the Access-Accept packet format is shown below. 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                     Response Authenticator                    |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Attributes ...
+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Code 2 for Access-Accept.

Identifier The Identifier field is a copy of the Identifier field of the Access-Request which caused this Access-Accept.

Request Authenticator The Response Authenticator value is calculated from the Access-Request value, as described earlier.

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

4.3. Access-Reject

Description

If any value of the received Attributes is not acceptable, then the RADIUS server MUST transmit a packet with the Code field set to 3 (Access-Reject). It MAY include one or more Reply-Message Attributes with a text message which the NAS MAY display to the user.

NCC Implementation Notes:

- In 3GPP2 mode, RCA includes exactly one Reply-Message attribute, as described above.
- In parameterised mode, RCA can be configured to either send the text message in a Reply-Message attribute, send it in another op level attribute or not send it at all.

A summary of the Access-Reject packet format is shown below. 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                     Response Authenticator                    |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Attributes ...
+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Code 3 for Access-Reject.

Identifier The Identifier field is a copy of the Identifier field of the Access-Request which caused this Access-Reject.

Response Authenticator The Response Authenticator value is calculated from the Access-Request value, as described earlier.

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

4.4. Access-Challenge

NCC Implementation Notes:

RCA does note support the following:

Description

If the RADIUS server desires to send the user a challenge requiring a response, then the RADIUS server MUST respond to the Access-Request by transmitting a packet with the Code field set to 11 (Access-Challenge).

The Attributes field MAY have one or more Reply-Message Attributes, and MAY have a single State Attribute, or none. Vendor-Specific, Idle-Timeout, Session-Timeout and Proxy-State attributes MAY also be included.   No other Attributes defined in this document are permitted in an Access-Challenge.

On receipt of an Access-Challenge, the Identifier field is matched with a pending Access-Request. Additionally, the Response Authenticator field MUST contain the correct response for the pending Access-Request.   Invalid packets are silently discarded.

If the NAS does not support challenge/response, it MUST treat an Access-Challenge as though it had received an Access-Reject instead.

If the NAS supports challenge/response, receipt of a valid Access-Challenge indicates that a new Access-Request SHOULD be sent.   The NAS MAY display the text message, if any, to the user, and then prompt the user for a response.   It then sends its original Access-Request with a new request ID and Request Authenticator, with the User-Password Attribute replaced by the user's response (encrypted), and including the State Attribute from the Access-Challenge, if any.   Only 0 or 1 instances of the State Attribute can be present in an Access-Request.

A NAS which supports PAP MAY forward the Reply-Message to the dialing client and accept a PAP response which it can use as though the user had entered the response.   If the NAS cannot do so, it MUST treat the Access-Challenge as though it had received an Access-Reject instead.

A summary of the Access-Challenge packet format is shown below.   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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                     Response Authenticator                    |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Attributes ...
+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Code	11 for Access-Challenge.
Identifier	The Identifier field is a copy of the Identifier field of the Access-Request which caused this Access-Challenge.
Response Authenticator	The Response Authenticator value is calculated from the Access- Request value, as described earlier.
Attributes	The Attributes field is variable in length, and contains a list of zero or more Attributes.

5. Attributes

5. Attributes

NCC Implementation Notes:

- In 3GPP2 mode, RCA can only cope with a fixed set of attributes. I.e. in 3GPP2 mode RCA deals with all the attributes listed in this document unless this document explicitly states that the attribute is not used in 3GPP2 mode.
- In parameterized mode, RCA can deal with any core Radius attribute or vendor specific attribute (whether or not they are listed in this document) provided that: - It is (or can be treated as) of type text, octets, IPv4 address or number - The attribute is at the top level. (I.e. RCA in parameterized mode cannot cope with attributes within attributes.

In the above note, the phrase "RCA can deal with [an attribute]" means that, in parameterized mode, RCA can use the attribute in decision making concerning credit control, in a configurable way, and / or store the attribute to be sent out again in Radius messages and / or send the attribute in outgoing messages on a per message type basis.

RADIUS Attributes carry the specific authentication, authorization, information and configuration details for the request and reply.

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

Some Attributes MAY be included more than once.   The effect of this is Attribute specific, and is specified in each Attribute description.   A summary table is provided at the end of the "Attributes" section.

NCC Implementation Notes:

RCA cannot cope with receiving multiple attributes of the same type, except Vendor-Specific attributes which it can cope with.

In 3GPP2 mode, RCA will never send a Radius message with more than one attribute of the same type, except for Vendor-Specific attributes.

In parameterised mode, RCA can be configured to send more than one attribute of the same type, for any type.

If multiple Attributes with the same Type are present, the order of Attributes with the same Type MUST be preserved by any proxies.   The order of Attributes of different Types is not required to be preserved. A RADIUS server or client MUST NOT have any dependencies on the order of attributes of different types.   A RADIUS server or client MUST NOT require attributes of the same type to be contiguous.

Where an Attribute's description limits which kinds of packet it can be contained in, this applies only to the packet types defined in this document, namely Access-Request, Access-Accept, Access-Reject and Access-Challenge (Codes 1, 2, 3, and 11).   Other documents defining other packet types may also use Attributes described here. To determine which Attributes are allowed in Accounting-Request and Accounting-Response packets (Codes 4 and 5) refer to the RADIUS Accounting document [5].

Likewise where packet types defined here state that only certain Attributes are permissible in them, future memos defining new Attributes should indicate which packet types the new Attributes may be present in.

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

```
    0                   1                   2
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
   |     Type      |    Length     |  Value ...
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

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.

NCC Implementation Notes:

In parameterised mode, RCA can be configured to decode or send attributes with any value for Type, regardless of the ranges specified above.

A RADIUS server MAY ignore Attributes with an unknown Type.

NCC Implementation Note:

RCA will ignore attributes with an unknown type.

A RADIUS client MAY ignore Attributes with an unknown Type.

This specification concerns the following values:

1	User-Name
2	User-Password
3	CHAP-Password
4	NAS-IP-Address
5	NAS-Port
6	Service-Type
7	Framed-Protocol
8	Framed-IP-Address
9	Framed-IP-Netmask
10	Framed-Routing
11	Filter-Id
12	Framed-MTU
13	Framed-Compression
14	Login-IP-Host
15	Login-Service
16	Login-TCP-Port
17	(unassigned)
18	Reply-Message
19	Callback-Number
20	Callback-Id
21	(unassigned)
22	Framed-Route
23	Framed-IPX-Network
24	State

25	Class
26	Vendor-Specific
27	Session-Timeout
28	Idle-Timeout
29	Termination-Action
30	Called-Station-Id
31	Calling-Station-Id
32	NAS-Identifier
33	Proxy-State
34	Login-LAT-Service
35	Login-LAT-Node
36	Login-LAT-Group
37	Framed-AppleTalk-Link
38	Framed-AppleTalk-Network
39	Framed-AppleTalk-Zone
40-59	(reserved for accounting)
60	CHAP-Challenge
61	NAS-Port-Type
62	Port-Limit
63	Login-LAT-Port

Length

The Length field is one octet, and indicates the length of this Attribute including the Type, Length and Value fields.

NCC Implementation Note

RCA does not support the following:

If an Attribute is received in an Access-Request but with an invalid Length, an Access-Reject SHOULD be transmitted.   If an Attribute is received in an Access-Accept, Access-Reject or Access-Challenge packet with an invalid length, the packet MUST either be treated as an Access-Reject or else 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.

NCC Implementation Note

RCA cannot cope with embedded nulls in text type attributes.

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. User-Name

Description

This Attribute indicates the name of the user to be authenticated.

It MUST be sent in Access-Request packets if available.

It MAY be sent in an Access-Accept packet, in which case the client SHOULD use the name returned in the Access-Accept packet in all Accounting-Request packets for this session. If the Access-Accept includes Service-Type = Rlogin and the User-Name attribute, a NAS MAY use the returned User-Name when performing the Rlogin function.

NCC Implementation Notes:

- In 3GPP2 mode, RCA includes the User-Name attribute in Disconnect-Request messages only. In 3GPP2 mode, RCA will use the user-Name attribute in the Access-Request attribute to identify the subscriber to be charged (if it is present.)
- In parameterised mode, RCA can be configured to send the User-Name attribute in any message, on a per message type basis.

A summary of the User-Name Attribute format is shown below. The fields are transmitted from left to right.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Length     |  String ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type 1 for User-Name.

Length >= 3

String The String field is one or more octets. The NAS may limit the maximum length of the User-Name but the ability to handle at least 63 octets is recommended.

The format of the username MAY be one of several forms:

text	Consisting only of UTF-8 encoded 10646 [7] characters.

NCC Implementation Note:

RCA does note support the following:

network access identifier	A Network Access Identifier as described in RFC 2486 [8]
distinguished name	A name in ASN.1 form used in Public Key authentication systems.

5.2. User-Password

Description

This Attribute indicates the password of the user to be authenticated, or the user's input following an Access-Challenge. It is only used in Access-Request packets.

On transmission, the password is hidden. The password is first padded at the end with nulls to a multiple of 16 octets. A one- way MD5 hash is calculated over a stream of octets consisting of the shared secret followed by the Request Authenticator. This value is XORed with the first 16 octet segment of the password and placed in the first 16 octets of the String field of the User- Password Attribute.

If the password is longer than 16 characters, a second one-way MD5 hash is calculated over a stream of octets consisting of the shared secret followed by the result of the first xor. That hash is XORed with the second 16 octet segment of the password and placed in the second 16 octets of the String field of the User- Password Attribute.

If necessary, this operation is repeated, with each xor result being used along with the shared secret to generate the next hash to xor the next segment of the password, to no more than 128 characters.

The method is taken from the book "Network Security" by Kaufman, Perlman and Speciner [9] pages 109-110. A more precise explanation of the method follows:

Call the shared secret S and the pseudo-random 128-bit Request Authenticator RA. Break the password into 16-octet chunks p1, p2, etc. with the last one padded at the end with nulls to a 16-octet boundary. Call the ciphertext blocks c(1), c(2), etc. We'll need intermediate values b1, b2, etc.

b1 = MD5(S + RA)	c(1) = p1 xor b1
b2 = MD5(S + c(1))	c(2) = p2 xor b2
.	.
.	.
.	.
bi = MD5(S + c(i-1))	c(i) = pi xor bi

The String will contain c(1)+c(2)+...+c(i) where + denotes concatenation.

On receipt, the process is reversed to yield the original password.

A summary of the User-Password Attribute format is shown below. The fields are transmitted from left to right.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Length     |  String ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type	2 for User-Password.
Length	At least 18 and no larger than 130.
String	The String field is between 16 and 128 octets long, inclusive.

NCC Implementation Note:

RCA does not verify the password.

5.3. CHAP-Password

Description

This Attribute indicates the response value provided by a PPP Challenge-Handshake Authentication Protocol (CHAP) user in response to the challenge. It is only used in Access-Request packets.

The CHAP challenge value is found in the CHAP-Challenge Attribute (60) if present in the packet, otherwise in the Request Authenticator field.

A summary of the CHAP-Password Attribute format is shown below. The fields are transmitted from left to right.

```
 0                   1                   2
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Length     |  CHAP Ident   |  String ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type	3 for CHAP-Password.
Length	19
CHAP Ident	This field is one octet, and contains the CHAP Identifier from the user's CHAP Response.
String	The String field is 16 octets, and contains the CHAP Response from the user.

NCC Implementation Notes:

RCA does not verify the CHAP-Password.

5.4. NAS-IP-Address

Description

This Attribute indicates the identifying IP Address of the NAS which is requesting authentication of the user, and SHOULD be unique to the NAS within the scope of the RADIUS server.   NAS-IP- Address is only used in Access-Request packets.   Either NAS-IP- Address or NAS-Identifier MUST be present in an Access-Request packet.

Note that NAS-IP-Address MUST NOT be used to select the shared secret used to authenticate the request.   The source IP address of the Access-Request packet MUST be used to select the shared secret.

A summary of the NAS-IP-Address Attribute format is shown below.   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     |            Address
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
         Address (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	4 for NAS-IP-Address.
Length	6
Address	The Address field is four octets.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis. In order to comply to RFC 3576, RCA should be configured to add this parameter, if present in the Access-Accept, to Disconnect-request messages.

5.5. NAS-Port

Description

This Attribute indicates the physical port number of the NAS which is authenticating the user.   It is only used in Access-Request packets.   Note that this is using "port" in its sense of a physical connection on the NAS, not in the sense of a TCP or UDP port number.   Either NAS-Port or NAS-Port-Type (61) or both SHOULD be present in an Access-Request packet, if the NAS differentiates among its ports.

A summary of the NAS-Port Attribute format is shown below.   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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	5 for NAS-Port.
Length	6
Value	The Value field is four octets.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.6. Service-Type

Description

This Attribute indicates the type of service the user has requested, or the type of service to be provided. It MAY be used in both Access-Request and Access-Accept packets. A NAS is not required to implement all of these service types, and MUST treat unknown or unsupported Service-Types as though an Access-Reject had been received instead.

A summary of the Service-Type Attribute format is shown below. 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	6 for Service-Type.	
Length	6	
Value	The Value field is four octets.	
	1	Login
	2	Framed
	3	Callback Login
	4	Callback Framed
	5	Outbound
	6	Administrative
	7	NAS Prompt
	8	Authenticate Only
	9	Callback NAS Prompt
	10	Call Check
	11	Callback Administrative

The service types are defined as follows when used in an Access-Accept. When used in an Access-Request, they MAY be considered to be a hint to the RADIUS server that the NAS has reason to believe the user would prefer the kind of service indicated, but the server is not required to honor the hint.

Login The user should be connected to a host.

Framed A Framed Protocol should be started for the User, such as PPP or SLIP.

Callback Login	The user should be disconnected and called back, then connected to a host.
Callback Framed	The user should be disconnected and called back, then a Framed Protocol should be started for the User, such as PPP or SLIP.
Outbound	The user should be granted access to outgoing devices.
Administrative	The user should be granted access to the administrative interface to the NAS from which privileged commands can be executed.
NAS Prompt	The user should be provided a command prompt on the NAS from which non-privileged commands can be executed.
Authenticate Only	Only Authentication is requested, and no authorization information needs to be returned in the Access-Accept (typically used by proxy servers rather than the NAS itself).
Callback NAS Prompt	The user should be disconnected and called back, then provided a command prompt on the NAS from which non-privileged commands can be executed.
Call Check	Used by the NAS in an Access-Request packet to indicate that a call is being received and that the RADIUS server should send back an Access-Accept to answer the call, or an Access-Reject to not accept the call, typically based on the Called-Station-Id or Calling-Station-Id attributes. It is recommended that such Access-Requests use the value of Calling-Station-Id as the value of the User-Name.
Callback Administrative	The user should be disconnected and called back, then granted access to the administrative interface to the NAS from which privileged commands can be executed.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.7. Framed-Protocol

Description

This Attribute indicates the framing to be used for framed access. It MAY be used in both Access-Request and Access-Accept packets.

A summary of the Framed-Protocol Attribute format is shown below. 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	7 for Framed-Protocol.
Length	6
Value	The Value field is four octets.

1	PPP
2	SLIP
3	AppleTalk Remote Access Protocol (ARAP)
4	Gandalf proprietary SingleLink/MultiLink protocol
5	Xylogics proprietary IPX/SLIP
6	X.75 Synchronous

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.8. Framed-IP-Address

This Attribute indicates the address to be configured for the user. It MAY be used in Access-Accept packets. It MAY be ued in an Access-Request packet as a hint by the NAS to the server that it would prefer that address, but the server is not required to honor the hint.

A summary of the Framed-IP-Address Attribute format is shown below. 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     |            Address
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
         Address (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type 8 for Framed-IP-Address.

Length 6

Address The Address field is four octets. The value 0xFFFFFFFF indicates that the NAS Should allow the user to select an address (e.g. Negotiated). The value 0xFFFFFFFE indicates that the NAS should select an address for the user (e.g. Assigned from a pool of addresses kept by the NAS). Other valid values indicate that the NAS should use that value as the user's IP address.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.9. Framed-IP-Netmask

Description

This Attribute indicates the IP netmask to be configured for the user when the user is a router to a network. It MAY be used in Access-Accept packets. It MAY be used in an Access-Request packet as a hint by the NAS to the server that it would prefer that netmask, but the server is not required to honor the hint.

A summary of the Framed-IP-Netmask Attribute format is shown below. 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     |            Address
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
         Address (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	9 for Framed-IP-Netmask.
Length	6
Address	The Address field is four octets specifying the IP netmask of the user.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.10. Framed-Routing

Description

This Attribute indicates the routing method for the user, when the user is a router to a network.   It is only used in Access-Accept packets.

A summary of the Framed-Routing Attribute format is shown below.   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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	10 for Framed-Routing.	
Length	6	
Value	The Value field is four octets.	
	0	None
	1	Send routing packets
	2	Listen for routing packets
	3	Send and Listen

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.11. Filter-Id

Description

This Attribute indicates the name of the filter list for this user.   Zero or more Filter-Id attributes MAY be sent in an Access-Accept packet.

Identifying a filter list by name allows the filter to be used on different NASes without regard to filter-list implementation details.

A summary of the Filter-Id Attribute format is shown below.   The fields are transmitted from left to right.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Length     |  Text ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type 11 for Filter-Id.

Length >= 3

Address The Text field is one or more octets, and its contents are implementation dependent.   It is intended to be human readable and MUST NOT affect operation of the protocol.   It is recommended that the message contain UTF-8 encoded 10646 [7] characters.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.12. Framed-MTU

Description

This Attribute indicates the Maximum Transmission Unit to be configured for the user, when it is not negotiated by some other means (such as PPP).   It MAY be used in Access-Accept packets.   It MAY be used in an Access-Request packet as a hint by the NAS to the server that it would prefer that value, but the server is not required to honor the hint.

A summary of the Framed-MTU Attribute format is shown below.   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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type 12 for Framed-MTU.

Length 6

Value The Value field is four octets.   Despite the size of the field, values range from 64 to 65535.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.13. Framed-Compression

Description

This Attribute indicates a compression protocol to be used for the link. It MAY be used in Access-Accept packets. It MAY be used in an Access-Request packet as a hint to the server that the NAS would prefer to use that compression, but the server is not required to honor the hint.

More than one compression protocol Attribute MAY be sent. It is the responsibility of the NAS to apply the proper compression protocol to appropriate link traffic.

A summary of the Framed-Compression Attribute format is shown below. 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	13 for Framed-Compression.	
Length	6	
Value	The Value field is four octets.	
	0	None
	1	VJ TCP/IP header compression [10]
	2	IPX header compression
	3	Stac-LZS compression

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.14. Login-IP-Host

Description

This Attribute indicates the system with which to connect the user, when the Login-Service Attribute is included. It MAY be used in Access-Accept packets. It MAY be used in an Access-Request packet as a hint to the server that the NAS would prefer to use that host, but the server is not required to honor the hint.

A summary of the Login-IP-Host Attribute format is shown below. 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     |            Address
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
         Address (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	14 for Login-IP-Host.
Length	6

Value	The Address field is four octets. The value 0xFFFFFFFF indicates that the NAS SHOULD allow the user to select an address. The value 0 indicates that the NAS SHOULD select a host to connect the user to. Other values indicate the address the NAS SHOULD connect the user to.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.15. Login-Service

This Attribute indicates the service to use to connect the user to the login host. It is only used in Access-Accept packets.

A summary of the Login-Service Attribute format is shown below. 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          Value (cont)          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	15 for Login-Service.	
Length	6	
Value	The Value field is four octets.	
	0	Telnet
	1	Rlogin
	2	TCP Clear
	3	PortMaster (proprietary)
	4	LAT
	5	X25-PAD
	6	X25-T3POS
	8	TCP Clear Quiet (suppresses any NAS-generated connect string)

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.16. Login-TCP-Port

Description

This Attribute indicates the TCP port with which the user is to be connected, when the Login-Service Attribute is also present. It is only used in Access-Accept packets.

A summary of the Login-TCP-Port Attribute format is shown below. 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type 16 for Login-TCP-Port.

Length 6

Value The Value field is four octets. Despite the size of the field, values range from 0 to 65535.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.17. (unassigned)

Description

ATTRIBUTE TYPE 17 HAS NOT BEEN ASSIGNED.

5.18. Reply-Message

Description

This Attribute indicates text which MAY be displayed to the user.

When used in an:

- Access-Accept, it is the success message.
- Access-Reject, it is the failure message. It MAY indicate a dialog message to prompt the user before another Access-Request attempt.
- Access-Challenge, it MAY indicate a dialog message to prompt the user for a response.

Multiple Reply-Message's MAY be included and if any are displayed, they MUST be displayed in the same order as they appear in the packet.

NCC Implementation Notes:

- In 3GPP2 mode, RCA includes the Reply-Message attribute in Access-Reject messages and in no other message.
- In parameterised mode, RCA can be configured to send this attribute in a Radius message, on a per message type basis.

A summary of the Reply-Message Attribute format is shown below. The fields are transmitted from left to right.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Length     |  Text ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type 18 for Reply-Message.

Length >= 3

Text The Text field is one or more octets, and its contents are implementation dependent. It is intended to be human readable, and MUST NOT affect operation of the protocol. It is recommended that the message contain UTF-8 encoded 10646 [7] characters.

5.19. Callback-Number

Description

This Attribute indicates a dialing string to be used for callback. It MAY be used in Access-Accept packets. It MAY be used in an Access-Request packet as a hint to the server that a Callback service is desired, but the server is not required to honor the hint.

A summary of the Callback-Number Attribute format is shown below. The fields are transmitted from left to right.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Length     |  String ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type 19 for Callback-Number.

Length >= 3

String The String field is one or more octets. The actual format of the information is site or application specific, and a robust implementation SHOULD support the field as undistinguished octets.

The codification of the range of allowed usage of this field is outside the scope of this specification.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.20. Callback-Id

Description

This Attribute indicates the name of a place to be called, to be interpreted by the NAS. It MAY be used in Access-Accept packets.

A summary of the Callback-Id Attribute format is shown below. The fields are transmitted from left to right.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Length     |  String ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type 20 for Callback-Id.

Length >= 3

String

The String field is one or more octets. The actual format of the information is site or application specific, and a robust implementation SHOULD support the field as undistinguished octets.

The codification of the range of allowed usage of this field is outside the scope of this specification.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.21. (unassigned)

Description

ATTRIBUTE TYPE 21 HAS NOT BEEN ASSIGNED.

5.22. Framed-Route

Description

This Attribute provides routing information to be configured for the user on the NAS. It is used in the Access-Accept packet and can appear multiple times.

A summary of the Framed-Route Attribute format is shown below. The fields are transmitted from left to right.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Length     |  Text ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type

22 for Framed-Route.

Length

>= 3

Text

The Text field is one or more octets, and its contents are implementation dependent. It is intended to be human readable and MUST NOT affect operation of the protocol. It is recommended that the message contain UTF-8 encoded 10646 [7] characters.

For IP routes, it SHOULD contain a destination prefix in dotted quad form optionally followed by a slash and a decimal length specifier stating how many high order bits of the prefix to use. That is followed by a space, a gateway address in dotted quad form, a space, and one or more metrics separated by spaces. For example, "192.168.1.0/24 192.168.1.1 1 2 -1 3 400". The length specifier may be omitted, in which case it defaults to 8 bits for class A prefixes, 16 bits for class B prefixes, and 24 bits for class C prefixes. For example, "192.168.1.0 192.168.1.1 1".

Whenever the gateway address is specified as "0.0.0.0" the IP address of the user SHOULD be used as the gateway address.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.23. Framed-IPX-Network

Description

This Attribute indicates the IPX Network number to be configured for the user. It is used in Access-Accept packets.

A summary of the Framed-IPX-Network Attribute format is shown below. 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	23 for Framed-IPX-Network.
Length	6
Value3	The Value field is four octets. The value 0xFFFFFFFE indicates that the NAS should select an IPX network for the user (e.g. assigned from a pool of one or more IPX networks kept by the NAS). Other values should be used as the IPX network for the link to the user.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.24. State

Description

This Attribute is available to be sent by the server to the client in an Access-Challenge and MUST be sent unmodified from the client to the server in the new Access-Request reply to that challenge, if any.

This Attribute is available to be sent by the server to the client in an Access-Accept that also includes a Termination-Action Attribute with the value of RADIUS-Request. If the NAS performs the Termination-Action by sending a new Access-Request upon termination of the current session, it MUST include the State attribute unchanged in that Access-Request.

In either usage, the client MUST NOT interpret the attribute locally. A packet must have only zero or one State Attribute. Usage of the State Attribute is implementation dependent.

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

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Length     |  String ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type	24 for State.
Length	>= 3
String	The String field is one or more octets. The actual format of the information is site or application specific, and a robust implementation SHOULD support the field as undistinguished octets.

The codification of the range of allowed usage of this field is outside the scope of this specification.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.25. Class

Description

This Attribute is available to be sent by the server to the client in an Access-Accept and SHOULD be sent unmodified by the client to the accounting server as part of the Accounting-Request packet if accounting is supported.   The client MUST NOT interpret the attribute locally.

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

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Length     |  String ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type 25 for Class.

Length >= 3

String The String field is one or more octets.   The actual format of the information is site or application specific, and a robust implementation SHOULD support the field as undistinguished octets.

The codification of the range of allowed usage of this field is outside the scope of this specification.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.26. Vendor-Specific

Description

This Attribute is available to allow vendors to support their own extended Attributes not suitable for general usage.   It MUST not affect the operation of the RADIUS protocol.

Servers not equipped to interpret the vendor-specific information sent by a client MUST ignore it (although it may be reported). Clients which do not receive desired vendor-specific information SHOULD make an attempt to operate without it, although they may do so (and report they are doing so) in a degraded mode.

A summary of the Vendor-Specific Attribute format is shown below.   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)           |  String...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type 26 for Vendor-Specific.

Length >= 7

Vendor-Id The high-order octet is 0 and the low-order 3 octets are the SMI Network Management Private Enterprise Code of the Vendor in network byte order, as defined in the "Assigned Numbers" RFC [6].

String The String field is one or more octets. The actual format of the information is site or application specific, and a robust implementation SHOULD support the field as undistinguished octets.

The codification of the range of allowed usage of this field is outside the scope of this specification.

It SHOULD be encoded as a sequence of vendor type / vendor length / value fields, as follows. The Attribute-Specific field is dependent on the vendor's definition of that attribute. An example encoding of the Vendor-Specific attribute using this method follows:

```
 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 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Attribute-Specific...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Multiple subattributes MAY be encoded within a single Vendor- Specific attribute, although they do not have to be.

NCC Implementation Notes:

RCA cannot cope with a Vendor-Specific attribute containing multiple sub-attributes, i.e. RCA will not send Vendor-specific attributes containing multiple sub-attributes and RCA will ignore everything apart from the first sub-attribute when decoding vendor-specific attribute.

5.27. Session-Timeout

Description

This Attribute sets the maximum number of seconds of service to be provided to the user before termination of the session or prompt. This Attribute is available to be sent by the server to the client in an Access-Accept or Access-Challenge.

A summary of the Session-Timeout Attribute format is shown below. 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type 27 for Session-Timeout.

Length 6

Value The field is 4 octets, containing a 32-bit unsigned integer with the maximum number of seconds this user should be allowed to remain connected by the NAS.

NCC Implementation Notes:

In both modes of operation, RCA can be configured to set or not set the Session-Timeout in Access-Accept messages. The value used for the session timeout does NOT depend on the amount of credit the subscriber has – it is configured to be the same value for all sessions.

5.28. Idle-Timeout

Description

This Attribute sets the maximum number of consecutive seconds of idle connection allowed to the user before termination of the session or prompt. This Attribute is available to be sent by the server to the client in an Access-Accept or Access-Challenge.

A summary of the Idle-Timeout Attribute format is shown below. 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          Value (cont)          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type 28 for Idle-Timeout.

Length 6

Value The field is 4 octets, containing a 32-bit unsigned integer with the maximum number of consecutive seconds of idle time this user should be permitted before being disconnected by the NAS.

NCC Implementation Notes:

- In 3GPP2 mode, RCA can be configured to set or not set the Idle-Timeout in Access-Accept messages. The value used for the idletimeout does NOT depend on the amount of credit the subscriber has – it is configured to be the same value for all sessions.
- In parameterized mode, there is no special treatment for the Idle-Timeout parameter.

5.29. Termination-Action

Description

This Attribute indicates what action the NAS should take when the specified service is completed. It is only used in Access-Accept packets.

A summary of the Termination-Action Attribute format is shown below. 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          Value (cont)          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type 29 for Termination-Action.

Length 6

Value The Value field is four octets.

0 Default

1 RADIUS-Request

If the Value is set to RADIUS-Request, upon termination of the specified service the NAS MAY send a new Access-Request to the RADIUS server, including the State attribute if any.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.30. Called-Station-Id

Description

This Attribute allows the NAS to send in the Access-Request packet the phone number that the user called, using Dialed Number Identification (DNIS) or similar technology. Note that this may be different from the phone number the call comes in on. It is only used in Access-Request packets.

A summary of the Called-Station-Id Attribute format is shown below. The fields are transmitted from left to right.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Length     |  String ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type 30 for Called-Station-Id.

Length >= 3

String The String field is one or more octets, containing the phone number that the user's call came in on.

The actual format of the information is site or application specific. UTF-8 encoded 10646 [7] characters are recommended, but a robust implementation SHOULD support the field as undistinguished octets.

The codification of the range of allowed usage of this field is outside the scope of this specification.

NCC Implementation Notes:

- In 3GPP2 mode, RCA decodes this attribute from the Access-Request (if it is present) and may use it to decide how much to charge the subscriber.
- In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis. RCA may be configured to use this attribute in deciding how much to charge the subscriber.

5.31. Calling-Station-Id

Description

This Attribute allows the NAS to send in the Access-Request packet the phone number that the call came from, using Automatic Number Identification (ANI) or similar technology. It is only used in Access-Request packets.

A summary of the Calling-Station-Id Attribute format is shown below. The fields are transmitted from left to right.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Length     |  String ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type 31 for Calling-Station-Id.

Length >= 3

String The String field is one or more octets, containing the phone number that the user placed the call from.

The actual format of the information is site or application specific. UTF-8 encoded 10646 [7] characters are recommended, but a robust implementation SHOULD support the field as undistinguished octets.

The codification of the range of allowed usage of this field is outside the scope of this specification.

NCC Implementation Notes:

- In 3GPP2 mode, RCA decodes this attribute from the Access-Request (if it is present) and uses it to identify the subscriber to be charged, if the user-Name attribute uis not present in the Access-Request.
- In parameterized mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis. RCA may be configured to use this attribute to identify the subscriber. It may also be configured to use this attribute in deciding how much to charge the subscriber.

5.32. NAS-Identifier

Description

This Attribute contains a string identifying the NAS originating the Access-Request. It is only used in Access-Request packets. Either NAS-IP-Address or NAS-Identifier MUST be present in an Access-Request packet.

Note that NAS-Identifier MUST NOT be used to select the shared secret used to authenticate the request. The source IP address of the Access-Request packet MUST be used to select the shared secret.

A summary of the NAS-Identifier Attribute format is shown below. The fields are transmitted from left to right.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Length     |  String ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type 32 for NAS-Identifier.

Length >= 3

String The String field is one or more octets, and should be unique to the NAS within the scope of the RADIUS server. For example, a fully qualified domain name would be suitable as a NAS-Identifier.

The actual format of the information is site or application specific, and a robust implementation SHOULD support the field as undistinguished octets.

The codification of the range of allowed usage of this field is outside the scope of this specification.

NCC Implementation Notes:

- In 3GPP2 mode, RCA assumes this attribute is text and writes it to the trace file if the session is being traced. It also adds this attribute to any Disconnect-Request messages it sends.
- In parameterized mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis. In order to comply to RFC 3576, RCA should be configured to add this parameter, if present in the Access-Accept, to Disconnect-request messages.

5.33. Proxy-State

Description

This Attribute is available to be sent by a proxy server to another server when forwarding an Access-Request and MUST be returned unmodified in the Access-Accept, Access-Reject or Access-Challenge. When the proxy server receives the response to its request, it MUST remove its own Proxy-State (the last Proxy- State in the packet) before forwarding the response to the NAS.

If a Proxy-State Attribute is added to a packet when forwarding the packet, the Proxy-State Attribute MUST be added after any existing Proxy-State attributes.

The content of any Proxy-State other than the one added by the current server should be treated as opaque octets and MUST NOT affect operation of the protocol.

Usage of the Proxy-State Attribute is implementation dependent. A description of its function is outside the scope of this specification.

A summary of the Proxy-State Attribute format is shown below. The fields are transmitted from left to right.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Length     |  String ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type 33 for Proxy-State.

Length >= 3

String The String field is one or more octets. The actual format of the information is site or application specific, and a robust implementation SHOULD support the field as undistinguished octets.

The codification of the range of allowed usage of this field is outside the scope of this specification.

NCC Implementation Notes:

In 3GPP2 mode, RCA cannot deal with Proxy-State attribute.

In parameterized mode, RCA can deal with Proxy-State attributes to a limited extent. RCA can do one of the following:

- Copy a single proxy-State attribute of any type from the Access-Request to the response packet.
- Copy multiple Proxy-State attributes that are ASCII strings from the Access-Request to the response packet, provided each ASCII string starts with a unique string which distinguished that Proxy-State attribute from all the others. In this case, the order of the attributes id dependent on the order specified in configuration, and not on the order of the attributes in the Access-Request.

5.34. Login-LAT-Service

Description

This Attribute indicates the system with which the user is to be connected by LAT. It MAY be used in Access-Accept packets, but only when LAT is specified as the Login-Service. It MAY be used in an Access-Request packet as a hint to the server, but the server is not required to honor the hint.

Administrators use the service attribute when dealing with clustered systems, such as a VAX or Alpha cluster. In such an environment several different time sharing hosts share the same resources (disks, printers, etc.), and administrators often configure each to offer access (service) to each of the shared resources. In this case, each host in the cluster advertises its services through LAT broadcasts.

Sophisticated users often know which service providers (machines) are faster and tend to use a node name when initiating a LAT connection. Alternately, some administrators want particular users to use certain machines as a primitive form of load balancing (although LAT knows how to do load balancing itself)

A summary of the Login-LAT-Service Attribute format is shown below. The fields are transmitted from left to right.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Length     |  String ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type — 34 for Login-LAT-Service.

Length — >= 3

String — The String field is one or more octets, and contains the identity of the LAT service to use. The LAT Architecture allows this string to contain $ (dollar), - (hyphen), . (period), _ (underscore), numerics, upper and lower case alphabetics, and the ISO Latin-1 character set extension [11]. All LAT string comparisons are case insensitive.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.35. Login-LAT-Node

Description

This Attribute indicates the Node with which the user is to be automatically connected by LAT. It MAY be used in Access-Accept packets, but only when LAT is specified as the Login-Service. It MAY be used in an Access-Request packet as a hint to the server, but the server is not required to honor the hint.

A summary of the Login-LAT-Node Attribute format is shown below. The fields are transmitted from left to right.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Length     |  String ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type 35 for Login-LAT-Node.

Length >= 3

String The String field is one or more octets, and contains the identity of the LAT Node to connect the user to. The LAT Architecture allows this string to contain $ (dollar), - (hyphen), . (period), _ (underscore), numerics, upper and lower case alphabetics, and the ISO Latin-1 character set extension. All LAT string comparisons are case insensitive.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.36. Login-LAT-Group

Description

This Attribute contains a string identifying the LAT group codes which this user is authorized to use. It MAY be used in Access- Accept packets, but only when LAT is specified as the Login- Service. It MAY be used in an Access-Request packet as a hint to the server, but the server is not required to honor the hint.

LAT supports 256 different group codes, which LAT uses as a form of access rights. LAT encodes the group codes as a 256 bit bitmap.

Administrators can assign one or more of the group code bits at the LAT service provider; it will only accept LAT connections that have these group codes set in the bit map. The administrators assign a bitmap of authorized group codes to each user; LAT gets these from the operating system, and uses these in its requests to the service providers.

A summary of the Login-LAT-Group Attribute format is shown below. The fields are transmitted from left to right.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Length     |  String ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type 36 for Login-LAT-Group.

Length 34

String The String field is a 32 octet bit map, most significant octet first. A robust implementation SHOULD support the field as undistinguished octets.

The codification of the range of allowed usage of this field is outside the scope of this specification.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.37. Framed-AppleTalk-Link

Description

This Attribute indicates the AppleTalk network number which should be used for the serial link to the user, which is another AppleTalk router.   It is only used in Access-Accept packets.   It is never used when the user is not another router.

A summary of the Framed-AppleTalk-Link Attribute format is shown below.   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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	37 for Framed-AppleTalk-Link.
Length	6
String	The Value field is four octets. Despite the size of the field, values range from 0 to 65535. The special value of 0 indicates that this is an unnumbered serial link. A value of 1-65535 means that the serial line between the NAS and the user should be assigned that value as an AppleTalk network number.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.38. Framed-AppleTalk-Network

Description

This Attribute indicates the AppleTalk Network number which the NAS should probe to allocate an AppleTalk node for the user.   It is only used in Access-Accept packets.   It is never used when the user is another router.   Multiple instances of this Attribute indicate that the NAS may probe using any of the network numbers specified.

A summary of the Framed-AppleTalk-Network Attribute format is shown below.   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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	38 for Framed-AppleTalk-Network.
Length	6
Value	The Value field is four octets. Despite the size of the field, values range from 0 to 65535. The special value 0 indicates that the NAS should assign a network for the user, using its default cable range. A value between 1 and 65535 (inclusive) indicates the AppleTalk Network the NAS should probe to find an address for the user.

5.39. Framed-AppleTalk-Zone

Description

This Attribute indicates the AppleTalk Default Zone to be used for this user. It is only used in Access-Accept packets. Multiple instances of this attribute in the same packet are not allowed.

A summary of the Framed-AppleTalk-Zone Attribute format is shown below. The fields are transmitted from left to right.

```
 0                   1                   2
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Length     |  String ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type 39 for Framed-AppleTalk-Zone.

Length >= 3

String The name of the Default AppleTalk Zone to be used for this user. A robust implementation SHOULD support the field as undistinguished octets.

The codification of the range of allowed usage of this field is outside the scope of this specification.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.40. CHAP-Challenge

Description

This Attribute contains the CHAP Challenge sent by the NAS to a PPP Challenge-Handshake Authentication Protocol (CHAP) user. It is only used in Access-Request packets.

If the CHAP challenge value is 16 octets long it MAY be placed in the Request Authenticator field instead of using this attribute.

A summary of the CHAP-Challenge Attribute format is shown below. The fields are transmitted from left to right.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Length     |    String...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type 60 for CHAP-Challenge.

Length >= 7

String The String field contains the CHAP Challenge.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.41. NAS-Port-Type

Description

This Attribute indicates the type of the physical port of the NAS which is authenticating the user. It can be used instead of or in addition to the NAS-Port (5) attribute. It is only used in Access-Request packets. Either NAS-Port (5) or NAS-Port-Type or both SHOULD be present in an Access-Request packet, if the NAS differentiates among its ports.

A summary of the NAS-Port-Type Attribute format is shown below. 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type 61 for NAS-Port-Type.

Length 6

Value The Value field is four octets. "Virtual" refers to a connection to the NAS via some transport protocol, instead of through a physical port. For example, if a user telnetted into a NAS to authenticate himself as an Outbound-User, the Access-Request might include NAS-Port-Type = Virtual as a hint to the RADIUS server that the user was not on a physical port.

0	Async
1	Sync
2	ISDN Sync
3	ISDN Async V.120
4	ISDN Async V.110
5	Virtual
6	PIAFS
7	HDLC Clear Channel
8	X.25
9	X.75
10	G.3 Fax
11	SDSL - Symmetric DSL
12	ADSL-CAP - Asymmetric DSL, Carrierless Amplitude Phase Modulation
13	ADSL-DMT - Asymmetric DSL, Discrete Multi-Tone
14	IDSL - ISDN Digital Subscriber Line
15	Ethernet
16	xDSL - Digital Subscriber Line of unknown type
17	Cable
18	Wireless - Other

19 Wireless - IEEE 802.11

PIAFS is a form of wireless ISDN commonly used in Japan, and stands for PHS (Personal Handyphone System) Internet Access Forum Standard (PIAFS).

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.42. Port-Limit

Description

This Attribute sets the maximum number of ports to be provided to the user by the NAS. This Attribute MAY be sent by the server to the client in an Access-Accept packet. It is intended for use in conjunction with Multilink PPP [12] or similar uses. It MAY also be sent by the NAS to the server as a hint that that many ports are desired for use, but the server is not required to honor the hint.

A summary of the Port-Limit Attribute format is shown below. 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type 62 for Port-Limit.

Length 6

Value The field is 4 octets, containing a 32-bit unsigned integer with the maximum number of ports this user should be allowed to connect to on the NAS.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.43. Login-LAT-Port

Description

This Attribute indicates the Port with which the user is to be connected by LAT. It MAY be used in Access-Accept packets, but only when LAT is specified as the Login-Service. It MAY be used in an Access-Request packet as a hint to the server, but the server is not required to honor the hint.

A summary of the Login-LAT-Port Attribute format is shown below. The fields are transmitted from left to right.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Length     |  String ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type	63 for Login-LAT-Port.
Length	>= 3
String	The String field is one or more octets, and contains the identity of the LAT port to use. The LAT Architecture allows this string to contain $ (dollar), - (hyphen), . (period), _ (underscore), numerics, upper and lower case alphabetics, and the ISO Latin-1 character set extension. All LAT string comparisons are case insensitive.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

5.44. Table of Attributes

The following table provides a guide to which attributes may be found in which kinds of packets, and in what quantity.

Request	Accept	Reject	Challenge	#	Attribute
0-1	0-1	0	0 (not supported)	1	User-Name
0-1	0	0	0 (not supported)	2	User-Password [Note 1]
0-1	0	0	0 (not supported)	3	CHAP-Password [Note 1]
0-1	0	0	0 (not supported)	4	NAS-IP-Address [Note 2]
0-1	0	0	0 (not supported)	5	NAS-Port
0-1	0-1	0	0 (not supported)	6	Service-Type
0-1	0-1	0	0 (not supported)	7	Framed-Protocol
0-1	0-1	0	0 (not supported)	8	Framed-IP-Address
0-1	0-1	0	0 (not supported)	9	Framed-IP-Netmask
0	0-1	0	0 (not supported)	10	Framed-Routing
0	0+(not supported) 0-1 (non-standard implementation)	0	0 (not supported)	11	Filter-Id
0-1	0-1	0	0 (not supported)	12	Framed-MTU
0+(not supported) 0-1 (non-standard implementation)	0+(not supported) 0-1 (non-standard implementation)	0	0 (not supported)	13	Framed-Compression
0+(not supported) 0-1 (non-standard implementation)	0+(not supported) 0-1 (non-standard implementation)	0	0 (not supported)	14	Login-IP-Host
0	0-1	0	0 (not supported)	15	Login-Service

Request	Accept	Reject	Challenge	#	Attribute
0	0-1	0	0 (not supported)	16	Login-TCP-Port
0	0+(not supported) 0-1 (non-standard implementation)	0+	0+ (not supported)	18	Reply-Message
0-1	0-1	0	0 (not supported)	19	Callback-Number
0	0-1	0	0 (not supported)	20	Callback-Id
0	0+(not supported) 0-1 (non-standard implementation)	0	0 (not supported)	22	Framed-Route
0	0-1	0	0 (not supported)	23	Framed-IPX-Network
0-1	0-1	0	0-1 (not supported)	24	State [Note 1]
0	0+(not supported) 0-1 (non-standard implementation)	0	0 (not supported)	25	Class
0+(not supported) 0-1 (non-standard implementation)	0+(not supported) 0-1 (non-standard implementation)	0	0+ (not supported)	26	Vendor-Specific
0	0-1	0	0-1 (not supported)	27	Session-Timeout
0	0-1	0	0-1 (not supported)	28	Idle-Timeout
0	0-1	0	0 (not supported)	29	Termination-Action
0-1	0	0	0 (not supported)	30	Called-Station-Id
0-1	0	0	0 (not supported)	31	Calling-Station-Id
0-1	0	0	0 (not supported)	32	NAS-Identifier [Note 2]
0+(not supported) 0-1 (non-standard implementation)	0+(not supported) 0-1 (non-standard implementation)	0+	0+ (not supported)	33	Proxy-State
0-1	0-1	0	0 (not supported)	34	Login-LAT-Service
0-1	0-1	0	0 (not supported)	35	Login-LAT-Node
0-1	0-1	0	0 (not supported)	36	Login-LAT-Group
0	0-1	0	0 (not supported)	37	Framed-AppleTalk-Link

Request	Accept	Reject	Challenge	#	Attribute
0	0+(not supported) 0-1 (non-standard implementation)	0	0 (not supported)	38	Framed-AppleTalk-Network
0	0-1	0	0 (not supported)	39	Framed-AppleTalk-Zone
0-1	0	0	0 (not supported)	60	CHAP-Challenge
0-1	0	0	0 (not supported)	61	NAS-Port-Type
0-1	0-1	0	0 (not supported)	62	Port-Limit
0-1	0-1	0	0 (not supported)	63	Login-LAT-Port

NCC Implementation Notes:

The following is not implemented for RCA:

[Note 1] An Access-Request MUST contain either a User-Password or a CHAP-Password or State. An Access-Request MUST NOT contain both a User-Password and a CHAP-Password. If future extensions allow other kinds of authentication information to be conveyed, the attribute for that can be used in an Access-Request instead of User-Password or CHAP-Password.

[Note 2] An Access-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

This section provides guidance to the Internet Assigned Numbers Authority (IANA) regarding registration of values related to the RADIUS protocol, in accordance with BCP 26 [13].

There are three name spaces in RADIUS that require registration: Packet Type Codes, Attribute Types, and Attribute Values (for certain Attributes).

RADIUS is not intended as a general-purpose Network Access Server (NAS) management protocol, and allocations should not be made for purposes unrelated to Authentication, Authorization or Accounting.

6.1. Definition of Terms

The following terms are used here with the meanings defined in BCP 26:

- "name space",
- "assigned value",
- "registration".

The following policies are used here with the meanings defined in BCP 26:

- "Private Use",
- "First Come First Served",

- "Expert Review",
- "Specification Required",
- "IETF Consensus",
- "Standards Action".

6.2. Recommended Registration Policies

For registration requests where a Designated Expert should be consulted, the IESG Area Director for Operations should appoint the Designated Expert.

For registration requests requiring Expert Review, the ietf-radius mailing list should be consulted.

Packet Type Codes have a range from 1 to 254, of which 1-5,11-13 have been allocated.   Because a new Packet Type has considerable impact on interoperability, a new Packet Type Code requires Standards Action, and should be allocated starting at 14.

Attribute Types have a range from 1 to 255, and are the scarcest resource in RADIUS, thus must be allocated with care.   Attributes 1-53,55,60-88,90-91 have been allocated, with 17 and 21 available for re-use.   Attributes 17, 21, 54, 56-59, 89, 92-191 may be allocated following Expert Review, with Specification Required.   Release of blocks of Attribute Types (more than 3 at a time for a given purpose) should require IETF Consensus.   It is recommended that attributes 17 and 21 be used only after all others are exhausted.

Note that RADIUS defines a mechanism for Vendor-Specific extensions (Attribute 26) and the use of that should be encouraged instead of allocation of global attribute types, for functions specific only to one vendor's implementation of RADIUS, where no interoperability is deemed useful.

As stated in the "Attributes" section above:

"[Attribute Type] 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."

Therefore Attribute values 192-240 are considered Private Use, and values 241-255 require Standards Action.

Certain attributes (for example, NAS-Port-Type) in RADIUS define a list of values to correspond with various meanings.   There can be 4 billion (2^32) values for each attribute. Adding additional values to the list can be done on a First Come, First Served basis by the IANA.

7. Examples

A few examples are presented to illustrate the flow of packets and use of typical attributes.   These examples are not intended to be exhaustive, many others are possible.   Hexadecimal dumps of the example packets are given in network byte order, using the shared secret "xyzzy5461".

7.1. User Telnet to Specified Host

The NAS at 192.168.1.16 sends an Access-Request UDP packet to the RADIUS Server for a user named nemo logging in on port 3 with password "arctangent".

The Request Authenticator is a 16 octet random number generated by the NAS.

The User-Password is 16 octets of password padded at end with nulls, XORed with MD5(shared secret|Request Authenticator).

```
01 00 00 38 0f 40 3f 94 73 97 80 57 bd 83 d5 cb
98 f4 22 7a 01 06 6e 65 6d 6f 02 12 0d be 70 8d
93 d4 13 ce 31 96 e4 3f 78 2a 0a ee 04 06 c0 a8
01 10 05 06 00 00 00 03
```

1 Code = Access-Request (1)

1 ID = 0

2 Length = 56

16 Request Authenticator

Attributes:

6 User-Name = "nemo"

18 User-Password

6 NAS-IP-Address = 192.168.1.16

6 NAS-Port = 3

The RADIUS server authenticates nemo, and sends an Access-Accept UDP packet to the NAS telling it to telnet nemo to host 192.168.1.3.

NCC Implementation Notes:

RCA does not check the authenticator against a shared secret. It does perform credit control and sends an Access_accept if the subscriber has sufficient credit.

The Response Authenticator is a 16-octet MD5 checksum of the code (2), id (0), Length (38), the Request Authenticator from above, the attributes in this reply, and the shared secret.

```
02 00 00 26 86 fe 22 0e 76 24 ba 2a 10 05 f6 bf
9b 55 e0 b2 06 06 00 00 00 01 0f 06 00 00 00 00
0e 06 c0 a8 01 03
```

1 Code = Access-Accept (2)

1 ID = 0 (same as in Access-Request)

2 Length = 38

16 Response Authenticator

Attributes:

6 Service-Type (6) = Login (1)

6 Login-Service (15) = Telnet (0)

6 Login-IP-Host (14) = 192.168.1.3

NCC Implementation Notes:

RCA can send an Access-Accept with these parameters but only in parameterized mode.

7.2. Framed User Authenticating with CHAP

The NAS at 192.168.1.16 sends an Access-Request UDP packet to the RADIUS Server for a user named flopsy logging in on port 20 with PPP, authenticating using CHAP. The NAS sends along the Service-Type and Framed-Protocol attributes as a hint to the RADIUS server that this user is looking for PPP, although the NAS is not required to do so.

The Request Authenticator is a 16 octet random number generated by the NAS, and is also used as the CHAP Challenge.

The CHAP-Password consists of a 1 octet CHAP ID, in this case 22, followed by the 16 octet CHAP response.

```
01 01 00 47 2a ee 86 f0 8d 0d 55 96 9c a5 97 8e
0d 33 67 a2 01 08 66 6c 6f 70 73 79 03 13 16 e9
75 57 c3 16 18 58 95 f2 93 ff 63 44 07 72 75 04
06 c0 a8 01 10 05 06 00 00 00 14 06 06 00 00 00
02 07 06 00 00 00 01
```

1 Code = 1 (Access-Request)

1 ID = 1

2 Length = 71

16 Request Authenticator

Attributes:

8 8 User-Name (1) = "flopsy"

19 CHAP-Password (3)

6 NAS-IP-Address (4) = 192.168.1.16

6 NAS-Port (5) = 20

6 Service-Type (6) = Framed (2)

6 Framed-Protocol (7) = PPP (1)

The RADIUS server authenticates flopsy, and sends an Access-Accept UDP packet to the NAS telling it to start PPP service and assign an address for the user out of its dynamic address pool.

NCC Implementation Notes:

RCA does not check the CHAP-password. It does perform credit control and send an Access_accept if the subscriber has sufficient credit.

The Response Authenticator is a 16-octet MD5 checksum of the code (2), id (1), Length (56), the Request Authenticator from above, the attributes in this reply, and the shared secret.

```
02 01 00 38 15 ef bc 7d ab 26 cf a3 dc 34 d9 c0
3c 86 01 a4 06 06 00 00 00 02 07 06 00 00 00 01
08 06 ff ff ff fe 0a 06 00 00 00 02 0d 06 00 00
00 01 0c 06 00 00 05 dc
```

1 Code = Access-Accept (2)

1 ID = 0 (same as in Access-Request)

2 Length = 56

16 Response Authenticator

Attributes:

6 Service-Type (6) = Framed (2)

6 Framed-Protocol (7) = PPP (1)

6 Framed-IP-Address (8) = 255.255.255.254

6 Framed-Routing (10) = None (0)

6 Framed-Compression (13) = VJ TCP/IP Header Compression (1)

6 Framed-MTU (12) = 1500

NCC Implementation Notes:

RCA can send an Access-Accept with these parameters but only in parameterised mode.

7.3. User with Challenge-Response card

NCC Implementation Notes:

The RCA does not support the following:

The NAS at 192.168.1.16 sends an Access-Request UDP packet to the RADIUS Server for a user named mopsy logging in on port 7. The user enters the dummy password "challenge" in this example. The challenge and response generated by the smart card for this example are "32769430" and "99101462".

The Request Authenticator is a 16 octet random number generated by the NAS.

The User-Password is 16 octets of password, in this case "challenge", padded at the end with nulls, XORed with MD5(shared secret|Request Authenticator).

```
01 02 00 39 f3 a4 7a 1f 6a 6d 76 71 0b 94 7a b9
30 41 a0 39 01 07 6d 6f 70 73 79 02 12 33 65 75
73 77 82 89 b5 70 88 5e 15 08 48 25 c5 04 06 c0
a8 01 10 05 06 00 00 00 07
```

1 Code = Access-Request (1)
1 ID = 2
2 Length = 57
16 Request Authenticator

Attributes:

7 User-Name (1) = "mopsy"
18 User-Password (2)
6 NAS-IP-Address (4) = 192.168.1.16
6 NAS-Port (5) = 7

The RADIUS server decides to challenge mopsy, sending back a challenge string and looking for a response. The RADIUS server therefore and sends an Access-Challenge UDP packet to the NAS.

The Response Authenticator is a 16-octet MD5 checksum of the code (11), id (2), length (78), the Request Authenticator from above, the attributes in this reply, and the shared secret.

The Reply-Message is "Challenge 32769430. Enter response at prompt."

The State is a magic cookie to be returned along with user's response; in this example 8 octets of data (33 32 37 36 39 34 33 30 in hex).

```
0b 02 00 4e 36 f3 c8 76 4a e8 c7 11 57 40 3c 0c
71 ff 9c 45 12 30 43 68 61 6c 6c 65 6e 67 65 20
33 32 37 36 39 34 33 30 2e 20 20 45 6e 74 65 72
20 72 65 73 70 6f 6e 73 65 20 61 74 20 70 72 6f
6d 70 74 2e 18 0a 33 32 37 36 39 34 33 30
```

1 Code = Access-Challenge (11)
1 ID = 2 (same as in Access-Request)
2 Length = 78
16 Response Authenticator

Attributes:

48 Reply-Message (18)
10 State (24)

The user enters his response, and the NAS send a new Access-Request with that response, and includes the State Attribute.

The Request Authenticator is a new 16 octet random number.

The User-Password is 16 octets of the user's response, in this case "99101462", padded at the end with nulls, XORed with MD5(shared secret|Request Authenticator).

The state is the magic cookie from the Access-Challenge packet, unchanged.

```
01 03 00 43 b1 22 55 6d 42 8a 13 d0 d6 25 38 07
c4 57 ec f0 01 07 6d 6f 70 73 79 02 12 69 2c 1f
20 5f c0 81 b9 19 b9 51 95 f5 61 a5 81 04 06 c0
a8 01 10 05 06 00 00 00 07 18 10 33 32 37 36 39
34 33 30
```

1 Code = Access-Request (1)
1 ID = 3 (Note that this changes.)
2 Length = 67
16 Request Authenticator

Attributes:

7 User-Name = "mopsy"
18 User-Password
6 NAS-IP-Address (4) = 192.168.1.16
6 NAS-Port (5) = 7
10 State (24)

The Response was incorrect (for the sake of example), so the RADIUS server tells the NAS to reject the login attempt.

The Response Authenticator is a 16 octet MD5 checksum of the code (3), id (3), length(20), the Request Authenticator from above, the attributes in this reply (in this case, none), and the shared secret.

```
03 03 00 14 a4 2f 4f ca 45 91 6c 4e 09 c8 34 0f
9e 74 6a a0
```

1 Code = Access-Reject (3)
1 ID = 3 (same as in Access-Request)
2 Length = 20
16 Response Authenticator

Attributes:

(none, although a Reply-Message could be sent)

8. Security Considerations

Security issues are the primary topic of this document.

NCC Implementation Notes:

The RCA does not support the following:

- In practice, within or associated with each RADIUS server, there is a database which associates "user" names with authentication information ("secrets").

NCC Implementation Notes:

RCA does not check passwords or the shared secret.

It is not anticipated that a particular named user would be authenticated by multiple methods. This would make the user vulnerable to attacks which negotiate the least secure method from among a set. Instead, for each named user there should be an indication of exactly one method used to authenticate that user name. If a user needs to make use of different authentication methods under different circumstances, then distinct user names SHOULD be employed, each of which identifies exactly one authentication method.

Passwords and other secrets should be stored at the respective ends such that access to them is as limited as possible. Ideally, the secrets should only be accessible to the process requiring access in order to perform the authentication.

The secrets should be distributed with a mechanism that limits the number of entities that handle (and thus gain knowledge of) the secret. Ideally, no unauthorized person should ever gain knowledge of the secrets. It is possible to achieve this with SNMP Security Protocols [14], but such a mechanism is outside the scope of this specification.

Other distribution methods are currently undergoing research and experimentation. The SNMP Security document [14] also has an excellent overview of threats to network protocols.

The User-Password hiding mechanism described in Section 5.2 has not been subjected to significant amounts of cryptanalysis in the published literature. Some in the IETF community are concerned that this method might not provide sufficient confidentiality protection [15] to passwords transmitted using RADIUS. Users should evaluate their threat environment and consider whether additional security mechanisms should be employed.

9. Change Log

The following changes have been made from RFC 2138:

Strings should use UTF-8 instead of US-ASCII and should be handled as 8-bit data.

Integers and dates are now defined as 32 bit unsigned values.

Updated list of attributes that can be included in Access-Challenge to be consistent with the table of attributes.

User-Name mentions Network Access Identifiers.

User-Name may now be sent in Access-Accept for use with accounting and Rlogin.

Values added for Service-Type, Login-Service, Framed-Protocol, Framed-Compression, and NAS-Port-Type.

NAS-Port can now use all 32 bits.

Examples now include hexadecimal displays of the packets.

Source UDP port must be used in conjunction with the Request Identifier when identifying duplicates.

Multiple subattributes may be allowed in a Vendor-Specific attribute.

An Access-Request is now required to contain either a NAS-IP-Address or NAS-Identifier (or may contain both).

Added notes under "Operations" with more information on proxy, retransmissions, and keep-alives.

If multiple Attributes with the same Type are present, the order of Attributes with the same Type MUST be preserved by any proxies.

Clarified Proxy-State.

Clarified that Attributes must not depend on position within the packet, as long as Attributes of the same type are kept in order.

Added IANA Considerations section.

Updated section on "Proxy" under "Operations".

Framed-MTU can now be sent in Access-Request as a hint.

Updated Security Considerations.

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

10. References

[1] Rigney, C., Rubens, A., Simpson, W. and S. Willens, "Remote Authentication Dial In User Service (RADIUS)", RFC 2138, April 1997.

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

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

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

[5] Rigney, C., "RADIUS Accounting", RFC 2866, June 2000.

[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] Aboba, B. and M. Beadles, "The Network Access Identifier", RFC 2486, January 1999.

[9] Kaufman, C., Perlman, R., and Speciner, M., "Network Security: Private Communications in a Public World", Prentice Hall, March 1995, ISBN 0-13-061466-1.

[10] Jacobson, V., "Compressing TCP/IP headers for low-speed serial links", RFC 1144, February 1990.

[11] ISO 8859. International Standard -- Information Processing -- 8-bit Single-Byte Coded Graphic Character Sets -- Part 1: Latin Alphabet No. 1, ISO 8859-1:1987.

[12] Sklower, K., Lloyd, B., McGregor, G., Carr, D. and T. Coradetti, "The PPP Multilink Protocol (MP)", RFC 1990, August 1996.

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

[14] Galvin, J., McCloghrie, K. and J. Davin, "SNMP Security Protocols", RFC 1352, July 1992.

15] Dobbertin, H., "The Status of MD5 After a Recent Attack", CryptoBytes Vol.2 No.2, Summer 1996.

11. Acknowledgements

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

12. Chair's Address

The 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

13. Authors' Addresses

Questions about this memo can also be directed to:

Carl Rigney
Livingston Enterprises
4464 Willow Road
Pleasanton, California 94588

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

Allan C. Rubens
Merit Network, Inc.
4251 Plymouth Road
Ann Arbor, Michigan 48105-2785

EMail: acr@merit.edu

William Allen Simpson
Daydreamer
Computer Systems Consulting Services
1384 Fontaine
Madison Heights, Michigan 48071

EMail: wsimpson@greendragon.com

Steve Willens
Livingston Enterprises
4464 Willow Road
Pleasanton, California 94588

EMail: steve@livingston.com

14. Full Copyright Statement

Acknowledgement

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

Chapter 2

Compliance to RFC 2866 (RADIUS Accounting)

Overview

Introduction

This chapter identifies the compliance of Oracle to RFC 2866 - Remote Authentication Dial In User Service (RADIUS) - Accounting protocol.

In this chapter

This chapter contains the following topics.

RADIUS Accounting

RADIUS Accounting

Network Working Group	C. Rigney
Request for Comments: 2866	Livingston
Category: Informational	June 2000
Obsoletes: 2139	

RADIUS Accounting

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

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.

NCC Implementation Notes:

In RCA, the default port for core Accounting messages is 1813 but any other port number can be configured instead.

Table of Contents

NCC Implementation Notes:

RCA can handle Accounting-Request messages in 3GPP mode but all it does is send Accounting-Response and increase a statistics counter. It does not use the Accounting-request information for credit control.

In parameterized mode, RCA can be configured to use the information from the Accounting-Request for credit control. It can also be configured to not send Accounting-response messages on a per client type basis. (This is non-standard behaviour but some clients do not support Accounting-Response.)

1. Introduction

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, authorisation 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 [17] specifies the RADIUS protocol used for Authentication and Authorisation. 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.

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.

NCC Implementation Notes:

RCA does not validate the sending client – it accepts clients with any shared secret. If such validation is done it must be done by a proxy.

- 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 capitalised or not.

1.2. Terminology

This document frequently 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 & 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

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.

NCC Implementation Notes:

RCA does not support the following:

- 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

NCC Implementation Notes:

RCA does not act as a RADIUS 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

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).

NCC Implementation Notes:

In RCA, the default port for core Accounting messages is 1813 but any other port number can be configured instead.

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.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         Authenticator                         |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Attributes ...
+-+-+-+-+-+-+-+-+-+-+-+-+-
```

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 [3] 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.

NCC Implementation Notes:

RCA does not validate the authenticator.

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.

NCC Implementation Notes:

- RCA cannot cope with receiving multiple attributes of the same type, except Vendor-Specific attributes which it can cope with.
- In 3GPP2 mode, RCA will never send a Radius message with more than one attribute of the same type, except for Vendor-Specific attributes.
- In parameterised mode, RCA can be configured to send more than one attribute of the same type, for any type.

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.

NCC Implementation Notes:

RCA does not support the following on receipt of an Accounting-Request:

"and MUST NOT transmit any reply if it fails to record the accounting packet."

NCC Implementation Notes:

- In 3GPP mode, RCA can handle Accounting-Request messages but all it does is send Accounting-Response and increase a statistics counter. It does not use the Accounting-request information for credit control.
- In parameterized mode, RCA can be configured to use the information from the Accounting-Request for credit control. It can also be configured to not send Accounting-response messages on a per client type basis. (This is non-standard behaviour but some clients do not support Accounting-Response.)
- In both modes, whether RCA sends an Accounting-Response is not affected by whether RCA has successfully recorded the information.

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.

NCC Implementation Notes:

RCA will not raise errors if these attributes are present in an Accounting-Request. In 3GPP mode, it will ignore any parameters it does not expect. in parameterized mode, it will decode any parameters it is configured to decode (and no others) whether or not they are listed above.

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.

NCC Implementation Notes:

RCA does not check that the Accounting-Request conforms to the above paragraph.

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.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                     Request Authenticator                     |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Attributes ...
+-+-+-+-+-+-+-+-+-+-+-+-+-
```

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.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                     Response Authenticator                    |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Attributes ...
+-+-+-+-+-+-+-+-+-+-+-+-+-
```

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.

Request 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

5. Attributes

NCC Implementation Notes:

- In 3GPP2 mode, RCA can only cope with a fixed set of attributes. I.e. in 3GPP2 mode RCA deals with all the attributes listed in this document unless this document explicitly states that the attribute is not used in 3GPP2 mode.
- In parameterized mode, RCA can deal with any core Radius attribute or vendor specific attribute (whether or not they are listed in this document) provided that: - It is (or can be treated as) of type text, octets, IPv4 address or number - The attribute is at the top level. (I.e. RCA in parameterised mode cannot cope with attributes within attributes.

In the above note, the phrase “RCA can deal with [an attribute]” means that, in parameterised mode, RCA can use the attribute in decision making concerning credit control, in a configurable way, and / or store the attribute to be sent out again in Radius messages and / or send the attribute in outgoing messages on a per message type basis.

RADIUS Attributes carry the specific authentication, authorisation 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.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |  Value ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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. **NCC Implementation Note:** RCA does not support the following: • 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.

NCC Implementation Notes:

RCA cannot cope with embedded nulls in text type attributes.

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

NCC Implementation Notes:

- In 3GPP mode, RCA ignores this attribute.
- In parameterized mode, RCA can be configured to use this attribute to determine whether the session is still active. Also, RCA can be configured to set this attribute to STOP in the Accounting-Response to indicate to the client that the session should be stopped. (This is non-standard behaviour and RCA, by default, does not do this.)

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.

```
 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	40 for Acct-Status-Type.	
Length	6	
Value	The Value field is four octets.	
	1	Start
	2	Stop
	3	Interim-Update
NCC Implementation Note: Unsupported by RCA	7 8 9-14 15	Accounting-On Accounting-Off Reserved for Tunnel Accounting Reserved for Failed

5.2. Acct-Delay-Time

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

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.

```
 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	41 for Acct-Delay-Time.
Length	6
Value	The Value field is four octets.

5.3. Acct-Input-Octets

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

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.

```
 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	42 for Acct-Input-Octets.
Length	6
Value	The Value field is four octets.

5.4. Acct-Output-Octets

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.   It will be common to configure RCA to use this parameter to determine, in part, how many units to charge for when performing credit control.

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.

```
 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	43 for Acct-Output-Octets.
Length	6
Value	The Value field is four octets.

5.5. Acct-Session-Id

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

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.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |  Text ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

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.

```
 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis. It will be common to configure RCA to use this parameter to determine, in part, how many units to charge for when performing credit control.

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.

```
 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	46 for Acct-Session-Time.
Length	6
Value	The Value field is four octets.

5.8. Acct-Input-Packets

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis. It will be common to configure RCA to use this parameter to determine, in part, how many units to charge for when performing credit control.

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.

```
 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	47 for Acct-Input-Packets.
Length	6
Value	The Value field is four octets.

5.9. Acct-Output-Packets

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis. It will be common to configure RCA to use this parameter to determine, in part, how many units to charge for when performing credit control.

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.

```
 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	48 for Acct-Output-Packets.
Length	6
Value	The Value field is four octets.

5.11. Acct-Multi-Session-Id

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

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.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |  String ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

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.

```
 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          Value (cont)          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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.

NCC Implementation Notes:

- In 3GPP2 mode, RCA does not put any attributes in Accounting-Response.
- In parameterised mode, RCA can be configured to comply with the above sentence. Alernatively, RCA can be configured to send any attribute in Accounting-Response.

#	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

#	Attribute
0-1	Framed-IP-Address
0-1	Framed-IP-Netmask
0-1	Framed-Routing
0+(not supported) 0-1 (non-standard implementation)	Filter-Id
0-1	Framed-MTU
0+(not supported) 0-1	Framed-Compression
0+(not supported) 0-1 (non-standard implementation)	Login-IP-Host
0-1	Login-Service
0-1	Login-TCP-Port
0	Reply-Message
0-1	Callback-Number
0-1	Callback-Id
0+(not supported) 0-1 (non-standard implementation)	Framed-Route
0-1	Framed-IPX-Network
0	State
0+(not supported) 0-1 (non-standard implementation)	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+(not supported) 0-1 (non-standard implementation)	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

#	Attribute
0-1	Acct-Status-Type
0-1	Acct-Delay-Time
0-1	Acct-Input-Octets
0-1	Acct-Output-Octets
0-1	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+(not supported) 0-1 (non-standard implementation)	Acct-Multi-Session-Id
0+(not supported) 0-1 (non-standard implementation)	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 [17], in accordance with BCP 26 [13].

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 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

5.10. Acct-Terminate-Cause

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

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.

```
 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          Value (cont)          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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:

Cause	Description
User Request	User requested termination of service, for example with LCP Terminate or by logging out.
Lost Carrirer	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.

13. Full Copyright Statement

Acknowledgement

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

Chapter 3

Compliance to RFC 2869 (RADIUS Extensions)

Overview

Introduction

This chapter describes additional attributes for carrying authentication, authorization and accounting information between a Network Access Server (NAS) and a shared Accounting Server using the Remote Authentication Dial In User Service (RADIUS) protocol described in RFC 2865 [17] and RFC 2866 [5].

In this chapter

This chapter contains the following topics.

RADIUS Extensions

RADIUS Extensions

Network Working Group
Request for Comments: 2869
Category: Informational

C. Rigney
Livingston
W. Willats
Cyno Technologies
P. Calhoun
Sun Microsystems

June 2000

RADIUS Extensions

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This document describes additional attributes for carrying authentication, authorization and accounting information between a Network Access Server (NAS) and a shared Accounting Server using the Remote Authentication Dial In User Service (RADIUS) protocol described in RFC 2865 [1] and RFC 2866 [2].

Table of Contents

1. Introduction

RFC 2865 [1] describes the RADIUS Protocol as it is implemented and deployed today, and RFC 2866 [2] describes how Accounting can be performed with RADIUS.

This memo suggests several additional Attributes that can be added to RADIUS to perform various useful functions.   These Attributes do not have extensive field experience yet and should therefore be considered experimental.

The Extensible Authentication Protocol (EAP) [3] is a PPP extension that provides support for additional authentication methods within PPP.   This memo describes how the EAP-Message and Message-Authenticator attributes may be used for providing EAP support within RADIUS.

All attributes are comprised of variable length Type-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 [4].

An implementation is not compliant if it fails to satisfy one or more of the must or must not requirements for the protocols it implements.

An implementation that satisfies all the must, must not, should and should not requirements for its protocols is said to be "unconditionally compliant"; one that satisfies all the must and must not requirements but not all the should or should not requirements for its protocols is said to be "conditionally compliant."

A NAS that does not implement a given service MUST NOT implement the RADIUS attributes for that service.   For example, a NAS that is unable to offer ARAP service MUST NOT implement the RADIUS attributes for ARAP.   A NAS MUST treat a RADIUS access-request requesting an unavailable service as an access-reject instead.

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   series if the NAS supports that, with each session generating a separate start and stop accounting record.

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

Operation is identical to that defined in RFC 2865 [1] and RFC 2866 [2].

2.1. RADIUS support for Interim Accounting Updates

When a user is authenticated, a RADIUS server issues an Access-Accept in response to a successful Access-Request. If the server wishes to receive interim accounting messages for the given user it must include the Acct-Interim-Interval RADIUS attribute in the message, which indicates the interval in seconds between interim messages.

NCC Implementation Notes:

- In 3GPP2 mode, RCA does not send this attribute.
- In parameterised mode, RCA can be configured to send or not send this attribute in an Access-Accept message.

It is also possible to statically configure an interim value on the NAS itself. Note that a locally configured value on the NAS MUST override the value found in an Access-Accept.

This scheme does not break backward interoperability since a RADIUS server not supporting this extension will simply not add the new Attribute. NASes not supporting this extension will ignore the Attribute.

Note that all information in an interim message is cumulative (i.e. number of packets sent is the total since the beginning of the session, not since the last interim message).

NCC Implementation Notes:

- In 3GPP mode, RCA does noting with the information from Accounting-request messages.
- In parameterised mode, RCA treats the information as cumulative, as specified above, but can also be configured to treat the values as "total since last interim message" in case the client is implemented in a non-standard way.

It is envisioned that an Interim Accounting record (with Acct-Status-Type = Interim-Update (3)) would contain all of the attributes normally found in an Accounting Stop message with the exception of the Acct-Term-Cause attribute.

NCC Implementation Notes:

RCA does not produce accounting records as such. rather, it uses the Accounting-Request messages for credit control.

Since all the information is cumulative, a NAS MUST ensure that only a single generation of an interim Accounting message for a given session is present in the retransmission queue at any given time.

A NAS MAY use a fudge factor to add a random delay between Interim Accounting messages for separate sessions. This will ensure that a cycle where all messages are sent at once is prevented, such as might otherwise occur if a primary link was recently restored and many dial-up users were directed to the same NAS at once.

The Network and NAS CPU load of using Interim Updates should be carefully considered, and appropriate values of Acct-Interim-Interval chosen.

2.2. RADIUS support for Apple Remote Access Protocol

NCC Implementation Notes:

RCA does not support the following:

The RADIUS (Remote Authentication Dial-In User Service) protocol provides a method that allows multiple dial-in Network Access Server (NAS) devices to share a common authentication database.

The Apple Remote Access Protocol (ARAP) provides a method for sending AppleTalk network traffic over point-to-point links, typically, but not exclusively, asynchronous and ISDN switched-circuit connections.

Though Apple is moving toward ATCP on PPP for future remote access services, ARAP is still a common way for the installed base of Macintosh users to make remote network connections, and is likely to remain so for some time.

ARAP is supported by several NAS vendors who also support PPP, IPX and other protocols in the same NAS. ARAP connections in these multi-protocol devices are often not authenticated with RADIUS, or if they are, each vendor creates an individual solution to the problem.

This section describes the use of additional RADIUS attributes to support ARAP. RADIUS client and server implementations that implement this specification should be able to authenticate ARAP connections in an interoperable manner.

This section assumes prior knowledge of RADIUS, and will go into some detail on the operation of ARAP before entering a detailed discussion of the proposed ARAP RADIUS attributes.

There are two features of ARAP this document does not address:

1 User initiated password changing. This is not part of RADIUS, but can be implemented through a software process other than RADIUS.

2 Out-of-Band messages. At any time, the NAS can send messages to an ARA client which appear in a dialog box on the dial-in user's screen. These are not part of authentication and do not belong here. However, we note that a Reply-Message attribute in an Access-Accept may be sent down to the user as a sign-on message of the day string using the out-of-band channel.

We have tried to respect the spirit of the existing RADIUS protocol as much as possible, making design decisions compatible with prior art. Further, we have tried to strike a balance between flooding the RADIUS world with new attributes, and hiding all of ARAP operation within a single multiplexed ARAP attribute string or within Extended Authentication Protocol (EAP) [3] machinery.

However, we feel ARAP is enough of a departure from PPP to warrant a small set of similarly named attributes of its own.

We have assumed that an ARAP-aware RADIUS server will be able to do DES encryption and generate security module challenges. This is in keeping with the general RADIUS goal of smart server / simple NAS.

ARAP authenticates a connection in two phases. The first is a "Two-Way DES" random number exchange, using the user's password as a key.

We say "Two-Way" because the ARAP NAS challenges the dial-in client to authenticate itself, and the dial-in client challenges the ARAP NAS to authenticate itself.

Specifically, ARAP does the following:

1 The NAS sends two 32-bit random numbers to the dial-in client in an ARAP msg_auth_challenge packet.

2 The dial-in client uses the user's password to DES encrypt the two random numbers sent to it by the NAS. The dial-in client then sends this result, the user's name and two 32-bit random numbers of its own back to the NAS in an ARAP msg_auth_request packet.

3 The NAS verifies the encrypted random numbers sent by the dial-in client are what it expected. If so, it encrypts the dial-in client's challenge using the password and sends it back to the dial-in client in an ARAP msg_auth_response packet.

Note that if the dial-in client's response was wrong, meaning the user has the wrong password, the server can initiate a retry sequence up to the maximum amount of retries allowed by the NAS. In this case, when the dial-in client receives the ARAP msg_auth_response packet it will acknowledge it with an ARAP msg_auth_again packet.

After this first "DES Phase" the ARAP NAS MAY initiate a secondary authentication phase using what Apple calls "Add-In Security Modules." Security Modules are small pieces of code which run on both the client and server and are allowed to read and write arbitrary data across the communications link to perform additional authentication functions. Various security token vendors use this mechanism to authenticate ARA callers.

Although ARAP allows security modules to read and write anything they like, all existing security modules use simple challenge and response cycles, with perhaps some overall control information. This document assumes all existing security modules can be supported with one or more challenge/response cycles.

To complicate RADIUS and ARAP integration, ARAP sends down some profile information after the DES Phase and before the Security Module phase. This means that besides the responses to challenges, this profile information must also be present, at somewhat unusual times. Fortunately the information is only a few pieces of numeric data related to passwords, which this document packs into a single new attribute.

Presenting an Access-Request to RADIUS on behalf of an ARAP connection is straightforward. The ARAP NAS generates the random number challenge, and then receives the dial-in client's response, the dial-in client's challenge, and the user's name. Assuming the user is not a guest, the following information is forwarded in an Access-Request packet: User-Name (up to 31 characters long),

Framed-Protocol (set to 3, ARAP), ARAP-Password, and any additional attributes desired, such as Service-Type, NAS-IP-Address, NAS-Id, NAS-Port-Type, NAS-Port, NAS-Port-Id, Connect-Info, etc.

The Request Authenticator is a NAS-generated 16 octet random number.

The low-order 8 octets of this number are sent to the dial-in user as the two 4 octet random numbers required in the ARAP msg_auth_challenge packet. Octets 0-3 are the first random number and Octets 4-7 are the second random number.

The ARAP-Password in the Access-Request contains a 16 octet random number field, and is used to carry the dial-in user's response to the NAS challenge and the client's own challenge to the NAS. The high-order octets contain the dial-in user's challenge to the NAS (2 32-bit numbers, 8 octets) and the low-order octets contain the dial-in user's response to the NAS challenge (2 32-bit numbers, 8 octets).

Only one of User-Password, CHAP-Password, or ARAP-Password needs to be present in an Access-Request, or one or more EAP-Messages.

If the RADIUS server does not support ARAP it SHOULD return an Access-Reject to the NAS.

If the RADIUS server does support ARAP, it should verify the user's response using the Challenge (from the lower order 8 octets of the Request Authenticator) and the user's response (froM the low order 8 octets of the ARAP-Password).

If that authentication fails, the RADIUS server should return an Access-Reject packet to the NAS, with optional Password-Retry and Reply-Messages attributes. The presence of Password-Retry indicates the ARAP NAS MAY choose to initiate another challenge-response cycle, up to a total number of times equal to the integer value of the Password-Retry attribute.

If the user is authenticated, the RADIUS server should return an Access-Accept packet (Code 2) to the NAS, with ID and Response Authenticator as usual, and attributes as follows:

Service-Type of Framed-Protocol.

Framed-Protocol of ARAP (3).

Session-Timeout with the maximum connect time for the user in seconds. If the user is to be given unlimited time, Session-Timeout should not be included in the Access-Accept packet, and ARAP will treat that as an unlimited timeout (-1).

ARAP-Challenge-Response, containing 8 octets with the response to the dial-in client's challenge. The RADIUS server calculates this value by taking the dial-in client's challenge from the high order 8 octets of the ARAP-Password attribute and performing DES encryption on this value with the authenticating user's password as the key. If the user's password is less than 8 octets in length, the password is padded at the end with NULL octets to a length of 8 before using it as a key. If the user's password is greater than 8 octets in length, an Access-Reject MUST be sent instead.

ARAP-Features, containing information that the NAS should send to the user in an ARAP "feature flags" packet.

Octet 0: If zero, user cannot change their password. If non-zero user can. (RADIUS does not handle the password changing, just the attribute which indicates whether ARAP indicates they can.)

Octet 1: Minimum acceptable password length (0-8).

Octet 2-5: Password creation date in Macintosh format, defined as 32 bits unsigned representing seconds since Midnight GMT January 1, 1904.

Octet 6-9 Password Expiration Delta from create date in seconds.

Octet 10-13: Current RADIUS time in Macintosh format

Optionally, a single Reply-Message with a text string up to 253 characters long which MAY be sent down to the user to be displayed in a sign-on/message of the day dialog.

Framed-AppleTalk-Network may be included.

Framed-AppleTalk-Zone, up to 32 characters in length, may be included.

ARAP defines the notion of a list of zones for a user. Along with a list of zone names, a Zone Access Flag is defined (and used by the NAS) which says how to use the list of zone names. That is, the dial-in user may only be allowed to see the Default Zone, or only the zones in the zone list (inclusive) or any zone except those in the zone list (exclusive).

The ARAP NAS handles this by having a named filter which contains (at least) zone names. This solves the problem where a single RADIUS server is managing disparate NAS clients who may not be able to "see" all of the zone names in a user zone list. Zone names only have meaning "at the NAS." The disadvantage of this approach is that zone filters must be set up on the NAS somehow, then referenced by the RADIUS Filter-Id.

ARAP-Zone-Access contains an integer which specifies how the "zone list" for this user should be used. If this attribute is present and the value is 2 or 4 then a Filter-Id must also be present to name a zone list filter to apply the access flag to.

The inclusion of a Callback-Number or Callback-Id attribute in the Access-Accept MAY cause the ARAP NAS to disconnect after sending the Feature Flags to begin callback processing in an ARAP specific way.

Other attributes may be present in the Access-Accept packet as well.

An ARAP NAS will need other information to finish bringing up the connection to the dial in client, but this information can be provided by the ARAP NAS without any help from RADIUS, either through configuration by SNMP, a NAS administration program, or deduced by the AppleTalk stack in the NAS. Specifically:

1 AppearAsNet and AppearAsNode values, sent to the client to tell it what network and node numbers it should use in its datagram packets. AppearAsNet can be taken from the Framed-AppleTalk-Network attribute or from the configuration or AppleTalk stack onthe NAS.

2 The "default" zone - that is the name of the AppleTalk zone in which the dial-in client will appear. (Or can be specified with the Framed-AppleTalk-Zone attribute.)

3 Other very NAS specific stuff such as the name of the NAS, and smartbuffering information. (Smartbuffering is an ARAP mechanism for replacing common AppleTalk datagrams with small tokens, to improve slow link performance in a few common traffic situations.)

4 "Zone List" information for this user. The ARAP specification defines a "zone count" field which is actually unused.

RADIUS supports ARAP Security Modules in the following manner.

After DES authentication has been completed, the RADIUS server may instruct the ARAP NAS to run one or more security modules for the dial-in user. Although the underlying protocol supports executing multiple security modules in series, in practice all current implementations only allow executing one. Through the use of multiple Access-Challenge requests, multiple modules can be supported, but this facility will probably never be used.

We also assume that, even though ARAP allows a free-form dialog between security modules on each end of the point-to-point link, in actual practice all security modules can be reduced to a simple challenge/response cycle.

If the RADIUS server wishes to instruct the ARAP NAS to run a security module, it should send an Access-Challenge packet to the NAS with (optionally) the State attribute, plus the ARAP-Challenge-Response, ARAP-Features, and two more attributes:

ARAP-Security: a four octet security module signature, containing a Macintosh OSType.

ARAP-Security-Data, a string to carry the actual security module challenge and response.

When the security module finishes executing, the security module response is passed in an ARAP-Security-Data attribute from the NAS to the RADIUS server in a second Access-Request, also including the State from the Access-Challenge. The authenticator field contains no special information in this case, and this can be discerned by the presence of the State attribute.

2.3. RADIUS Support for Extensible Authentication Protocol (EAP)

The Extensible Authentication Protocol (EAP), described in [3], provides a standard mechanism for support of additional authentication methods within PPP. Through the use of EAP, support for a number of authentication schemes may be added, including smart cards, Kerberos, Public Key, One Time Passwords, and others. In order to provide for support of EAP within RADIUS, two new attributes, EAP-Message and Message-Authenticator, are introduced in this document. This section describes how these new attributes may be used for providing EAP support within RADIUS.

In the proposed scheme, the RADIUS server is used to shuttle RADIUS-encapsulated EAP Packets between the NAS and a backend security server. While the conversation between the RADIUS server and the backend security server will typically occur using a proprietary protocol developed by the backend security server vendor, it is also possible to use RADIUS-encapsulated EAP via the EAP-Message attribute. This has the advantage of allowing the RADIUS server to support EAP without the need for authentication-specific code, which can instead reside on the backend security server.

2.3.1. Protocol Overview

The EAP conversation between the authenticating peer (dial-in user) and the NAS begins with the negotiation of EAP within LCP. Once EAP has been negotiated, the NAS MUST send an EAP-Request/Identity message to the authenticating peer, unless identity is determined via some other means such as Called-Station-Id or Calling-Station-Id.

The peer will then respond with an EAP-Response/Identity which the the NAS will then forward to the RADIUS server in the EAP-Message attribute of a RADIUS Access-Request packet. The RADIUS Server will typically use the EAP-Response/Identity to determine which EAP type is to be applied to the user.

In order to permit non-EAP aware RADIUS proxies to forward the Access-Request packet, if the NAS sends the EAP-Request/Identity, the NAS MUST copy the contents of the EAP-Response/Identity into the User-Name attribute and MUST include the EAP-Response/Identity in the User-Name attribute in every subsequent Access-Request. NAS-Port or NAS-Port-Id SHOULD be included in the attributes issued by the NAS in the Access-Request packet, and either NAS-Identifier or NAS-IP-Address MUST be included. In order to permit forwarding of the Access-Reply by EAP-unaware proxies, if a User-Name attribute was included in an Access-Request, the RADIUS Server MUST include the User-Name attribute in subsequent Access-Accept packets. Without the User-Name attribute, accounting and billing becomes very difficult to manage.

If identity is determined via another means such as Called-Station-Id or Calling-Station-Id, the NAS MUST include these identifying attributes in every Access-Request.

While this approach will save a round-trip, it cannot be universally employed. There are circumstances in which the user's identity may not be needed (such as when authentication and accounting is handled based on Called-Station-Id or Calling-Station-Id), and therefore an EAP-Request/Identity packet may not necessarily be issued by the NAS to the authenticating peer. In cases where an EAP-Request/Identity packet will not be sent, the NAS will send to the RADIUS server a RADIUS Access-Request packet containing an EAP-Message attribute signifying EAP-Start. EAP-Start is indicated by sending an EAP-Message attribute with a length of 2 (no data). However, it should be noted that since no User-Name attribute is included in the Access-Request, this approach is not compatible with RADIUS as specified in [1], nor can it easily be applied in situations where proxies are deployed, such as roaming or shared use networks.

If the RADIUS server supports EAP, it MUST respond with an Access-Challenge packet containing an EAP-Message attribute. If the RADIUS server does not support EAP, it MUST respond with an Access-Reject.

The EAP-Message attribute includes an encapsulated EAP packet which is then passed on to the authenticating peer. In the case where the NAS does not initially send an EAP-Request/Identity message to the peer, the Access-Challenge typically will contain an EAP-Message attribute encapsulating an EAP-Request/Identity message, requesting the dial-in user to identify themself. The NAS will then respond with a RADIUS Access-Request packet containing an EAP-Message attribute encapsulating an EAP-Response. The conversation continues until either a RADIUS Access-Reject or Access-Accept packet is received.

Reception of a RADIUS Access-Reject packet, with or without an EAP-Message attribute encapsulating EAP-Failure, MUST result in the NAS issuing an LCP Terminate Request to the authenticating peer. A RADIUS Access-Accept packet with an EAP-Message attribute encapsulating EAP-Success successfully ends the authentication phase.

The RADIUS Access-Accept/EAP-Message/EAP-Success packet MUST contain all of the expected attributes which are currently returned in an Access-Accept packet.

The above scenario creates a situation in which the NAS never needs to manipulate an EAP packet. An alternative may be used in situations where an EAP-Request/Identity message will always be sent by the NAS to the authenticating peer.

For proxied RADIUS requests there are two methods of processing. If the domain is determined based on the Called-Station-Id, the RADIUS Server may proxy the initial RADIUS Access-Request/EAP-Start. If the domain is determined based on the user's identity, the local RADIUS Server MUST respond with a RADIUS Access-Challenge/EAP-Identity packet. The response from the authenticating peer MUST be proxied to the final authentication server.

For proxied RADIUS requests, the NAS may receive an Access-Reject packet in response to its Access-Request/EAP-Identity packet. This would occur if the message was proxied to a RADIUS Server which does not support the EAP-Message extension. On receiving an Access-Reject, the NAS MUST send an LCP Terminate Request to the authenticating peer, and disconnect.

2.3.2. Retransmission

As noted in [3], the EAP authenticator (NAS) is responsible for retransmission of packets between the authenticating peer and the NAS. Thus if an EAP packet is lost in transit between the authenticating peer and the NAS (or vice versa), the NAS will retransmit. As in RADIUS [1], the RADIUS client is responsible for retransmission of packets between the RADIUS client and the RADIUS server.

Note that it may be necessary to adjust retransmission strategies and authentication timeouts in certain cases. For example, when a token card is used additional time may be required to allow the user to find the card and enter the token. Since the NAS will typically not have knowledge of the required parameters, these need to be provided by the RADIUS server. This can be accomplished by inclusion of Session-Timeout and Password-Retry attributes within the Access-Challenge packet.

If Session-Timeout is present in an Access-Challenge packet that also contains an EAP-Message, the value of the Session-Timeout provides the NAS with the maximum number of seconds the NAS should wait for an EAP-Response before retransmitting the EAP-Message to the dial-in user.

2.3.3. Fragmentation

Using the EAP-Message attribute, it is possible for the RADIUS server to encapsulate an EAP packet that is larger than the MTU on the link between the NAS and the peer. Since it is not possible for the RADIUS server to use MTU discovery to ascertain the link MTU, the Framed-MTU attribute may be included in an Access-Request packet containing an EAP-Message attribute so as to provide the RADIUS server with this information.

2.3.4. Examples

The example below shows the conversation between the authenticating peer, NAS, and RADIUS server, for the case of a One Time Password (OTP) authentication. OTP is used only for illustrative purposes; other authentication protocols could also have been used, although they might show somewhat different behavior.

```
Authenticating Peer     NAS                    RADIUS Server
-------------------     ---                    -------------
```

```
                        <- PPP LCP Request-EAP
                        auth
PPP LCP ACK-EAP
auth ->
                        <- PPP EAP-Request/
                        Identity
PPP EAP-Response/
Identity (MyID) ->
                        RADIUS
                        Access-Request/
                        EAP-Message/
                        EAP-Response/
                        (MyID) ->
                                                <- RADIUS
                                                Access-Challenge/
                                                EAP-Message/EAP-Request
                                                OTP/OTP Challenge
                        <- PPP EAP-Request/
                        OTP/OTP Challenge
PPP EAP-Response/
OTP, OTPpw ->
                        RADIUS
                        Access-Request/
                        EAP-Message/
                        EAP-Response/
                        OTP, OTPpw ->
                                                 <- RADIUS
                                                 Access-Accept/
                                                 EAP-Message/EAP-Success
                                                 (other attributes)
                        <- PPP EAP-Success
PPP Authentication
Phase complete,
NCP Phase starts
```

In the case where the NAS first sends an EAP-Start packet to the RADIUS server,   the conversation would appear as follows:

```
Authenticating Peer      NAS                     RADIUS Server
-------------------     ---                  -------------
```

```
                        <- PPP LCP Request-EAP
                        auth
PPP LCP ACK-EAP
auth ->
                        RADIUS
                        Access-Request/
                       EAP-Message/Start ->
                                               <- RADIUS
                                               Access-Challenge/
                                               EAP-Message/Identity
                        <- PPP EA-Request/
                        Identity
PPP EAP-Response/
Identity (MyID) ->
                        RADIUS
                        Access-Request/
                        EAP-Message/
                        EAP-Response/
                        (MyID) ->
                                                <- RADIUS
                                                Access-Challenge/
                                                EAP-Message/EAP-Request
                                                OTP/OTP Challenge
                        <- PPP EAP-Request/
                        OTP/OTP Challenge
PPP EAP-Response/
OTP, OTPpw ->
                        RADIUS
                        Access-Request/
                        EAP-Message/
                        EAP-Response/
                        OTP, OTPpw ->
                                                 <- RADIUS
                                                 Access-Accept/
                                                 EAP-Message/EAP-Success
                                                 (other attributes)
                        <- PPP EAP-Success
PPP Authentication
Phase complete,
NCP Phase starts
```

In the case where the client fails EAP authentication, the conversation would appear as follows:

```
Authenticating Peer      NAS                      RADIUS Server
-------------------      ---                 -------------
                         <- PPP LCP Request-EAP
                         auth
PPP LCP ACK-EAP
auth ->
                         Access-Request/
                         EAP-Message/Start ->
                                                <- RADIUS
                                                Access-Challenge/
                                                EAP-Message/Identity
                         <- PPP EAP-Request/
                         Identity
PPP EAP-Response/
Identity (MyID) ->
                         RADIUS
                         Access-Request/
                         EAP-Message/
                         EAP-Response/
                         (MyID) ->
                                                 <- RADIUS
                                                 Access-Challenge/
                                                 EAP-Message/EAP-Request
                                                 OTP/OTP Challenge
                         <- PPP EAP-Request/
                         OTP/OTP Challenge
PPP EAP-Response/
OTP, OTPpw ->
                         RADIUS
                         Access-Request/
                         EAP-Message/
                         EAP-Response/
                         OTP, OTPpw ->
                                                  <- RADIUS
                                                  Access-Reject/
                                                  EAP-Message/EAP-Failure
                         <- PPP EAP-Failure
                         (client disconnected)
```

In the case that the RADIUS server or proxy does not support EAP-Message, the conversation would appear as follows:

```
Authenticating Peer      NAS                         RADIUS Server
-------------------      ---                    -------------

                         <- PPP LCP Request-EAP
                         auth
PPP LCP ACK-EAP
auth ->
                         RADIUS
                         Access-Request/
                         EAP-Message/Start ->
                                                   <- RADIUS
                                                   Access-Reject
                         <- PPP LCP Terminate
                         (User Disconnected)
```

In the case where the local RADIUS Server does support EAP-Message, but the remote RADIUS Server does not, the conversation would appear as follows:

```
Authenticating Peer      NAS                              RADIUS Server
-------------------      ---                              -------------
                              <- PPP LCP Request-EAP
                              auth
PPP LCP ACK-EAP
auth ->
                              RADIUS
                              Access-Request/
                              EAP-Message/Start ->
                                                          <- RADIUS
                                                          Access-Challenge/
                                                          EAP-Message/Identity
                              <- PPP EAP-Request/
                              Identity
PPP EAP-Response/
Identity
(MyID) ->
                              RADIUS
                              Access-Request/
                              EAP-Message/EAP-Response/
                              (MyID) ->
                                                          <- RADIUS
                                                          Access-Reject
                                                          (proxied from remote
                                                           RADIUS Server)
                              <- PPP LCP Terminate
                              (User Disconnected)
```

In the case where the authenticating peer does not support EAP, but where EAP is required for that user, the conversation would appear as follows:

```
Authenticating Peer      NAS                              RADIUS Server
-------------------      ---                              -------------
                              <- PPP LCP Request-EAP
                              auth
PPP LCP NAK-EAP
auth ->
                              <- PPP LCP Request-CHAP
                              auth
PPP LCP ACK-CHAP
auth ->
                              <- PPP CHAP Challenge
PPP CHAP Response ->
                              RADIUS
                              Access-Request/
                              User-Name,
                              CHAP-Password ->
                                                          <- RADIUS
                                                          Access-Reject
                              <-   PPP LCP Terminate
                              (User Disconnected)
```

In the case where the NAS does not support EAP, but where EAP is required for that user, the conversation would appear as follows:

```
Authenticating Peer       NAS                              RADIUS Server
-------------------       ---                              -------------

                         <- PPP LCP Request-CHAP
                         auth
PP LCP ACK-CHAP
auth ->
                         <- PPP CHAP Challenge
PPP CHAP Response ->
                         RADIUS
                         Access-Request/
                         User-Name,
                         CHAP-Password ->

                                                   <- RADIUS
                                                   Access-Reject
                         <-   PPP LCP Terminate
                         (User Disconnected)
```

2.3.5. Alternative uses

Currently the conversation between the backend security server and the RADIUS server is proprietary because of lack of standardization.

In order to increase standardization and provide interoperability between Radius vendors and backend security vendors, it is recommended that RADIUS-encapsulated EAP be used for this conversation.

This has the advantage of allowing the RADIUS server to support EAP without the need for authentication-specific code within the RADIUS server. Authentication-specific code can then reside on a backend security server instead.

In the case where RADIUS-encapsulated EAP is used in a conversation between a RADIUS server and a backend security server, the security server will typically return an Access-Accept/EAP-Success message without inclusion of the expected attributes currently returned in an Access-Accept. This means that the RADIUS server MUST add these attributes prior to sending an Access-Accept/EAP-Success message to the NAS.

3. Packet Format

Packet Format is identical to that defined in RFC 2865 [17] and 2866 [5].

4. Packet Types

Packet types are identical to those defined in RFC 2865 [17] and 2866 [5].

See "Table of Attributes" below to determine which types of packets can contain which attributes defined here.

5. Attributes

5. Attributes

NCC Implementation Notes:

In 3GPP2 mode, RCA cannot handle any of the attributes defined in this document, except event-Timestamp. (See below.)

In parameterised mode, RCA can deal with any core Radius attribute or vendor specific attribute (whether or not they are listed in this document) provided that:

- It is (or can be treated as) of type text, octets, IPv4 address or number
- The attribute is at the top level. (i.e. RCA in parameterised mode cannot cope with attributes within attributes.

In the above note, the phrase "RCA can deal with [an attribute]" means that, in parameterised mode, RCA can use the attribute in decision making concerning credit control, in a configurable way, and / or store the attribute to be sent out again in Radius messages and / or send the attribute in outgoing messages on a per message type basis.

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 order of attributes of the same type SHOULD be preserved. The order of attributes of different types is not required to be preserved.

NCC Implementation Notes:

- RCA cannot cope with receiving multiple attributes of the same type, except Vendor-Specific attributes which it can cope with.
- In 3GPP2 mode, RCA will never send a Radius message with more than one attribute of the same type, except for Vendor-Specific attributes.
- In parameterised mode, RCA can be configured to send more than one attribute of the same type, for any type.

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

A summary of the attribute format is the same as in RFC 2865 [1] but is included here for ease of reference. The fields are transmitted from left to right.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |  Value ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 [17])
40-51	(refer to RADIUS document [5])
52	Acct-Input-Gigawords
53	Acct-Output-Gigawords
54	Unused
55	Event-Timestamp
56-59	Unused
60-63	(refer to RADIUS document [17])
64-67	(refer to [19])

68	(refer to [20])
69	(refer to [19])
70	ARAP-Password
71	ARAP-Features
73	ARAP-Security
74	ARAP-Security-Data
75	Password-Retry
76	Prompt
77	Connect-Info
78	Configuration-Token
79	EAP-Message
80	Message-Authenticator
81-83	(refer to [19])
84	ARAP-Challenge-Response
85	Acct-Interim-Interval
86	(refer to [20])
87	NAS-Port-Id
88	Framed-Pool
89	Unused
90-91	(refer to [19])
92-191	Unused

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 a packet with an invalid Length, the entire request should 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."

NCC Implementation Note:

RCA cannot cope with embedded nulls in text type attributes.

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.

5.1. Acct-Input-Gigawords

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

- In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

Description

This attribute indicates how many times the Acct-Input-Octets counter has wrapped around 2^32 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 or Interim-Update.

A summary of the Acct-Input-Gigawords attribute format is shown below. 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	52 for Acct-Input-Gigawords.
Length	6
Value	The Value field is four octets.

5.2. Acct-Output-Gigawords

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

- In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

Description

This attribute indicates how many times the Acct-Output-Octets counter has wrapped around 2^32 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 or Interim-Update.

A summary of the Acct-Output-Gigawords attribute format is shown below. 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	53 for Acct-Output-Gigawords.
Length	6
Value	The Value field is four octets.

5.3. Event-Timestamp

NCC Implementation Notes:

- In 3GPP2 mode, RCA looks for this attribute in Access-Request messages. (NB. NOT in Accounting-Request messages.) If the attribute is present, RCA checks that the time makes sense and sends an Access-reject if it does not.
- In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

Description

This attribute is included in an Accounting-Request packet to record the time that this event occurred on the NAS, in seconds since January 1, 1970 00:00 UTC.

A summary of the Event-Timestamp attribute format is shown below. 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	55 for Event-Timestamp.
Length	6
Value	The Value field is four octets encoding an unsigned integer with the number of seconds since January 1, 1970 00:00 UTC.

5.4. ARAP-Password

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

- In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

Description

This attribute is only present in an Access-Request packet containing a Framed-Protocol of ARAP.

Only one of User-Password, CHAP-Password, or ARAP-Password needs to be present in an Access-Request, or one or more EAP-Messages.

A summary of the ARAP-Password attribute format is shown below. 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     |             Value1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |             Value2
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |             Value3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |             Value4
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	70 for ARAP-Password.
Length	18
Value	This attribute contains a 16 octet string, used to carry the dial-in user's response to the NAS challenge and the client's own challenge to the NAS. The high-order octets (Value1 and Value2) contain the dial-in user's challenge to the NAS (2 32-bit numbers, 8 octets) and the low-order octets (Value3 and Value4) contain the dial-in user's response to the NAS challenge (2 32-bit numbers, 8 octets).

5.5. ARAP-Features

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

- In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

Description

This attribute is sent in an Access-Accept packet with Framed-Protocol of ARAP, and includes password information that the NAS should sent to the user in an ARAP "feature flags" packet.

A summary of the ARAP-Features attribute format is shown below. 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     |     Value1    |    Value2     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Value3                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Value4                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Value5                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	71 for ARAP-Features.
Length	16

Value

The Value field is a compound string containing information the NAS should send to the user in the ARAP "feature flags" packet.

Value 1: If zero, user cannot change their password. If non-zero user can. (RADIUS does not handle the password changing, just the attribute which indicates whether ARAP indicates they can.)

Value 2: Minimum acceptable password length, from 0 to 8.

Value 3: Password creation date in Macintosh format, defined as 32 unsigned bits representing seconds since Midnight GMT January 1, 1904.

Value 4: Password Expiration Delta from create date in seconds.

Value 5: Current RADIUS time in Macintosh format

5.6. ARAP-Zone-Access

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

- In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

Description

This attribute is included in an Access-Accept packet with Framed-Protocol of ARAP to indicate how the ARAP zone list for the user should be used.

A summary of the ARAP-Zone-Access attribute format is shown below. 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type

72 for ARAP-Zone-Access.

Length

6

Value

The Value field is four octets encoding an integer with one of the following values:

1 Only allow access to default zone

2 Use zone filter inclusively

4 Use zone filter exclusively

The value 3 is skipped, not because these are bit flags, but because 3 in some ARAP implementations means "all zones" which is the same as not specifying a list at all under RADIUS.

If this attribute is present and the value is 2 or 4 then a Filter-Id must also be present to name a zone list filter to apply the access flag to.

ARAP-Security

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

- In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

Description

This attribute identifies the ARAP Security Module to be used in an Access-Challenge packet.

A summary of the ARAP-Security attribute format is shown below.   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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	73 for ARAP-Security.
Length	6
Value	The Value field is four octets, containing an integer specifying the security module signature, which is a Macintosh OSType. (Macintosh OSTypes are 4 ascii characters cast as a 32-bit integer)

5.8. ARAP-Security-Data

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

- In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

Description

This attribute contains the actual security module challenge or response, and can be found in Access-Challenge and Access-Request packets.

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

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |     String...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	74 for ARAP-Security-Data.
Length	>=3
String	The String field contains the security module challenge or response associated with the ARAP Security Module specified in ARAP-Security..

5.9. Password-Retry

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

- In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

Description

This attribute MAY be included in an Access-Reject to indicate how many authentication attempts a user may be allowed to attempt before being disconnected.

It is primarily intended for use with ARAP authentication.

A summary of the Password-Retry attribute format is shown below. 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	75 for Password-Retry.
Length	6
Value	The Value field is four octets, containing an integer specifying the number of password retry attempts to permit the user.

5.10. Prompt

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis. NB. This is a bit academic as RCA does not handle Access-Challenge messages.

Decription

This attribute is used only in Access-Challenge packets, and indicates to the NAS whether it should echo the user's response as it is entered, or not echo it.

A summary of the Prompt attribute format is shown below. 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	76 for Prompt.
Length	6
Value	The Value field is four octets.

0	No Echo
1	Echo

5.11. Connect-Info

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

Description

This attribute is sent from the NAS to indicate the nature of the user's connection.

The NAS MAY send this attribute in an Access-Request or Accounting-Request to indicate the nature of the user's connection.

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

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |     Text...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type 77 for Connect-Info.

Length >=3

Text The Text field consists of UTF-8 encoded 10646 [8] characters. The connection speed SHOULD be included at the beginning of the first Connect-Info attribute in the packet. If the transmit and receive connection speeds differ, they may both be included in the first attribute with the transmit speed first (the speed the NAS modem transmits at), a slash (/), the receive speed, then optionally other information.

For example, "28800 V42BIS/LAPM" or "52000/31200 V90"

More than one Connect-Info attribute may be present in an Accounting-Request packet to accommodate expected efforts by ITU to have modems report more connection information in a standard format that might exceed 252 octets.

5.12. Configuration-Token

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

Description

This attribute is for use in large distributed authentication networks based on proxy. It is sent from a RADIUS Proxy Server to a RADIUS Proxy Client in an Access-Accept to indicate a type of user profile to be used. It should not be sent to a NAS.

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

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |  String ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type 78 for Configuration-Token.

Length >=3

String The String field is one or more octets. The actual format of the information is site or application specific, and a robust implementation SHOULD support the field as undistinguished octets.

The codification of the range of allowed usage of this field is outside the scope of this specification.

5.13. EAP-Message

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

Description

This attribute encapsulates Extended Access Protocol [3] packets so as to allow the NAS to authenticate dial-in users via EAP without having to understand the EAP protocol.

The NAS places any EAP messages received from the user into one or more EAP attributes and forwards them to the RADIUS Server as part of the Access-Request, which can return EAP messages in Access-Challenge, Access-Accept and Access-Reject packets.

A RADIUS Server receiving EAP messages that it does not understand SHOULD return an Access-Reject.

The NAS places EAP messages received from the authenticating peer into one or more EAP-Message attributes and forwards them to the RADIUS Server within an Access-Request message. If multiple EAP-Messages are contained within an Access-Request or Access-Challenge packet, they MUST be in order and they MUST be consecutive attributes in the Access-Request or Access-Challenge packet. Access-Accept and Access-Reject packets SHOULD only have ONE EAP-Message attribute in them, containing EAP-Success or EAP-Failure.

It is expected that EAP will be used to implement a variety of authentication methods, including methods involving strong cryptography. In order to prevent attackers from subverting EAP by attacking RADIUS/EAP, (for example, by modifying the EAP-Success or EAP-Failure packets) it is necessary that RADIUS/EAP provide integrity protection at least as strong as those used in the EAP methods themselves.

Therefore the Message-Authenticator attribute MUST be used to protect all Access-Request, Access-Challenge, Access-Accept, and Access-Reject packets containing an EAP-Message attribute.

Access-Request packets including an EAP-Message attribute without a Message-Authenticator attribute SHOULD be silently discarded by the RADIUS server. A RADIUS Server supporting EAP-Message MUST calculate the correct value of the Message-Authenticator and silently discard the packet if it does not match the value sent.

A RADIUS Server not supporting EAP-Message MUST return an Access-Reject if it receives an Access-Request containing an EAP-Message attribute. A RADIUS Server receiving an EAP-Message attribute that it does not understand MUST return an Access-Reject.

Access-Challenge, Access-Accept, or Access-Reject packets including an EAP-Message attribute without a Message-Authenticator attribute SHOULD be silently discarded by the NAS. A NAS supporting EAP-Message MUST calculate the correct value of the Message-Authenticator and silently discard the packet if it does not match the value sent.

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

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |     String...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type 79 for EAP-Message.

Length >=3

String The String field contains EAP packets, as defined in [18]. If multiple EAP-Message attributes are present in a packet their values should be concatenated; this allows EAP packets longer than 253 octets to be passed by RADIUS.

5.14. Message-Authenticator

Description

This attribute MAY be used to sign Access-Requests to prevent spoofing Access-Requests using CHAP, ARAP or EAP authentication methods. It MAY be used in any Access-Request. It MUST be used in any Access-Request, Access-Accept, Access-Reject or Access-Challenge that includes an EAP-Message attribute.

NCC Implementation Notes:

The RCA does not support the following:

- A RADIUS Server receiving an Access-Request with a Message-Authenticator Attribute present MUST calculate the correct value of the Message-Authenticator and silently discard the packet if it does not match the value sent.
- A RADIUS Client receiving an Access-Accept, Access-Reject or Access-Challenge with a Message-Authenticator Attribute present MUST calculate the correct value of the Message-Authenticator and silently discard the packet if it does not match the value sent.
- Earlier drafts of this memo used "Signature" as the name of this attribute, but Message-Authenticator is more precise. Its operation has not changed, just the name.

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

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |     String...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type 80 for Message-Authenticator.

Length 18

String

When present in an Access-Request packet, Message-Authenticator is an HMAC-MD5 [9] checksum of the entire Access-Request packet, including Type, ID, Length and authenticator, using the shared secret as the key, as follows.

Message-Authenticator = HMAC-MD5 (Type, Identifier, Length, Request Authenticator, Attributes)

When the checksum is calculated the signature string should be considered to be sixteen octets of zero.

For Access-Challenge, Access-Accept, and Access-Reject packets, the Message-Authenticator is calculated as follows, using the Request-Authenticator from the Access-Request this packet is in reply to:

Message-Authenticator = HMAC-MD5 (Type, Identifier, Length, Request Authenticator, Attributes)

When the checksum is calculated the signature string should be considered to be sixteen octets of zero. The shared secret is used as the key for the HMAC-MD5 hash. The is calculated and inserted in the packet before the Response Authenticator is calculated.

This attribute is not needed if the User-Password attribute is present, but is useful for preventing attacks on other types of authentication. This attribute is intended to thwart attempts by an attacker to setup a "rogue" NAS, and perform online dictionary attacks against the RADIUS server. It does not afford protection against "offline" attacks where the attacker intercepts packets containing (for example) CHAP challenge and response, and performs a dictionary attack against those packets offline.

IP Security will eventually make this attribute unnecessary, so it should be considered an interim measure.

5.15. ARAP-Challenge-Response

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

Description

This attribute is sent in an Access-Accept packet with Framed-Protocol of ARAP, and contains the response to the dial-in client's challenge.

A summary of the ARAP-Challenge-Response attribute format is shown below. 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     |     Value...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type

84 for ARAP-Challenge-Response.

Length

10

Value	The Value field contains an 8 octet response to the dial-in client's challenge. The RADIUS server calculates this value by taking the dial-in client's challenge from the high order 8 octets of the ARAP-Password attribute and performing DES encryption on this value with the authenticating user's password as the key. If the user's password is less than 8 octets in length, the password is padded at the end with NULL octets to a length of 8 before using it as a key.

5.16. Acct-Interim-Interval

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

Description

This attribute indicates the number of seconds between each interim update in seconds for this specific session. This value can only appear in the Access-Accept message.

A summary of the Acct-Interim-Interval attribute format is shown below. 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	85 for Acct-Interim-Interval.

Length	6

Value	The Value field contains the number of seconds between each interim update to be sent from the NAS for this session. The value MUST NOT be smaller than 60. The value SHOULD NOT be smaller than 600, and careful consideration should be given to its impact on network traffic.

5.17. NAS-Port-Id

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

Description

This Attribute contains a text string which identifies the port of the NAS which is authenticating the user. It is only used in Access-Request and Accounting-Request packets. Note that this is using "port" in its sense of a physical connection on the NAS, not in the sense of a TCP or UDP port number.

Either NAS-Port or NAS-Port-Id SHOULD be present in an Access-Request packet, if the NAS differentiates among its ports. NAS-Port-Id is intended for use by NASes which cannot conveniently number their ports.

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

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |     Text...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	87 for NAS-Port-Id.
Length	>=3
Text	The Text field contains the name of the port using UTF-8 encoded 10646 [7] characters.

5.18. Framed-Pool

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

Description

This Attribute contains the name of an assigned address pool that SHOULD be used to assign an address for the user.   If a NAS does not support multiple address pools, the NAS should ignore this Attribute.   Address pools are usually used for IP addresses, but can be used for other protocols if the NAS supports pools for those protocols.

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

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |     String...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	88 for Framed-Pool.
Length	>=3
String	The string field contains the name of an assigned address pool configured on the NAS.

5.19. Table of Attributes

The following table provides a guide to which attributes may be found in which kind of packets.   Acct-Input-Gigawords, Acct-Output-Gigawords, Event-Timestamp, and NAS-Port-Id may have 0-1 instances in an Accounting-Request packet.

Connect-Info may have the following instances in an Accounting-Request packet:

0+ 0-1

Where 0+ is not supported, and 0-1 uses non-standard implementation.

The other attributes added in this document must not be present in an Accounting-Request.

Request	Accept	Reject	Challenge	#	Attribute
0-1	0	0	0	70	ARAP-Password [Note 1]

0	0-1	0	0-1	71	ARAP-Features
0	0-1	0	0	72	ARAP-Zone-Access
0-1	0	0	0-1	73	ARAP-Security
0+(not supported) 0-1 (non-standard implementation)	0	0	0+(not supported) 0-1 (non-standard implementation)	74	ARAP-Security-Data
0	0	0-1	0	75	Password-Retry
0	0	0	0-1	76	Prompt
0-1	0	0	0	77	Connect-Info
0	0+(not supported) 0-1 (non-standard implementation)	0	0	78	Configuration-Token
0+(not supported) 0-1 (non-standard implementation)	0+(not supported) 0-1 (non-standard implementation)	0+(not supported) 0-1 (non-standard implementation)	0+(not supported) 0-1 (non-standard implementation)	79	EAP-Message [Note 1]
0-1	0-1	0-1	0-1	80	Message-Authenticator [Note 1]
0	0-1	0	0-1	84	ARAP-Challenge-Response
0	0-1	0	0	85	Acct-Interim-Interval
0-1	0	0	0	87	NAS-Port-Id
0	0-1	0	0	88	Framed-Pool

[Note 1] An Access-Request that contains either a User-Password or CHAP-Password or ARAP-Password or one or more EAP-Message attributes MUST NOT contain more than one type of those four attributes. If it does not contain any of those four attributes, it SHOULD contain a Message-Authenticator. If any packet type contains an EAP-Message attribute it MUST also contain a Message-Authenticator.

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 [17], in accordance with BCP 26 [13].

7. Security Considerations

The attributes other than Message-Authenticator and EAP-Message in this document have no additional security considerations beyond those already identified in [17].

7.1. Message Authenticator

Access-Request packets with a User-Password establish the identity of both the user and the NAS sending the Access-Request, because of the way the shared secret between NAS and RADIUS server is used.

Access-Request packets with CHAP-Password or EAP-Message do not have a User-Password attribute, so the Message-Authenticator attribute should be used in access-request packets that do not have a User-Password, in order to establish the identity of the NAS sending the request.

7.2. EAP security

Since the purpose of EAP is to provide enhanced security for PPP authentication, it is critical that RADIUS support for EAP be secure.

In particular, the following issues must be addressed:

- Separation of EAP server and PPP authenticator
- Connection hijacking
- Man in the middle attacks
- Multiple databases
- Negotiation attacks

7.2.1. Separation of EAP server

It is possible for the EAP endpoints to mutually authenticate, negotiate a ciphersuite, and derive a session key for subsequent use in PPP encryption.

This does not present an issue on the peer, since the peer and EAP client reside on the same machine; all that is required is for the EAP client module to pass the session key to the PPP encryption module.

The situation is more complex when EAP is used with RADIUS, since the PPP authenticator will typically not reside on the same machine as the EAP server. For example, the EAP server may be a backend security server, or a module residing on the RADIUS server.

In the case where the EAP server and PPP authenticator reside on different machines, there are several implications for security.

Firstly, mutual authentication will occur between the peer and the EAP server, not between the peer and the authenticator. This means that it is not possible for the peer to validate the identity of the NAS or tunnel server that it is speaking to.

As described earlier, when EAP/RADIUS is used to encapsulate EAP packets, the Message-Authenticator attribute is required in EAP/RADIUS Access-Requests sent from the NAS or tunnel server to the RADIUS server. Since the Message-Authenticator attribute involves a HMAC-MD5 hash, it is possible for the RADIUS server to verify the integrity of the Access-Request as well as the NAS or tunnel server's identity. Similarly, Access-Challenge packets sent from the RADIUS server to the NAS are also authenticated and integrity protected using an HMAC-MD5 hash, enabling the NAS or tunnel server to determine the integrity of the packet and verify the identity of the RADIUS server. Moreover, EAP packets sent via methods that contain their own integrity protection cannot be successfully modified by a rogue NAS or tunnel server.

The second issue that arises in the case of an EAP server and PPP authenticator residing on different machines is that the session key negotiated between the peer and EAP server will need to be transmitted to the authenticator. Therefore a mechanism needs to be provided to transmit the session key from the EAP server to the authenticator or tunnel server that needs to use the key. The specification of this transit mechanism is outside the scope of this document.

7.2.2. Connection hijacking

In this form of attack, the attacker attempts to inject packets into the conversation between the NAS and the RADIUS server, or between the RADIUS server and the backend security server. RADIUS does not support encryption, and as described in [17], only Access-Reply and Access-Challenge packets are integrity protected. Moreover, the integrity protection mechanism described in [17] is weaker than that likely to be used by some EAP methods, making it possible to subvert those methods by attacking EAP/RADIUS.

In order to provide for authentication of all packets in the EAP exchange, all EAP/RADIUS packets MUST be authenticated using the Message-Authenticator attribute, as described previously.

7.2.3. Man in the middle attacks

Since RADIUS security is based on shared secrets, end-to-end security is not provided in the case where authentication or accounting packets are forwarded along a proxy chain. As a result, attackers gaining control of a RADIUS proxy will be able to modify EAP packets in transit.

7.2.4. Multiple databases

In many cases a backend security server will be deployed along with a RADIUS server in order to provide EAP services. Unless the backend security server also functions as a RADIUS server, two separate user databases will exist, each containing information about the security requirements for the user. This represents a weakness, since security may be compromised by a successful attack on either of the servers, or their backend databases. With multiple user databases, adding a new user may require multiple operations, increasing the chances for error. The problems are further magnified in the case where user information is also being kept in an LDAP server. In this case, three stores of user information may exist.

In order to address these threats, consolidation of databases is recommended. This can be achieved by having both the RADIUS server and backend security server store information in the same backend database; by having the backend security server provide a full RADIUS implementation; or by consolidating both the backend security server and the RADIUS server onto the same machine.

7.2.5. Negotiation attacks

In a negotiation attack, a rogue NAS, tunnel server, RADIUS proxy or RADIUS server causes the authenticating peer to choose a less secure authentication method so as to make it easier to obtain the user's password. For example, a session that would normally be authenticated with EAP would instead authenticated via CHAP or PAP; alternatively, a connection that would normally be authenticated via one EAP type occurs via a less secure EAP type, such as MD5. The threat posed by rogue devices, once thought to be remote, has gained currency given compromises of telephone company switching systems, such as those described in [11].

Protection against negotiation attacks requires the elimination of downward negotiations. This can be achieved via implementation of per-connection policy on the part of the authenticating peer, and per-user policy on the part of the RADIUS server.

For the authenticating peer, authentication policy should be set on a per-connection basis. Per-connection policy allows an authenticating peer to negotiate EAP when calling one service, while negotiating CHAP for another service, even if both services are accessible via the same phone number.

With per-connection policy, an authenticating peer will only attempt to negotiate EAP for a session in which EAP support is expected. As a result, there is a presumption that an authenticating peer selecting EAP requires that level of security. If it cannot be provided, it is likely that there is some kind of misconfiguration, or even that the authenticating peer is contacting the wrong server. Should the NAS not be able to negotiate EAP, or should the EAP-Request sent by the NAS be of a different EAP type than what is expected, the authenticating peer MUST disconnect. An authenticating peer expecting EAP to be negotiated for a session MUST NOT negotiate CHAP or PAP.

For a NAS, it may not be possible to determine whether a user is required to authenticate with EAP until the user's identity is known. For example, for shared-uses NASes it is possible for one reseller to implement EAP while another does not. In such cases, if any users of the NAS MUST do EAP, then the NAS MUST attempt to negotiate EAP for every call. This avoids forcing an EAP-capable client to do more than one authentication, which weakens security.

If CHAP is negotiated, the NAS will pass the User-Name and CHAP-Password attributes to the RADIUS Server in an Access-Request packet. If the user is not required to use EAP, then the RADIUS Server will respond with an Access-Accept or Access-Reject packet as appropriate. However, if CHAP has been negotiated but EAP is required, the RADIUS server MUST respond with an Access-Reject, rather than an Access-Challenge/EAP-Message/EAP-Request packet. The authenticating peer MUST refuse to renegotiate authentication, even if the renegotiation is from CHAP to EAP.

If EAP is negotiated but is not supported by the RADIUS proxy or server, then the server or proxy MUST respond with an Access-Reject.

In these cases, the NAS MUST send an LCP-Terminate and disconnect the user. This is the correct behavior since the authenticating peer is expecting EAP to be negotiated, and that expectation cannot be fulfilled. An EAP-capable authenticating peer MUST refuse to renegotiate the authentication protocol if EAP had initially been negotiated. Note that problems with a non-EAP capable RADIUS proxy could prove difficult to diagnose, since a user dialing in from one location (with an EAP-capable proxy) might be able to successfully authenticate via EAP, while the same user dialing into another location (and encountering an EAP-incapable proxy) might be consistently disconnected.

8. References

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

[2] Rigney, C., "RADIUS Accounting", RFC 2866, June 2000.

[3] Blunk, L. and J. Vollbrecht, "PPP Extensible Authentication Protocol (EAP)", RFC 2284, March 1998.

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

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

[6] Zorn, G., Leifer, D., Rubens, A., Shriver, J., Holdrege, M. and I. Goyret, "RADIUS Attributes for Tunnel Protocol Support", RFC 2868, June 2000.

[7] Zorn, G., Aboba, B. and D. Mitton, "RADIUS Accounting Modifications for Tunnel Protocol Support", RFC 2867, June 0.

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

[9] Krawczyk, H., Bellare, M. and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[10] Alvestrand, H. and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1988.

[11] Slatalla, M., and Quittner, J., "Masters of Deception." HarperCollins, New York, 1995.

9. Acknowledgements

RADIUS and RADIUS Accounting were originally developed by Livingston Enterprises (now part of Lucent Technologies) for their PortMaster series of Network Access Servers.

The section on ARAP is adopted with permission from "Using RADIUS to Authenticate Apple Remote Access Connections" by Ward Willats of Cyno Technologies (ward@cyno.com).

The section on Acct-Interim-Interval is adopted with permission from an earlier work in progress by Pat Calhoun of Sun Microsystems, Mark Beadles of Compuserve, and Alex Ratcliffe of UUNET Technologies.

The section on EAP is adopted with permission from an earlier work in progress by Pat Calhoun of Sun Microsystems, Allan Rubens of Merit Network, and Bernard Aboba of Microsoft. Thanks also to Dave Dawson and Karl Fox of Ascend, and Glen Zorn and Narendra Gidwani of Microsoft for useful discussions of this problem space.

10. 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

11. Authors' Addresses

Questions about this memo can also be directed to:

Carl Rigney
Livingston Enterprises
4464 Willow Road
Pleasanton, California 94588

EMail: cdr@telemancy.com

Questions on ARAP and RADIUS may be directed to:

Ward Willats
Cyno Technologies
1082 Glen Echo Ave
San Jose, CA 95125
Phone: +1 408 297 7766
EMail: ward@cyno.com

Questions on EAP and RADIUS may be directed to any of the following:

Pat R. Calhoun
Network and Security Research Center
Sun Microsystems, Inc.
15 Network Circle
Menlo Park, CA 94025

Phone: +1 650 786 7733
EMail: pcalhoun@eng.sun.com

Allan C. Rubens
Tut Systems, Inc.
220 E. Huron, Suite 260
Ann Arbor, MI 48104

Phone: +1 734 995 1697
EMail: arubens@tutsys.com

Bernard Aboba
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Phone: +1 425 936 6605
EMail: bernarda@microsoft.com

12. Full Copyright Statement

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

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

Chapter 4

Compliance to RFC 3576 (Dynamic Extensions)

Overview

Introduction

This chapter describes Oracle Communications Network Charging and Control (NCC) compliance to RFC 3576 (Dynamic Extensions).

In this chapter

This chapter contains the following topics.

Dynamic Extensions

Dynamic Extensions

Network Working Group	M. Chiba
Request for Comments: 3576	G. Dommety
Category: Informational	M. Eklund
	Cisco Systems, Inc.
	D. Mitton
	Circular Logic, UnLtd.
	B. Aboba
	Microsoft Corporation
	July 2003

Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)

NCC Implementation Notes:

RCA uses the Disconnect-Request message described in this RFC. It sends Disconnect-Request under some error conditions and can laso be configured to send Disconnect-Request when the subscriber is out of credit.

Status of this Memo

Copyright Notice

Abstract

This document describes a currently deployed extension to the Remote Authentication Dial In User Service (RADIUS) protocol, allowing dynamic changes to a user session, as implemented by network access   server products.   This includes support for disconnecting users and changing authorizations applicable to a user session.

Table of Contents

1. Introduction

The RADIUS protocol, defined in [RFC2865], does not support unsolicited messages sent from the RADIUS server to the Network Access Server (NAS).

However, there are many instances in which it is desirable for changes to be made to session characteristics, without requiring the NAS to initiate the exchange.   For example, it may be desirable for administrators to be able to terminate a user session in progress.

Alternatively, if the user changes authorisation level, this may require that authorisation attributes be added/deleted from a user session.

To overcome these limitations, several vendors have implemented additional RADIUS commands in order to be able to support unsolicited messages sent from the RADIUS server to the NAS.   These extended commands provide support for Disconnect and Change-of-Authorisation (CoA) messages. Disconnect messages cause a user session to be terminated immediately, whereas CoA messages modify session authorisation attributes such as data filters.

NCC Implementation Notes:

RCA does not use CoA messages.

1.1. Applicability

This protocol is being recommended for publication as an Informational RFC rather than as a standards-track RFC because of problems that cannot be fixed without creating incompatibilities with deployed implementations.   This includes security vulnerabilities, as well as semantic ambiguities resulting from the design of the Change-of-Authorization (CoA) commands.   While fixes are recommended, they cannot be made mandatory since this would be incompatible with existing implementations.

Existing implementations of this protocol do not support authorisation checks, so that an ISP sharing a NAS with another ISP could disconnect or change authorizations for another ISP's users.   In order to remedy this problem, a "Reverse Path Forwarding" check is recommended.   See Section 5.1. for details.

Existing implementations utilize per-packet authentication and integrity protection algorithms with known weaknesses [MD5Attack]. To provide stronger per-packet authentication and integrity protection, the use of IPsec is recommended.   See Section 5.3. for details.

Existing implementations lack replay protection.   In order to support replay detection, it is recommended that the Event-Timestamp Attribute be added to all messages in situations where IPsec replay protection is not employed.   Implementations should be configurable to silently discard messages lacking the Event-Timestamp Attribute. See Section 5.4. for details.

The approach taken with CoA commands in existing implementations results in a semantic ambiguity. Existing implementations of the CoA-Request identify the affected session, as well as supply the authorization changes.   Since RADIUS Attributes included within existing implementations of the CoA-Request can be used for session identification or authorization change, it may not be clear which function a given attribute is serving.

The problem does not exist within [Diameter], in which authorization change is requested by a command using Attribute Value Pairs (AVPs) solely for identification, resulting in initiation of a standard Request/Response sequence where authorization changes are supplied.   As a result, in no command can Diameter AVPs have multiple potential meanings.

Due to differences in handling change-of-authorization requests in RADIUS and Diameter, it may be difficult or impossible for a Diameter/RADIUS gateway to successfully translate existing implementations of this specification to equivalent messages in Diameter.   For example, a Diameter command changing any attribute used for identification within existing CoA-Request implementations cannot be translated, since such an authorization change is impossible to carry out in existing implementations.   Similarly, translation between existing implementations of Disconnect-Request or CoA-Request messages and Diameter is tricky because a Disconnect-Request or CoA-Request message will need to be translated to multiple Diameter commands.

To simplify translation between RADIUS and Diameter, a Service-Type Attribute with value "Authorize Only" can (optionally) be included within a Disconnect-Request or CoA-Request.   Such a Request contains only identification attributes.   A NAS supporting the "Authorize Only" Service-Type within a Disconnect-Request or CoA-Request responds with a NAK containing a Service-Type Attribute with value "Authorize Only" and an Error-Cause Attribute with value "Request Initiated".   The NAS will then send an Access-Request containing a Service-Type Attribute with a value of "Authorize Only".   This usage sequence is akin to what occurs in Diameter and so is more easily translated by a Diameter/RADIUS gateway.

1.2. Requirements Language

In this document, several words are used to signify the requirements of the specification. These words are often capitalized. 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 [RFC2119].

1.3. Terminology

This document frequently uses the following terms:

Network Access Server (NAS): The device providing access to the network.

service: The NAS provides a service to the user, such as IEEE 802 or PPP.

session: Each service provided by the NAS to a 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.

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. Overview

This section describes the most commonly implemented features of Disconnect and Change-of-Authorization messages.

2.1. Disconnect Messages (DM)

A Disconnect-Request packet is sent by the RADIUS server in order to terminate a user session on a NAS and discard all associated session context. The Disconnect-Request packet is sent to UDP port 3799, and identifies the NAS as well as the user session to be terminated by inclusion of the identification attributes described in Section 3.

NCC Implementation Notes:

In RCA, the default port for sending Disconnect-Request messages is 3799 but any other port number can be configured instead.

```
+----------+   Disconnect-Request     +----------+
|          |   <--------------------  |          |
|    NAS   |                          |  RADIUS  |
|          |   Disconnect-Response    |  Server  |
|          |   ---------------------> |          |
+----------+                          +----------+
```

The NAS responds to a Disconnect-Request packet sent by a RADIUS server with a Disconnect-ACK if all associated session context is discarded and the user session is no longer connected, or a Disconnect-NAK, if the NAS was unable to disconnect the session and discard all associated session context. A NAS MUST respond to a Disconnect-Request including a Service-Type Attribute with value "Authorize Only" with a Disconnect-NAK; a Disconnect-ACK MUST NOT be sent. A NAS MUST respond to a Disconnect-Request including a Service-Type Attribute with an unsupported value with a Disconnect-NAK; an Error-Cause Attribute with value "Unsupported Service" MAY be included. A Disconnect-ACK MAY contain the Attribute Acct-Terminate-Cause (49) [RFC2866] with the value set to 6 for Admin-Reset.

NCC Implementation Notes:

In 3GPP2 mode, RCA expects to receive a reply (Disconnect-ACK or Disconnect-NAK) to a Disconnect-Request.

In parameterised mode, RCA usually expects a reply to a Disconnect-Request but ccn be configured to not expect a reply on a per client type basis. (This is no standard behaviour but some clients do not reply top Disconnect-Request.)

If RCA expects a reply to Disconnect-Request, it will resend the Disconnect-Request if no reply has been received for 3 seconds. RCA continues to resend the Disconnect-Request until either a reply is received or until the (configurable) maximum number of retries has been exceeded.

2.2. Change of Authorization Messages

NCC Implementation Notes:

RCA does not support the following:

CoA-Request packets contain information for dynamically changing session authorizations. This is typically used to change data filters. The data filters can be of either the ingress or egress kind, and are sent in addition to the identification attributes as described in section 3. The port used, and packet format (described in Section 2.3.), are the same as that for Disconnect-Request Messages.

The following attribute MAY be sent in a CoA-Request:

Filter-ID (11) - Indicates the name of a data filter list to be applied for the session that the identification attributes map to.

```
+----------+        CoA-Request          +----------+
|          |  <--------------------    |          |
|    NAS     |                              |  RADIUS  |
|          |      CoA-Response          |  Server  |
|          |    ---------------------> |          |
+----------+                          +----------+
```

The NAS responds to a CoA-Request sent by a RADIUS server with a CoA-ACK if the NAS is able to successfully change the authorizations for the user session, or a CoA-NAK if the Request is unsuccessful. A NAS MUST respond to a CoA-Request including a Service-Type Attribute with value "Authorize Only" with a CoA-NAK; a CoA-ACK MUST NOT be sent. A NAS MUST respond to a CoA-Request including a Service-Type Attribute with an unsupported value with a CoA-NAK; an Error-Cause Attribute with value "Unsupported Service" MAY be included.

2.3. Packet format

For either Disconnect-Request or CoA-Request messages UDP port 3799 is used as the destination port. For responses, the source and destination ports are reversed. Exactly one RADIUS packet is encapsulated in the UDP Data field.

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

The packet format consists of the fields: Code, Identifier, Length, Authenticator, and Attributes in Type:Length:Value (TLV) format. All fields hold the same meaning as those described in RADIUS [17].

The Authenticator field MUST be calculated in the same way as is specified for an Accounting-Request in [RFC2866].

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         Authenticator                         |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Attributes ...
+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Code

The Code field is one octet, and identifies the type of RADIUS packet.   Packets received with an invalid Code field MUST be silently discarded.

RADIUS Codes (decimal) are assigned as follows:

40	Access-Request
41	Access-Accept
42	Access-Reject

NCC Implementation Notes:

RCA does not support the following:

43	Access-Challenge
44	Status-Server (experimental)
45	Status-Client (experimental)

Identifier

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

Unlike RADIUS as defined in [RFC2865], the responsibility for retransmission of Disconnect-Request and CoA-Request messages lies with the RADIUS server. If after sending these messages, the RADIUS server does not receive a response, it will retransmit.

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

If the RADIUS server is retransmitting a Disconnect-Request or CoA-Request to the same client as before, and the Attributes have not changed, the same Request Authenticator, Identifier and source port MUST be used.   If any Attributes have changed, a new Authenticator and Identifier MUST be used.

Note that if the Event-Timestamp Attribute is included, it will be updated when the packet is retransmitted, changing the content of the Attributes field and requiring a new Identifier and Request Authenticator.

NCC Implementation Notes:

RCA does not include the Event-Timestamp Attribute in Disconnect-Request messages.

RCA does not support failing over to a secondary proxy.

The following is not implemented for RCA:

If the Request to a primary proxy fails, a secondary proxy must be queried, if available. Issues relating to failover algorithms are described in [AAATransport]. Since this represents a new request, a new Request Authenticator and Identifier MUST be used. However, where the RADIUS server is sending directly to the client, failover typically does not make sense, since Disconnect or CoA messages need to be delivered to the NAS where the sessionresides.

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 the maximum length is 4096.

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 RADIUS server and client.

Request Authenticator

In Request packets, the Authenticator value is a 16 octet MD5 [RFC1321] checksum, called the Request Authenticator. The Request Authenticator is calculated the same way as for an Accounting-Request, specified in [RFC2866].

Note that the Request Authenticator of a Disconnect or CoA-Request cannot be done the same way as the Request Authenticator of a RADIUS Access-Request, because there is no User-Password Attribute in a Disconnect-Request or CoA-Request.

Response Authenticator

The Authenticator field in a Response packet (e.g. Disconnect-ACK, Disconnect-NAK, CoA-ACK, or CoA-NAK) is called the Response Authenticator, and contains a one-way MD5 hash calculated over a stream of octets consisting of the Code, Identifier, Length, the Request Authenticator field from the 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 Response packet.

Administrative note

As noted in [RFC2865] Section 3, the secret (password shared between the client and the RADIUS server) SHOULD be at least as large and unguessable as a well-chosen password. RADIUS clients MUST use the source IP address of the RADIUS UDP packet to decide which shared secret to use, so that requests can be proxied.

Attributes

In Disconnect and CoA-Request messages, all Attributes are treated as mandatory. A NAS MUST respond to a CoA-Request containing one or more unsupported Attributes or Attribute values with a CoA-NAK; a Disconnect-Request containing one or more unsupported Attributes or Attribute values MUST be answered with a Disconnect-NAK. State changes resulting from a CoA-Request MUST be atomic: if the Request is successful, a CoA-ACK is sent, and all requested authorization changes MUST be made. If the CoA-Request is unsuccessful, a CoA-NAK MUST be sent, and the requested authorization changes MUST NOT be made. Similarly, a state change MUST NOT occur as a result of an unsuccessful Disconnect-Request; here a Disconnect-NAK MUST be sent.

Since within this specification attributes may be used for identification, authorization or other purposes, even if a NAS implements an attribute for use with RADIUS authentication and accounting, it may not support inclusion of that attribute within Disconnect-Request or CoA-Request messages, given the difference in attribute semantics. This is true even for attributes specified within [RFC2865], [RFC2868], [RFC2869] or [RFC3162] as allowable within Access-Accept messages.

As a result, attributes beyond those specified in Table of Attributes section. SHOULD NOT be included within Disconnect or CoA messages since this could produce unpredictable results.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not include attributes beyond those specified in Section 3.2.

In parameterised mode, RCA can be configured to add any attribute to a Disconnect-Request

When using a forwarding proxy, the proxy must be able to alter the packet as it passes through in each direction. When the proxy forwards a Disconnect or CoA-Request, it MAY add a Proxy-State Attribute, and when the proxy forwards a response, it MUST remove its Proxy-State Attribute if it added one. Proxy-State is always added or removed after any other Proxy-States, but no other assumptions regarding its location within the list of Attributes can be made. Since Disconnect and CoA responses are authenticated on the entire packet contents, the stripping of the Proxy-State Attribute invalidates the integrity check - so the proxy needs to recompute it. A forwarding proxy MUST NOT modify existing Proxy-State, State, or Class Attributes present in the packet.

If there are any Proxy-State Attributes in a Disconnect-Request or CoA-Request received from the server, the forwarding proxy MUST include those Proxy-State Attributes in its response to the server. The forwarding proxy MAY include the Proxy-State Attributes in the Disconnect-Request or CoA-Request when it forwards the request, or it MAY omit them in the forwarded request. If the forwarding proxy omits the Proxy-State Attributes in the request, it MUST attach them to the response before sending it to the server.

2.3.1. Data format

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

The packet format consists of the fields: Code, Identifier, Length, Authenticator, and Attributes in Type:Length:Value (TLV) format. All fields hold the same meaning as those described in RADIUS [17].

The Authenticator field MUST be calculated in the same way as is specified for an Accounting-Request in [5].

Code

The Code field is one octet, and identifies the type of RADIUS packet.   Packets received with an invalid Code field MUST be silently discarded.

RADIUS Codes (decimal) are assigned as follows:

40	Access-Request
41	Access-Accept
42	Access-Reject

NCC Implementation Note:

Unsupported in RCA

43	Access-Challenge
44	Status-Server (experimental)
45	Status-Client (experimental)

Identifier

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

Unlike RADIUS as defined in [RFC2865], the responsibility for retransmission of Disconnect-Request and CoA-Request messages lies with the RADIUS server. If after sending these messages, the RADIUS server does not receive a response, it will retransmit.

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

If the RADIUS server is retransmitting a Disconnect-Request or CoA-Request to the same client as before, and the Attributes have not changed, the same Request Authenticator, Identifier and source port MUST be used.   If any Attributes have changed, a new Authenticator and Identifier MUST be used.

Note that if the Event-Timestamp Attribute is included, it will be updated when the packet is retransmitted, changing the content of the Attributes field and requiring a new Identifier and Request Authenticator.

NCC Implementation Notes:

RCA does not include the Event-Timestamp Attribute in Disconnect-Request messages.

RCA does not support failing over to a secondary proxy.

Unsupported in RCA:

If the Request to a primary proxy fails, a secondary proxy must be queried, if available.   Issues relating to failover algorithms are described in [AAATransport]. Since this represents a new request, a new Request Authenticator and Identifier MUST be used.   However,   where the RADIUS server is sending directly to the client, failover typically does not make sense, since Disconnect or CoA messages need to be delivered to the NAS where the session resides.

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 the maximum length is 4096.

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 RADIUS server and client.

Request Authenticator

In Request packets, the Authenticator value is a 16 octet MD5 [RFC1321] checksum, called the Request Authenticator. The Request Authenticator is calculated the same way as for an Accounting-Request, specified in [5].

Note that the Request Authenticator of a Disconnect or CoA-Request cannot be done the same way as the Request Authenticator of a RADIUS Access-Request, because there is no User-Password Attribute in a Disconnect-Request or CoA-Request.

Response Authenticator

The Authenticator field in a Response packet (e.g. Disconnect-ACK, Disconnect-NAK, CoA-ACK, or CoA-NAK) is called the Response Authenticator, and contains a one-way MD5 hash calculated over a stream of octets consisting of the Code, Identifier, Length, the Request Authenticator field from the 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 Response packet.

Administrative note

As noted in [17] Section 3, the secret (password shared between the client and the RADIUS server) SHOULD be at least as large and unguessable as a well-chosen password. RADIUS clients MUST use the source IP address of the RADIUS UDP packet to decide which shared secret to use, so that requests can be proxied.

Attributes

In Disconnect and CoA-Request messages, all Attributes are treated as mandatory. A NAS MUST respond to a CoA-Request containing one or more unsupported Attributes or Attribute values with a CoA-NAK; a Disconnect-Request containing one or more unsupported Attributes or Attribute values MUST be answered with a Disconnect-NAK. State changes resulting from a CoA-Request MUST be atomic: if the Request is successful, a CoA-ACK is sent, and all requested authorization changes MUST be made. If the CoA-Request is unsuccessful, a CoA-NAK MUST be sent, and the requestedauthorization changes MUST NOT be made. Similarly, a state change MUST NOT occur as a result of an unsuccessful Disconnect-Request; here a Disconnect-NAK MUST be sent.

Since within this specification attributes may be used for identification, authorization or other purposes, even if a NAS implements an attribute for use with RADIUS authentication and accounting, it may not support inclusion of that attribute within Disconnect-Request or CoA-Request messages, given the difference in attribute semantics. This is true even for attributes specified within [RFC2865], [RFC2868], [RFC2869] or [RFC3162] as allowable within Access-Accept messages.

As a result, attributes beyond those specified in Table of Attributes section. SHOULD NOT be included within Disconnect or CoA messages since this could produce unpredictable results.

NCC Implementation Notes:

In 3GPP2 mode, RCA does not include attributes beyond those specified in Section 3.2.

In parameterised mode, RCA can be configured to add any attribute to a Disconnect-Request

When using a forwarding proxy, the proxy must be able to alter the packet as it passes through in each direction. When the proxy forwards a Disconnect or CoA-Request, it MAY add a Proxy-State Attribute, and when the proxy forwards a response, it MUST remove its Proxy-State Attribute if it added one. Proxy-State is always added or removed after any other Proxy-States, but no other assumptions regarding its location within the list of Attributes can be made. Since Disconnect and CoA responses are authenticated on the entire packet contents, the stripping of the Proxy-State Attribute invalidates the integrity check - so the proxy needs to recompute it. A forwarding proxy MUST NOT modify existing Proxy-State, State, or Class Attributes present in the packet.

If there are any Proxy-State Attributes in a Disconnect-Request or CoA-Request received from the server, the forwarding proxy MUST include those Proxy-State Attributes in its response to the server. The forwarding proxy MAY include the Proxy-State Attributes in the Disconnect-Request or CoA-Request when it forwards the request, or it MAY omit them in the forwarded request. If the forwarding proxy omits the Proxy-State Attributes in the request, it MUST attach them to the response before sending it to the server.

3. Attributes

3. Attributes

In Disconnect-Request and CoA-Request packets, certain attributes are used to uniquely identify the NAS as well as a user session on the NAS. All NAS identification attributes included in a Request message MUST match in order for a Disconnect-Request or CoA-Request to be successful; otherwise a Disconnect-NAK or CoA-NAK SHOULD be sent.

For session identification attributes, the User-Name and Acct-Session-Id Attributes, if included, MUST match in order for a Disconnect-Request or CoA-Request to be successful; other session identification attributes SHOULD match. Where a mismatch of session identification attributes is detected, a Disconnect-NAK or CoA-NAK SHOULD be sent. The ability to use NAS or session identification attributes to map to unique/multiple sessions is beyond the scope of this document.

NAS identification attributes

Attribute	#	Reference	Description
NAS-IP-Address	4	[RFC2865]	The IPv4 address of the NAS.
NAS-Identifier	32	[RFC2865]	String identifying the NAS.
NAS-IPv6-Address	95	[RFC3162]	The IPv6 address of the NAS.

NCC Implementation Notes:

- In 3GPP2 mode, RCA only adds NAS-Identifier to Disconnect-Request messages. It does NOT add NAS-IP-Address or NAS_IPv6-Address.
- In parameterised mode, RCA can be configured to add all of the above attributes to a Disconnect-Request message, provided they are present in the Access-Request message.

Identification attributes for session identification attributes, are described below.

Attribute	#	Reference	Description
User-Name	1	[RFC2865]	The name of the user associated with the session.
NAS-Port	5	[RFC2865]	The port on which the session is terminated.
Framed-IP-Address	8	[RFC2865]	The IPv4 address associated with the session.
Called-Station-Id	30	[RFC2865]	The link address to which the session is connected.
Calling-Station-Id	31	[RFC2865]	The link address from which the session is connected.
Acct-Session-Id	44	[RFC2866]	The identifier uniquely identifying the session on the NAS.
Acct-Multi-Session-Id	50	[RFC2866]	The identifier uniquely identifying related sessions.
NAS-Port-Type	61	[RFC2865]	The type of port used.
NAS-Port-Id	87	[RFC2869]	String identifying the port where the session is.
Originating-Line-Info	94	[NASREQ]	Provides information on the characteristics of the line from which a session originated.
Framed-Interface-Id	96	[RFC3162]	The IPv6 Interface Identifier associated with the session; always sent with Framed-IPv6-Prefix.
Framed-IPv6-Prefix	97	[RFC3162]	The IPv6 prefix associated with the session, always sent with Framed-Interface-Id.

NCC Implementation Notes:

- In 3GPP2 mode, the only attributes RCA sends in Disconnect-Request are NAS-Id and User-Name (as specified in this RFC) and Correlation-Id.
- In parameterised mode, RCA can be configured to add any or all of the above attributes to a Disconnect-Request message.

To address security concerns described inSection 5.1., the User-Name Attribute SHOULD be present in Disconnect-Request or CoA-Request packets; one or more additional session identification attributes MAY also be present. To address security concerns described in Impersonation, one or more of the NAS-IP-Address or NAS-IPv6-Address Attributes SHOULD be present in Disconnect-Request or CoA-Request packets; the NAS-Identifier Attribute MAY be present in addition.

If one or more authorization changes specified in a CoA-Request cannot be carried out, or if one or more attributes or attribute- values is unsupported, a CoA-NAK MUST be sent. Similarly, if there are one or more unsupported attributes or attribute values in aDisconnect-Request, a Disconnect-NAK MUST be sent.

Where a Service-Type Attribute with value "Authorize Only" is included within a CoA-Request or Disconnect-Request, attributes representing an authorization change MUST NOT be included; only identification attributes are permitted. If attributes other than NAS or session identification attributes are included in such a CoA-Request, implementations MUST send a CoA-NAK; an Error-Cause Attribute with value "Unsupported Attribute" MAY be included.

Similarly, if attributes other than NAS or session identification attributes are included in such a Disconnect-Request, implementations MUST send a Disconnect-NAK; an Error-Cause Attribute with value "Unsupported Attribute" MAY be included.

3.1. Error cause

NCC Implementation Notes:

In 3GPP2 mode, RCA does not use this attribute.

In parameterised mode, RCA can be configured to read this attribute and / or send this attribute in a Radius message, on a per message type basis.

It is possible that the NAS cannot honor Disconnect-Request or CoA-Request messages for some reason. The Error-Cause Attribute provides more detail on the cause of the problem. It MAY be included within Disconnect-ACK, Disconnect-NAK and CoA-NAK messages.

A summary of the Error-Cause Attribute format is shown below. 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     |             Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Value (cont)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	101 for Error-Cause
Length	6

Value	The Value field is four octets, containing an integer specifying the cause of the error. Values 0-199 and 300-399 are reserved.

Values 200-299 represent successful completion, so that these values may only be sent within Disconnect-ACK or CoA-ACK message and MUST NOT be sent within a Disconnect-NAK or CoA-NAK.

Values 400-499 represent fatal errors committed by the RADIUS server, so that they MAY be sent within CoA-NAK or Disconnect-NAK messages, and MUST NOT be sent within CoA-ACK or Disconnect-ACK messages.

Values 500-599 represent fatal errors occurring on a NAS or RADIUS proxy, so that they MAY be sent within CoA-NAK and Disconnect-NAK messages, and MUST NOT be sent within CoA-ACK or Disconnect-ACK messages.

Error-Cause values SHOULD be logged by the RADIUS server.

Error-Code values (expressed in decimal) include:

#	Value
201	Residual Session Context Removed
202	Invalid EAP Packet (Ignored)
401	Unsupported Attribute
402	Missing Attribute
403	NAS Identification Mismatch
404	Invalid Request
405	Unsupported Service
406	Unsupported Extension
501	Administratively Prohibited
502	Request Not Routable (Proxy)
503	Session Context Not Found
504	Session Context Not Removable
505	Other Proxy Processing Error
506	Resources Unavailable
507	Request Initiated

"Residual Session Context Removed" is sent in response to a Disconnect-Request if the user session is no longer active, but residual session context was found and successfully removed. This value is only sent within a Disconnect-ACK and MUST NOT be sent within a CoA-ACK, Disconnect-NAK or CoA-NAK.

"Invalid EAP Packet (Ignored)" is a non-fatal error that MUST NOT be sent by implementations of this specification.

"Unsupported Attribute" is a fatal error sent if a Request contains an attribute (such as a Vendor-Specific or EAP-Message Attribute) that is not supported.

"Missing Attribute" is a fatal error sent if critical attributes (such as NAS or session identification attributes) are missing from a Request.

"NAS Identification Mismatch" is a fatal error sent if one or more NAS identification attributes do not match the identity of the NAS receiving the Request.

"Invalid Request" is a fatal error sent if some other aspect of the Request is invalid, such as if one or more attributes (such as EAP-Message Attribute(s)) are not formatted properly.

"Unsupported Service" is a fatal error sent if a Service-Type Attribute included with the Request is sent with an invalid or unsupported value.

"Unsupported Extension" is a fatal error sent due to lack of support for an extension such as Disconnect and/or CoA messages. This will typically be sent by a proxy receiving an ICMP port unreachable message after attempting to forward a Request to the NAS.

"Administratively Prohibited" is a fatal error sent if the NAS is configured to prohibit honoring of Request messages for the specified session.

"Request Not Routable (Proxy)" is a A fatal error which MAY be sent by a RADIUS proxy and MUST NOT be sent by a NAS. It indicates that the RADIUS proxy was unable to determine how to route the Request to the NAS. For example, this can occur if the required entries are not present in the proxy's realm routing table.

"Session Context Not Found" is a fatal error sent if the session context identified in the Request does not exist on the NAS.

"Session Context Not Removable" is a fatal error sent in response to a Disconnect-Request if the NAS was able to locate the session context, but could not remove it for some reason. It MUST NOT be sent within a CoA-ACK, CoA-NAK or Disconnect-ACK, only within a Disconnect-NAK.

"Other Proxy Processing Error" is a fatal error sent in response to a Request that could not be processed by a proxy, for reasons other than routing.

"Resources Unavailable" is a fatal error sent when a Request could not be honored due to lack of available NAS resources (memory, non-volatile storage, etc.).

"Request Initiated" is a A fatal error sent in response to a Request including a Service-Type Attribute with a value of "Authorize Only". It indicates that the Disconnect-Request or CoA-Request has not been honored, but that a RADIUS Access-Request including a Service-Type Attribute with value "Authorize Only" is being sent to the RADIUS server.

3.2. Table of Attributes

The following table provides a guide to which attributes may be found in which packets, and in what quantity.

NCC Implementation Notes:

The details listed in the following table are not supported:

Request	ACK	NAK	#	Attribute
0-1	0	0	1	User-Name [Note 1]
0-1	0	0	4	NAS-IP-Address [Note 1]
0-1	0	0	5	NAS-Port [Note 1]
0-1	0	0-1	6	Service-Type [Note 6]
0-1	0	0	7	Framed-Protocol [Note 3]
0-1	0	0	8	Framed-IP-Address [Note 1]

Request	ACK	NAK	#	Attribute
0-1	0	0	9	Framed-IP-Netmask [Note 3]
0-1	0	0	10	Framed-Routing [Note 3]
0+	0	0	11	Filter-ID [Note 3]
0-1	0	0	12	Framed-MTU [Note 3]
0+	0	0	13	Framed-Compression [Note 3]
0+	0	0	14	Login-IP-Host [Note 3]
0-1	0	0	15	Login-Service [Note 3]
0-1	0	0	16	Login-TCP-Port [Note 3]
0+	0	0	18	Reply-Message [Note 2]
0-1	0	0	19	Callback-Number [Note 3]
0-1	0	0	20	Callback-Id [Note 3]
0+	0	0	22	Framed-Route [Note 3]
0-1	0	0	23	Framed-IPX-Network [Note 3]
0-1	0-1	0-1	24	State [Note 7]
0+	0	0	25	Class [Note 3]
0+	0	0	26	Vendor-Specific [Note 3]
0-1	0	0	27	Session-Timeout [Note 3]
0-1	0	0	28	Idle-Timeout [Note 3]
0-1	0	0	29	Termination-Action [Note 3]
0-1	0	0	30	Called-Station-Id [Note 1]
0-1	0	0	31	Calling-Station-Id [Note 1]
0-1	0	0	32	NAS-Identifier [Note 1]
0+	0+	0+	33	Proxy-State
0-1	0	0	34	Login-LAT-Service [Note 3]
0-1	0	0	35	Login-LAT-Node [Note 3]
0-1	0	0	36	Login-LAT-Group [Note 3]
0-1	0	0	37	Framed-AppleTalk-Link [Note 3]
0+	0	0	38	Framed-AppleTalk-Network [Note 3]
0-1	0	0	39	Framed-AppleTalk-Zone [Note 3]
0-1	0	0	44	Acct-Session-Id [Note 1]
0-1	0	0	50	Acct-Multi-Session-Id [Note 1]
0-1	0-1	0-1	55	Event-Timestamp
0-1	0	0	61	NAS-Port-Type [Note 1]
0-1	0	0	62	Port-Limit [Note 3]
0-1	0	0	63	Login-LAT-Port [Note 3]
0+	0	0	64	Tunnel-Type [Note 5]
0+	0	0	65	Tunnel-Medium-Type [Note 5]
0+	0	0	66	Tunnel-Client-Endpoint [Note 5]
0+	0	0	67	Tunnel-Server-Endpoint [Note 5]
0+	0	0	69	Tunnel-Password [Note 5]
0-1	0	0	71	ARAP-Features [Note 3]
0-1	0	0	72	ARAP-Zone-Access [Note 3]

Request	ACK	NAK	#	Attribute
0+	0	0	78	Configuration-Token [Note 3]
0+	0-1	0	79	EAP-Message [Note 2]
0-1	0-1	0-1	80	Message-Authenticator
0+	0	0	81	Tunnel-Private-Group-ID [Note 5]
0+	0	0	82	Tunnel-Assignment-ID [Note 5]
0+	0	0	83	Tunnel-Preference [Note 5]
0-1	0	0	85	Acct-Interim-Interval [Note 3]
0-1	0	0	87	NAS-Port-Id [Note 1]
0-1	0	0	88	Framed-Pool [Note 3]
0+	0	0	90	Tunnel-Client-Auth-ID [Note 5]
0+	0	0	91	Tunnel-Server-Auth-ID [Note 5]
0-1	0	0	94	Originating-Line-Info [Note 1]
0-1	0	0	95	NAS-IPv6-Address [Note 1]
0-1	0	0	96	Framed-Interface-Id [Note 1]
0+	0	0	97	Framed-IPv6-Prefix [Note 1]
0+	0	0	98	Login-IPv6-Host [Note 3]
0+	0	0	99	Framed-IPv6-Route [Note 3]
0-1	0	0	100	Framed-IPv6-Pool [Note 3]
0	0	0+	101	Error-Cause

Disconnect Messages

Request	ACK	NAK	#	Attribute
0-1	0	0	1	User-Name [Note 1]
0-1	0	0	4	NAS-IP-Address [Note 1]
0-1	0	0	5	NAS-Port [Note 1]
0-1	0	0-1	6	Service-Type [Note 6]
0-1	0	0	8	Framed-IP-Address [Note 1]
0+ (not supported) 0-1 (non-standard implementation)	0	0	18	Reply-Message [Note 2]
0-1	0-1	0-1	24	State [Note 7]
0+	0	0	25	Class [Note 4]
0+	0	0	26	Vendor-Specific
0-1	0	0	30	Called-Station-Id [Note 1]
0-1	0	0	31	Calling-Station-Id [Note 1]
0-1	0	0	32	NAS-Identifier [Note 1]

Request	ACK	NAK	#	Attribute
0+ (not supported) 0-1 (non-standard implementation)	0+ (not supported) 0-1 (non-standard implementation)	0+ (not supported) 0-1 (non-standard implementation)	33	Proxy-State
0-1	0	0	44	Acct-Session-Id [Note 1]
0-1	0-1	0	49	Acct-Terminate-Cause
0-1	0	0	50	Acct-Multi-Session-Id [Note 1]
0-1	0-1	0-1	55	Event-Timestamp
0-1	0	0	61	NAS-Port-Type [Note 1]
0+ (not supported) 0-1 (non-standard implementation)	0-1	0	79	EAP-Message [Note 2]
0-1	0-1	0-1	80	Message-Authenticator
0-1	0	0	87	NAS-Port-Id [Note 1]
0-1	0	0	94	Originating-Line-Info [Note 1]
0-1	0	0	95	NAS-IPv6-Address [Note 1]
0-1	0	0	96	Framed-Interface-Id [Note 1]
0+ (not supported) 0-1 (non-standard implementation)	0	0	97	Framed-IPv6-Prefix [Note 1]
0	0+ (not supported) 0-1 (non-standard implementation)	0+ (not supported) 0-1 (non-standard implementation)	101	Error-Cause

[Note 1] Where NAS or session identification attributes are included in Disconnect-Request or CoA-Request messages, they are used for identification purposes only. These attributes MUST NOT be used for purposes other than identification (e.g. within CoA-Request messages to request authorization changes).

[Note 2] The Reply-Message Attribute is used to present a displayable message to the user. The message is only displayed as a result of a successful Disconnect-Request or CoA-Request (where a Disconnect-ACK or CoA-ACK is subsequently sent). Where EAP is used for authentication, an EAP-Message/Notification-Request Attribute is sent instead, and Disconnect-ACK or CoA-ACK messages contain an EAP- Message/Notification-Response Attribute.

[Note 3] When included within a CoA-Request, these attributes represent an authorization change request. When one of these attributes is omitted from a CoA-Request, the NAS assumes that the attribute value is to remain unchanged. Attributes included in a CoA-Request replace all existing value(s) of the same attribute(s).

[Note 4] When included within a successful Disconnect-Request (where a Disconnect-ACK is subsequently sent), the Class Attribute SHOULD be sent unmodified by the client to the accounting server in the Accounting Stop packet. If the Disconnect-Request is unsuccessful, then the Class Attribute is not processed.

[Note 5] When included within a CoA-Request, these attributes represent an authorization change request. Where tunnel attribute(s) are sent within a successful CoA-Request, all existing tunnel attributes are removed and replaced by the new attribute(s).

[Note 6] When included within a Disconnect-Request or CoA-Request, a Service-Type Attribute with value "Authorize Only" indicates that the Request only contains NAS and session identification attributes, and that the NAS should attempt reauthorization by sending an Access- Request with a Service-Type Attribute with value "Authorize Only". This enables a usage model akin to that supported in Diameter, thus easing translation between the two protocols. Support for the Service-Type Attribute is optional within CoA-Request and Disconnect-Request messages; where it is not included, the Request message may contain both identification and authorization attributes. A NAS that does not support the Service-Type Attribute with the value "Authorize Only" within a Disconnect-Request MUST respond with a Disconnect-NAK including no Service-Type Attribute; an Error-Cause Attribute with value "Unsupported Service" MAY be included. A NAS that does not support the Service-Type Attribute with the value "Authorize Only" within a CoA-Request MUST respond with a CoA-NAK including no Service-Type Attribute; an Error-Cause Attribute with value "Unsupported Service" MAY be included.

A NAS supporting the "Authorize Only" Service-Type value within Disconnect-Request or CoA-Request messages MUST respond with a Disconnect-NAK or CoA-NAK respectively, containing a Service-Type Attribute with value "Authorize Only", and an Error-Cause Attribute with value "Request Initiated". The NAS then sends an Access-Request to the RADIUS server with a Service-Type Attribute with value "Authorize Only". This Access-Request SHOULD contain the NAS attributes from the Disconnect or CoA-Request, as well as the session attributes from the Request legal for inclusion in an Access-Request as specified in [RFC2865], [RFC2868], [RFC2869] and [RFC3162]. As noted in [RFC2869] Section 5.19, a Message-Authenticator attribute SHOULD be included in an Access-Request that does not contain a User-Password, CHAP-Password, ARAP-Password or EAP-Message Attribute. The RADIUS server should send back an Access-Accept to (re-)authorize the session or an Access-Reject to refuse to (re-)authorize it.

[Note 7] The State Attribute is available to be sent by the RADIUS server to the NAS in a Disconnect-Request or CoA-Request message and MUST be sent unmodified from the NAS to the RADIUS server in a subsequent ACK or NAK message. If a Service-Type Attribute with value "Authorize Only" is included in a Disconnect-Request or CoA- Request along with a State Attribute, then the State Attribute MUST be sent unmodified from the NAS to the RADIUS server in the resulting Access-Request sent to the RADIUS server, if any. The State Attribute is also available to be sent by the RADIUS server to the NAS in a CoA-Request that also includes a Termination-Action Attribute with the value of RADIUS-Request. If the client performs the Termination-Action by sending a new Access-Request upon termination of the current session, it MUST include the State

Attribute unchanged in that Access-Request. In either usage, the client MUST NOT interpret the Attribute locally. A Disconnect- Request or CoA-Request packet must have only zero or one State Attribute. Usage of the State Attribute is implementation dependent. If the RADIUS server does not recognize the State Attribute in the Access-Request, then it MUST send an Access-Reject.

The following table defines the meaning of 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.

4. IANA Considerations

This document uses the RADIUS [RFC2865] namespace, see <http://www.iana.org/assignments/radius-types>. There are six updates for the section: RADIUS Packet Type Codes. These Packet Types are allocated in [RADIANA]:

40 - Disconnect-Request

41 - Disconnect-ACK

42 - Disconnect-NAK

43 - CoA-Request

44 - CoA-ACK

45 - CoA-NAK

Allocation of a new Service-Type value for "Authorize Only" is requested. This document also uses the UDP [RFC768] namespace, see <http://www.iana.org/assignments/port-numbers>. The authors request a port assignment from the Registered ports range. Finally, this specification allocates the Error-Cause Attribute (101) with the following decimal values:

#	Value
201	Residual Session Context Removed
202	Invalid EAP Packet (Ignored)
401	Unsupported Attribute
402	Missing Attribute
403	NAS Identification Mismatch
404	Invalid Request
405	Unsupported Service
406	Unsupported Extension
501	Administratively Prohibited
502	Request Not Routable (Proxy)
503	Session Context Not Found
504	Session Context Not Removable
505	Other Proxy Processing Error
506	Resources Unavailable
507	Request Initiated

5. Security Considerations

5.1. Authorization issues

Where a NAS is shared by multiple providers, it is undesirable for one provider to be able to send Disconnect-Request or CoA-Requests affecting the sessions of another provider.

A NAS or RADIUS proxy MUST silently discard Disconnect-Request or CoA-Request messages from untrusted sources. By default, a RADIUS proxy SHOULD perform a "reverse path forwarding" (RPF) check to verify that a Disconnect-Request or CoA-Request originates from an authorized RADIUS server. In addition, it SHOULD be possible to explicitly authorize additional sources of Disconnect-Request or CoA-Request packets relating to certain classes of sessions. For example, a particular source can be explicitly authorized to send CoA-Request messages relating to users within a set of realms.

To perform the RPF check, the proxy uses the session identification attributes included in Disconnect-Request or CoA-Request messages, in order to determine the RADIUS server(s) to which an equivalent Access-Request could be routed. If the source address of the Disconnect-Request or CoA-Request is within this set, then the Request is forwarded; otherwise it MUST be silently discarded.

Typically the proxy will extract the realm from the Network Access Identifier [RFC2486] included within the User-Name Attribute, and determine the corresponding RADIUS servers in the proxy routing tables. The RADIUS servers for that realm are then compared against the source address of the packet. Where no RADIUS proxy is present, the RPF check will need to be performed by the NAS itself.

Since authorization to send a Disconnect-Request or CoA-Request is determined based on the source address and the corresponding shared secret, the NASes or proxies SHOULD configure a different shared secret for each RADIUS server.

5.2. Impersonation

[RFC2865] Section 3 states:

> A RADIUS server MUST use the source IP address of the RADIUS UDP packet to decide which shared secret to use, so that RADIUS requests can be proxied.

When RADIUS requests are forwarded by a proxy, the NAS-IP-Address or NAS-IPv6-Address Attributes will typically not match the source address observed by the RADIUS server. Since the NAS-Identifier Attribute need not contain an FQDN, this attribute may not be resolvable to the source address observed by the RADIUS server, even when no proxy is present.

As a result, the authenticity check performed by a RADIUS server or proxy does not verify the correctness of NAS identification attributes. This makes it possible for a rogue NAS to forge NAS-IP-Address, NAS-IPv6-Address or NAS-Identifier Attributes within a RADIUS Access-Request in order to impersonate another NAS. It is also possible for a rogue NAS to forge session identification attributes such as the Called-Station-Id, Calling-Station-Id, or Originating-Line-Info [NASREQ]. This could fool the RADIUS server into sending Disconnect-Request or CoA-Request messages containing forged session identification attributes to a NAS targeted by an attacker.

To address these vulnerabilities RADIUS proxies SHOULD check whether NAS identification attributes (see Section 3.) match the source address of packets originating from the NAS. Where one or more attributes do not match, Disconnect-Request or CoA-Request messages SHOULD be silently discarded.

Such a check may not always be possible. Since the NAS-Identifier Attribute need not correspond to an FQDN, it may not be resolvable to an IP address to be matched against the source address. Also, where a NAT exists between the RADIUS client and proxy, checking the NAS-IP-Address or NAS-IPv6-Address Attributes may not be feasible.

5.3. IPsec Usage Guidelines

NCC Implementation Notes:

There is no support for IPsec in RCA. However, IPsec can be implemented by routers between the RADIUS server and the RADIUS client so this should not be a problem in practice.

RCA does not support the following:

In addition to security vulnerabilities unique to Disconnect or CoA messages, the protocol exchanges described in this document are susceptible to the same vulnerabilities as RADIUS [RFC2865]. It is RECOMMENDED that IPsec be employed to afford better security.

Implementations of this specification SHOULD support IPsec [RFC2401] along with IKE [RFC2409] for key management. IPsec ESP [RFC2406] with a non-null transform SHOULD be supported, and IPsec ESP with a non-null encryption transform and authentication support SHOULD be used to provide per-packet confidentiality, authentication, integrity and replay protection. IKE SHOULD be used for key management.

Within RADIUS [RFC2865], a shared secret is used for hiding Attributes such as User-Password, as well as used in computation of the Response Authenticator. In RADIUS accounting [RFC2866], the shared secret is used in computation of both the Request Authenticator and the Response Authenticator.

Since in RADIUS a shared secret is used to provide confidentiality as well as integrity protection and authentication, only use of IPsec ESP with a non-null transform can provide security services sufficient to substitute for RADIUS application-layer security.

Therefore, where IPsec AH or ESP null is used, it will typically still be necessary to configure a RADIUS shared secret.

Where RADIUS is run over IPsec ESP with a non-null transform, the secret shared between the NAS and the RADIUS server MAY NOT be configured. In this case, a shared secret of zero length MUST be assumed. However, a RADIUS server that cannot know whether incoming traffic is IPsec-protected MUST be configured with a non-null RADIUS shared secret.

When IPsec ESP is used with RADIUS, per-packet authentication, integrity and replay protection MUST be used.

3DES-CBC MUST be supported as an encryption transform and AES-CBC SHOULD be supported.

AES-CBC SHOULD be offered as a preferred encryption transform if supported. HMAC-SHA1-96 MUST be supported as an authentication transform. DES-CBC SHOULD NOT be used as the encryption transform.

A typical IPsec policy for an IPsec-capable RADIUS client is "Initiate IPsec, from me to any destination port UDP 1812". This IPsec policy causes an IPsec SA to be set up by the RADIUS client prior to sending RADIUS traffic. If some RADIUS servers contacted by the client do not support IPsec, then a more granular policy will be required: "Initiate IPsec, from me to IPsec-Capable-RADIUS-Server, destination port UDP 1812."

For a client implementing this specification, the policy would be "Accept IPsec, from any to me, destination port UDP 3799". This causes the RADIUS client to accept (but not require) use of IPsec.

It may not be appropriate to require IPsec for all RADIUS servers connecting to an IPsec-enabled RADIUS client, since some RADIUS servers may not support IPsec.

For an IPsec-capable RADIUS server, a typical IPsec policy is "Accept IPsec, from any to me, destination port 1812". This causes the RADIUS server to accept (but not require) use of IPsec. It may not be appropriate to require IPsec for all RADIUS clients connecting to an IPsec-enabled RADIUS server, since some RADIUS clients may not support IPsec.

For servers implementing this specification, the policy would be "Initiate IPsec, from me to any, destination port UDP 3799". This causes the RADIUS server to initiate IPsec when sending RADIUS extension traffic to any RADIUS client. If some RADIUS clients contacted by the server do not support IPsec, then a more granular policy will be required, such as "Initiate IPsec, from me to IPsec-capable-RADIUS-client, destination port UDP 3799".

Where IPsec is used for security, and no RADIUS shared secret is configured, it is important that the RADIUS client and server perform an authorization check. Before enabling a host to act as a RADIUS client, the RADIUS server SHOULD check whether the host is authorized to provide network access. Similarly, before enabling a host to act as a RADIUS server, the RADIUS client SHOULD check whether the host is authorized for that role.

RADIUS servers can be configured with the IP addresses (for IKE Aggressive Mode with pre-shared keys) or FQDNs (for certificate authentication) of RADIUS clients. Alternatively, if a separate Certification Authority (CA) exists for RADIUS clients, then the RADIUS server can configure this CA as a trust anchor [RFC3280] for use with IPsec.

Similarly, RADIUS clients can be configured with the IP addresses (for IKE Aggressive Mode with pre-shared keys) or FQDNs (for certificate authentication) of RADIUS servers. Alternatively, if a separate CA exists for RADIUS servers, then the RADIUS client can configure this CA as a trust anchor for use with IPsec.

Since unlike SSL/TLS, IKE does not permit certificate policies to be set on a per-port basis, certificate policies need to apply to all uses of IPsec on RADIUS clients and servers. In IPsec deployment supporting only certificate authentication, a management station initiating an IPsec-protected telnet session to the RADIUS server would need to obtain a certificate chaining to the RADIUS client CA.

Issuing such a certificate might not be appropriate if the management station was not authorized as a RADIUS client.

Where RADIUS clients may obtain their IP address dynamically (such as an Access Point supporting DHCP), Main Mode with pre-shared keys [RFC2409] SHOULD NOT be used, since this requires use of a group pre-shared key; instead, Aggressive Mode SHOULD be used. Where RADIUS client addresses are statically assigned, either Aggressive Mode or Main Mode MAY be used. With certificate authentication, Main Mode SHOULD be used.

Care needs to be taken with IKE Phase 1 Identity Payload selection in order to enable mapping of identities to pre-shared keys, even with Aggressive Mode. Where the ID_IPV4_ADDR or ID_IPV6_ADDR Identity Payloads are used and addresses are dynamically assigned, mapping of identities to keys is not possible, so that group pre-shared keys are still a practical necessity. As a result, the ID_FQDN identity payload SHOULD be employed in situations where Aggressive mode is utilized along with pre-shared keys and IP addresses are dynamically assigned. This approach also has other advantages, since it allows the RADIUS server and client to configure themselves based on the fully qualified domain name of their peers.

Note that with IPsec, security services are negotiated at the granularity of an IPsec SA, so that RADIUS exchanges requiring a set of security services different from those negotiated with existing IPsec SAs will need to negotiate a new IPsec SA. Separate IPsec SAs are also advisable where quality of service considerations dictate different handling RADIUS conversations. Attempting to apply different quality of service to connections handled by the same IPsec SA can result in reordering, and falling outside the replay window.

For a discussion of the issues, see [RFC2983].

5.4. Replay protection

NCC Implementation Notes:

The following is not supported for RCA:

- Where IPsec replay protection is not used, the Event-Timestamp (55) Attribute [RFC2869] SHOULD be included within all messages. When this attribute is present, both the NAS and the RADIUS server MUST check that the Event-Timestamp Attribute is current within an acceptable time window. If the Event-Timestamp Attribute is not current, then the message MUST be silently discarded. This implies the need for time synchronization within the network, which can be achieved by a variety of means, including secure NTP, as described in [NTPAUTH].
- Both the NAS and the RADIUS server SHOULD be configurable to silently discard messages lacking an Event-Timestamp Attribute. A default time window of 300 seconds is recommended.

6. Example Traces

Disconnect Request with User-Name:

```
 0: xxxx xxxx xxxx xxxx xxxx 2801 001c 1b23    .B.....$.-(....#
16: 624c 3543 ceba 55f1 be55 a714 ca5e 0108    bL5C..U..U...^..
32: 6d63 6869 6261
```

Disconnect Request with Acct-Session-ID:

```
 0: xxxx xxxx xxxx xxxx xxxx 2801 001e ad0d    .B..... ~.(.....
16: 8e53 55b6 bd02 a0cb ace6 4e38 77bd 2c0a    .SU.......N8w.,.
32: 3930 3233 3435 3637                        90234567
```

Disconnect Request with Framed-IP-Address:

```
 0: xxxx xxxx xxxx xxxx xxxx 2801 001a 0bda    .B....."2.(.....
16: 33fe 765b 05f0 fd9c c32a 2f6b 5182 0806    3.v[.....*/kQ...
32: 0a00 0203
```

7. References

7.1. Normative References

[RFC1305] Mills, D., "Network Time Protocol (version 3) Specification, Implementation and Analysis", RFC 1305, March 1992.

[RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[RFC2104] Krawczyk, H., Bellare, M. and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[RFC2406] Kent, S. and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

[RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[RFC2486] Aboba, B. and M. Beadles, "The Network Access Identifier", RFC 2486, January 1999.

[RFC2865] Rigney, C., Willens, S., Rubens, A. and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

[RFC2866] Rigney, C., "RADIUS Accounting", RFC 2866, June 2000.

[RFC2869] Rigney, C., Willats, W. and P. Calhoun, "RADIUS Extensions", RFC 2869, June 2000.

[RFC3162] Aboba, B., Zorn, G. and D. Mitton, "RADIUS and IPv6", RFC 3162, August 2001.

[RFC3280] Housley, R., Polk, W., Ford, W. and D. Solo, "Internet X.509 Public Key Infrastructure Certificate andCertificate Revocation List (CRL) Profile", RFC 3280, April 2002.

[RADIANA] Aboba, B., "IANA Considerations for RADIUS (Remote Authentication Dial In User Service)", RFC 3575, July 2003.

7.2. Informative References

[RFC2882]	Mitton, D., "Network Access Server Requirements: Extended RADIUS Practices", RFC 2882, July 2000.
[RFC2983]	Black, D. "Differentiated Services and Tunnels", RFC 2983, October 2000.
[AAATransport]	Aboba, B. and J. Wood, "Authentication, Authorization and Accounting (AAA) Transport Profile", RFC 3539, June 2003.
[Diameter]	Calhoun, P., et al., "Diameter Base Protocol", Work in Progress.
[MD5Attack]	Dobbertin, H., "The Status of MD5 After a Recent Attack", CryptoBytes Vol.2 No.2, Summer 1996.
[NASREQ]	Calhoun, P., et al., "Diameter Network Access Server Application", Work in Progress.
[NTPAUTH]	Mills, D., "Public Key Cryptography for the Network Time Protocol", Work in Progress.

8. Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards- related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

9. Acknowledgments

This protocol was first developed and distributed by Ascend Communications. Example code was distributed in their free server kit.

The authors would like to acknowledge the valuable suggestions and feedback from the following people:

Avi Lior <avi@bridgewatersystems.com>,
Randy Bush <randy@psg.net>,
Steve Bellovin <smb@research.att.com>
Glen Zorn <gwz@cisco.com>,
Mark Jones <mjones@bridgewatersystems.com>,
Claudio Lapidus <clapidus@hotmail.com>,
Anurag Batta <Anurag_Batta@3com.com>,
Kuntal Chowdhury <chowdury@nortelnetworks.com>, and
Tim Moore <timmoore@microsoft.com>.
Russ Housley <housley@vigilsec.com>

10. Authors' Addresses

Murtaza Chiba
Cisco Systems, Inc.
170 West Tasman Dr.
San Jose CA, 95134

EMail: mchiba@cisco.com
Phone: +1 408 525 7198

Gopal Dommety
Cisco Systems, Inc.
170 West Tasman Dr.
San Jose, CA 95134

EMail: gdommety@cisco.com
Phone: +1 408 525 1404

Mark Eklund
Cisco Systems, Inc.
170 West Tasman Dr.
San Jose, CA 95134

EMail: meklund@cisco.com
Phone: +1 865 671 6255

David Mitton
Circular Logic UnLtd.
733 Turnpike Street #154
North Andover, MA 01845

EMail: david@mitton.com
Phone: +1 978 683 1814

Bernard Aboba
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

EMail: bernarda@microsoft.com
Phone: +1 425 706 6605
Fax: +1 425 936 7329

11. Full Copyright Statement

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

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

Chapter 5

Miscellaneous Compliance

Overview

Introduction

This chapter describes the Oracle Communications Network Charging and Control (NCC) compliance to several RFC protocols not detailed above.

In this chapter

This chapter contains the following topics.

Compliance to RFC 2548 (Microsoft Vendor-specific RADIUS Attributes)

In parameterized mode, RCA does not support these attributes.

In parameterised mode, RCA can be configured to read any of these attributes and / or send any of these attribute in a Radius message, on a per message type basis.

Compliance to 3GPP TS 29.061

3GPP TS 29.061

This section states RCA’s compliance to this document:

3GPP TS 29.061 V3.14.1 (2005-06)
Technical Specification
3rd Generation Partnership Project;
Technical Specification Group Core Network and Terminals;
Interworking between the Public Land Mobile Network (PLMN)
supporting packet based services and
Packet Data Networks (PDN)
(Release 1999)
GLOBAL SYSTEM FOR
MOBILE COMMUNICATIONS

This document is not reproduced in full here, with differences, as is done for the RFCs, for copyright reasons.

The use of the RADIUS protocol is described in section 16 of 3GPP TS 29.061.

NCC Implementation Note:

In 3GPP2 mode, RCA does not comply with this document.

In parameterised mode, RCA can be configured to comply with this section, apart from the following comments and observations:

NCC Implementation Note:

- RCA does not perform authentication, other than identifying the subscriber and performing credit control.
- RCA does not act as a proxy.
- RCA does not use the DHCP protocol and cannot assign IP addresses.
- RCA assumes that there is only one PDP context per session. I.e. it cannot cope with receiving one Access-Request and then several strams of Accounting-Requests, one for each PDP context, as described in section 16.3.1, figure 2.
- RCA does not take any special notice of the "Session stop indicator" VSA. (RCA assumes that, as there is only one PDP per session then the whole session is over when Accounting-Request (STOP) is received.)
- RCA does not deal with Accounting ON and Accounting OFF messages.
- All the attributes listed in section 16.4.1 can be read by RCA and saved for use in RADIUS messages sent by RCA. The following attributes are of additional note:
- User-Name, 3GPP-IMSI, Calling-Station-ID RCA will probably be configured to use one of these attributes to identify the subscriber.
- Frames—IP-Address RCA does not assign IP addresses so it is not sensible to configure RCA to put this attribute in Access-Accept messages.
- RCA will usually be configured to perform correlation of sessions based on a combination of 3GPP-Charging-ID and 3GPP-GGSN-Address.

Compliance to 3GPP2 X.S0011-005-C (cdma2000 Wireless IP Network Standard: Accounting Services and 3GPP2 RADIUS VSAs)

3GPP2 X.S0011-005-C

RCA only complies with this document when run in 3GPP2 mode.

NCC Implementation Notes:

Section 3.1.2, shown below, is not implemented in RCA:

- Section 3.1.2 "It is the responsibility of the Visited RADIUS server to ensure the remote address table indices returned in a RADIUS Access-Accept message are consistent with the tables stored in the PDSN. For example, the Visited RADIUS server may filter out the Remote Address Table Index attributes contained in the RADIUS Access-Accept messages received from uncoordinated realms." RCA does not do this.

In addition:

- RCA does not produce any usage data records. It only performs credit control.
- In RCA in 3GPP2 mode, Accounting-request messages do not affect credit control in any way.

Section 5 3GPP2VSA Table: Not all the attributes for 3GPP2VSA are supported by RCA. The RCA ignores any unexpected attributes that it receives.

This table lists the attributes that are supported by RCA:

VSA	Type	Access-Request	Access-Accept	Accounting Start	Accounting Stop	Accounting Interim-Update
Correlation ID	26/44	1	0-1 (not implemented or non-standard in RCA)	1	1	1
Service Option Profile	26/74	0	0-1 (Note 1)	0	0	0
Session Termination Capability [Note 2]	26/88	1	1	0	0	0
PrePaidAccounting Quota (PPAQ) [Note 3]	26/90	0-1 (not implemented or non-standard in RCA)	0-1	0	0	0
PrePaidAccounting Capability (PPAC) [Note 4]	26/91	0-1	0-1	0	0	0

This table lists the attributes that RCA does not expect, or that RCA does not send, and that are not supported:

VSA	Type	Access-Request	Access-Accept	Accounting Start	Accounting Stop	Accounting Interim-Update
IKE Pre-shared Secret Request	26/01	0-1	0	0	0	0
Security Level	26/02	0	0-1	0	0	0
Pre-shared Secret	26/03	0	0-1	0	0	0
Reverse Tunnel Specification	26/04	0	0-1	0	0	0
Differentiated Services Class Option	26/05	0	0-1	0	0	0
Container	26/06	0	0	0	0+	0+
Home Agent	26/07	0-1	0-1	0-1	0-1	0-1
KeyID	26/08	0	0-1	0	0	0
Serving PCF	26/09	0	0	1	1	1
BSID	26/10	0	0	1	1	1
User Zone	26/11	0	0	0-1	0-1	0-1
Forward Mux Option	26/12	0	0	0-1	0-1	0-1

VSA	Type	Access-Request	Access-Accept	Accounting Start	Accounting Stop	Accounting Interim-Update
Reverse Mux Option	26/13	0	0	0-1	0-1	0-1
Service Option	26/16	0-1	0	1	1	1
Forward Traffic Type	26/17	0	0	0-1	0-1	0-1
Reverse Traffic Type	26/18	0	0	0-1	0-1	0-1
Fundamental Frame Size	26/19	0	0	0-1	0-1	0-1
Forward Fundamental RC	26/20	0	0	0-1	0-1	0-1
Reverse Fundamental RC	26/21	0	0	0-1	0-1	0-1
IP Technology	26/22	0-1	0	1	1	1
Compulsory Tunnel Indicator	26/23	0	0-1	1	1	1
Release Indicator	26/24	0	0	0	1	0
Bad PPP Frame Count	26/25	0	0	0	0-1	0-1
Number of Active Transitions	26/30	0	0	0	1	1
SDB Octet Count	26/31	0	0	0	0-1	0-1 14 See X.S0011-006-C for attributes of PrePaid Accounting (Terminating)
SDB Octet Count (Originating)	26/32	0	0	0	0-1	0-1
Number of SDBs (Terminating)	26/33	0	0	0	0-1	0-1
Number of SDBs (Originating)	26/34	0	0	0	0-1	0-1
IP Quality of Service	26/36	0	0	0-1	0-1	0-1
Airlink Priority15	26/39	0	0	0-1	0-1	0-1
Airlink Record Type15	26/40	0	0	0	0	0
Airlink Sequence Number15	26/42	0	0	0	0	0

VSA	Type	Access-Request	Access-Accept	Accounting Start	Accounting Stop	Accounting Interim-Update
Number of HDLC layer bytes received	26/43	0	0	0	0-1	0-1
Mobile Originated / Mobile Terminated Indicator15	26/45	0	0	0	0	0
Inbound Mobile IP Signaling Octet Count	26/46	0	0	0	0-1	0-1
Outbound Mobile IP Signaling Octet Count	26/47	0	0	0	0-1	0-1
Session Continue	26/48	0	0	0	1	0-1
Active Time	26/49	0	0	0	0-1	0-1
DCCH Frame Format	26/50	0	0	0-1	0-1	0-1
Beginning Session	26/51	0	0	0-1	0	0
ESN	26/52	0	0	1	1	1
.S. Key	26/54	0	0-1	0	0	0
.S. Request	26/55	0-1	0	0	0	0
.S. Lifetime	26/56	0	0-1	0	0	0
MN-HA SPI	26/57	0-1	0	0	0	0
MN-HA Shared Key	26/58	0	0-1	0	0	0
Remote Ipv4 Address	26/59	0	0+	0	0	0
Reserved16	26/60-69					
Remote Ipv6 Address	26/70	0	0+	0	0	0
Remote Address Table Index	26/71	0	0+	0	0	0
Remote IPv4 Address Octet Count	26/72	0	0	0	0+	0+
Allowed Differentiated Services Marking	26/73	0	0-1	0	0	0
DNS-Update-Required	26/75	0	0-1	0	0	0
Always On	26/78	0	0-1	0-1	0-1	0-1

VSA	Type	Access-Request	Access-Accept	Accounting Start	Accounting Stop	Accounting Interim-Update
Foreign Agent Address	26/79	0-1	0	0	0	0
Last User Activity	26/80	0	0	0	0-1	0-1
MN-AAA Removal Indication	26/81	0	0-1	0	0	0
RN Packet Data Inactivity Timer	26/82	0	0-1	0	0	0
Forward PDCH RC	26/83	0	0	0-1	0-1	0-1
Forward DCCH Mux Option	26/84	0	0	0-1	0-1	0-1
Reverse DCCH Mux Option	26/85	0	0	0-1	0-1	0-1
Forward DCCH RC	26/86	0	0	0-1	0-1	0-1
Reverse DCCH RC	26/87	0	0	0-1	0-1	0-1
Allowed Persistent TFTs	26/89	0	0-1	0	0	0
MIP Lifetime	26/92	0-1	0-1	0	0	0
Accounting-Stop-triggered-by-Active-Stop-Indication	26/93	0	0-1	0	0	0
Service Reference ID	26/94	0-1	0	1	1	1
DNS-Update-Capability	26/95	0-1	0	0	0	0
Remote IPv6 Address Octet Count17	26/97	0	0	0	0+	0+
PrePaidTariffSwitch (PTS)	26/98	0	0-1	0	0	0

NCC Implementation Notes:

Note 1 – RCA may be configured to add Service-Option-Profile to the first Access-Accept only. The contents of this attribute are completely configurable and are the same for all sessions.

Note 2 – The value of this attribute in the Access-Accept is configurable. RCA uses the value of this attribute in the Access-Request to determinewhether it is allowed to send Disconnect-Request messages.

Note 3 – In the Access-Accept message, RCA puts only the following sub-types in the PPAQ attribute:

- Update Reason
- Volume Quota
- Duration Quota

In the Access-request message, processes only the following sub-types in the PPAQ attribute:

- Update-Reason
- Volume-Quota (Only one out of Volume-Quota and Duration-Quota is expected.)
- Duration-Quota

Note 4 – In the Access-Accept message, RCA puts only the SelectForSession sub-type in the PPAC attribute. Possible values are “Prepaid Accounting for Volume” or “PrePaidAccounting for duration”. The value is the same for all sessions.

In the Access-Request message, RCA reads only the availableInClient sub-type of the PPAC attribute, checking that this indicates volume if RCA is configured for volume accounting, or duration if RCA is configured for duration accounting.

Service Factory

RCA supports one Service factory vendor specific attribute: SF-Access-Point-Id. This is is an ASCII string (the name of the WiFi hotspot the mobile user is connected to) and may be present in the Access-Request. RCA can use the value of this attribute to make decisions when performing credit control.

Compliance to RFC 3162 (RADIUS and IPv6)

RCA does not comply to this RFC. In parameterised mode, RCA can be configured to read attributes of type IPv6 address and store them for sending in other RADIUS messages. That is the limit of RCA support for IPv6. Source and destination addresses for UDP packets can only be IPv4 address with RCA.

In order to comply to RFC 3576, RCA should be configured to add the NAS-IPv6-Address parameter, if present in the Access-Accept, to Disconnect-request messages.

Chapter 6

Example message sequence diagrams

Overview

Introduction

This section is informational only. It gives some examples of different message RADIUS message sequences which RCA can be configured to achieve.

In this chapter

This chapter contains the following topics.

3GPP mode

```
RCA                   Radius Client
 |                              |
 |     Access-Request           |
 |<-----------------------------|
 |                              |
 |                              |
 |                              |
 |                              |
 |                              |
 |                              |
 |                              |
 |    Access-Accept (quota,trheshold)|
 |----------------------------->|
 |                              |
 |                              |
 |                              |
 |  Access-Request(threshold reached)|
 |<-----------------------------|
 |                              |
 |                              |
 |                              |
 |                              |
 |                              |
 |  Access-Accept (quota,trheshold)|
 |----------------------------->|
 |                              |
 |  Access-Request(             |
 |   3client service termination)|
 |<-----------------------------|
 |                              |
 | Access-Accept()              |
 |----------------------------->|
 |                              |
```

Paramerized mode, Accounting-Requests and Disconnect-Requests,Funds expire

In this message sequences, numbers in Mb refer to Input-Octets and Output-Octets attributes.

```
RCA                    Radius Client
 |                               |
 |     Access-Request            |
 |<------------------------------|
 |                               |
 |                               |
 |                               |
 |                               |
 |                               |
 |                               |
 |                               |
 |    Access-Accept              |
 |------------------------------>|
 |                               |
 |  Accounting-Request(START)(1) |
 |<------------------------------|
 |                               |
 | Accounting-Request(1Mb + 1Mb) |
 |<------------------------------|
 |                               |
 | Accounting-Request(2Mb + 2Mb) |
 |<------------------------------|
 |                               |
 |                               |
 |                               |
 |                               |
 |                               |
 |                               |
 |                               |
 |                               |
 |                               |
 |                               |
 |                               |
 |                               |
 | Accounting-Request(3Mb + 3Mb) |
 |<------------------------------|
 |                               |
 |                               |
 |                               |
 |                               |
 |                               |
 |                               |
 | Disconnect-Request (10)       |
 |------------------------------>|
 |                               |
 |                               |
 |                               |
 | Disconnect-Ack                |
 |<------------------------------|
```

Parameterised mode, Access-Requests and Access-Rejects, Funds expire

In this message sequences, numbers in Mb refer to Input-Octets and Output-Octets attributes.

```
RCA                     Radius Client
 |                              |
 |     Access-Request           |
 |<-----------------------------|
 |                              |
 |                              |
 |                              |
 |                              |
 |                              |
 |                              |
 |                              |
 |    Access-Accept (3Mb)       |
 |----------------------------->|
 |                              |
 |  Accounting-Request(START)   |
 |<-----------------------------|
 |                              |
 |  Accounting-Response         |
 |----------------------------->|
 |                              |
 | Accounting-Request(1Mb + 1Mb)|
 |<-----------------------------|
 |                              |
 |  Accounting-Response         |
 |----------------------------->|
 |                              |
 |                              |
 |  Access-Request(2.9Mb)       |
 |<-----------------------------|
 |                              |
 |                              |
 |                              |
 |                              |
 |                              |
 |  Access-Accept (4Mb)         |
 |----------------------------->|
 |                              |
 |  Access-Request(3.9Mb)       |
 |<-----------------------------|
 |                              |
 |                              |
 |                              |
 |                              |
 |                              |
 |  Access-Accept (5Mb)         |
 |----------------------------->|
 |                              |
 | Accounting-Request(2Mb + 2Mb)|
 |<-----------------------------|
 |                              |
 |  Accounting-Response         |
 |----------------------------->|
 |                              |
 |                              |
 |  Access-Request(4.9Mb)       |
 |<-----------------------------|
 |                              |
 |                              |
 |                              |
 |                              |
 |                              |
 |                              |
 |  Access-Accept (0.1Mb)       |
  |----------------------------->|
  |                              |
  |  Access-Request(5Mb,sesion active)|
  |<-----------------------------|
  |                              |
  |  Access-Reject               |
  |----------------------------->|
  |                              |
  |                              |
  |                              |
```

Parameterised mode, Access-Requests and Access-Rejects, Subscriber ends session

In this message sequences, numbers in Mb refer to Input-Octets and Output-Octets attributes.

```
 RCA                     Radius Client
  |                              |
  |     Access-Request           |
  |<-----------------------------|
  |                              |
  |                              |
  |                              |
  |                              |
  |                              |
  |                              |
  |                              |
  |    Access-Accept (3Mb)       |
  |----------------------------->|
  |                              |
  |  Accounting-Request(START)   |
  |<-----------------------------|
  |                              |
  |  Accounting-Response         |
  |----------------------------->|
  |                              |
  | Accounting-Request(1Mb + 1Mb)|
  |<-----------------------------|
  |                              |
  |  Accounting-Response         |
  |----------------------------->|
  |                              |
  |                              |
  |  Access-Request(2.9Mb)       |
  |<-----------------------------|
  |                              |
  |                              |
  |                              |
  |                              |
  |                              |
  |  Access-Accept (4Mb)         |
  |----------------------------->|
  |                              |
  |  Access-Request(             |
  |   3.5Mb,session not active)  |
  |<-----------------------------|
  |                              |
  | Access-Accept()              |
  |----------------------------->|
  |                              |
```