Network Working Group
Request for Comments: 2058
Category: Standards Track

C. Rigney
Livingston
A. Rubens
Merit
W. Simpson
Daydreamer
S. Willens
Livingston

January 1997

Remote Authentication Dial In User Service (RADIUS)

Status of this Memo

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.

Table of Contents

1. Introduction

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

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.

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

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

In this document, several words are used to signify the requirements of the specification. These words are often capitalized.

MUST This word, or the adjective "required", means that the definition is an absolute requirement of the specification.

MUST NOT This phrase means that the definition is an absolute prohibition of the specification.

SHOULD This word, or the adjective "recommended", means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications must be understood and carefully weighed before choosing a different course.

MAY This word, or the adjective "optional", means that this item is one of an allowed set of alternatives. An implementation which does not include this option MUST be prepared to interoperate with another implementation which does include the option.

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

When a client is configured to use RADIUS, any user of the client presents authentication information to the client. This might be with a customizable 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 [1].

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.

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 should 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 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 are permitted in an Access-Reject.

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 Attributes 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

In challenge/response authentication, the user is given an unpredictable number and challenged to encrypt it and give back the result. Authorized users are equipped with special devices such as smart cards or software that facilitate calculation of the correct response with ease. Unauthorized 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 should be in the possession of the authorized 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 what it should be, 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 16 octets) 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.

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 should 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. 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 that 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.

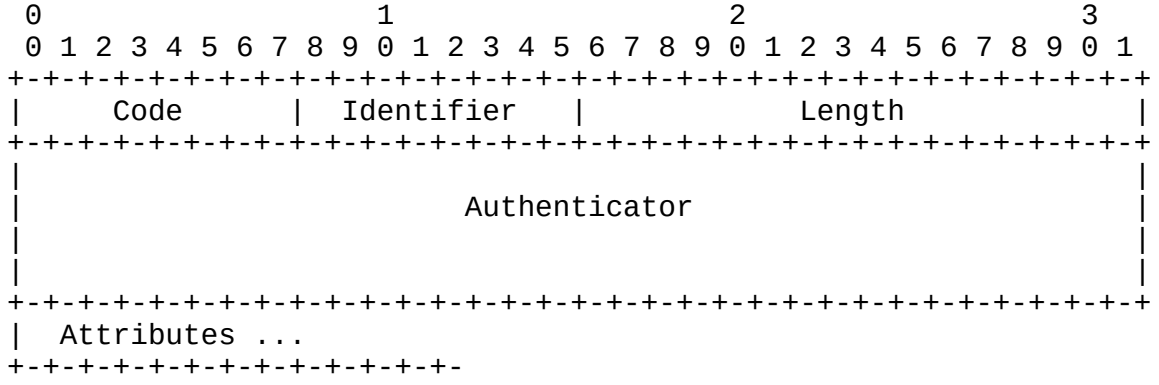
Without TCP we would still probably be using tin cans connected by string. But for this particular protocol, UDP is a better choice.

3. Packet Format

Exactly one RADIUS packet is encapsulated in the UDP Data field [2], where the UDP Destination Port field indicates 1812 (decimal).

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

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



Code

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

RADIUS Codes (decimal) are assigned as follows:

1	Access-Request
2	Access-Accept
3	Access-Reject
4	Accounting-Request
5	Accounting-Response
11	Access-Challenge
12	Status-Server (experimental)
13	Status-Client (experimental)
255	Reserved

Codes 4 and 5 will be covered in the RADIUS Accounting document [9], and are not further mentioned here. 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.

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 should be treated as padding and should be ignored on reception. If the packet is shorter than the Length field indicates, it should be silently discarded. The minimum length is 20 and 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 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 xored 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.

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. A RADIUS server SHOULD 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 can add a Proxy-State Attribute, and when the proxy forwards a response, it removes the Proxy-State Attribute. Since Access-Accept and Access-Reject replies are authenticated on the entire packet contents, the stripping of the Proxy-State attribute would invalidate 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.

Attributes

Many 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.

In the section below on "Attributes" where the text refers to which packets an attribute is allowed in, only packets with Codes 1, 2, 3 and 11 and attributes defined in this document are covered in this document. A summary table is provided at the end of the "Attributes" section. To determine which Attributes are allowed in packets with codes 4 and 5 refer to the RADIUS Accounting document [9].

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 MUST contain a User-Name attribute. It SHOULD contain either a NAS-IP-Address attribute or NAS-Identifier attribute (or both, although that is not recommended). It MUST contain either a User-Password attribute or CHAP-Password attribute. It 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 [1].

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

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

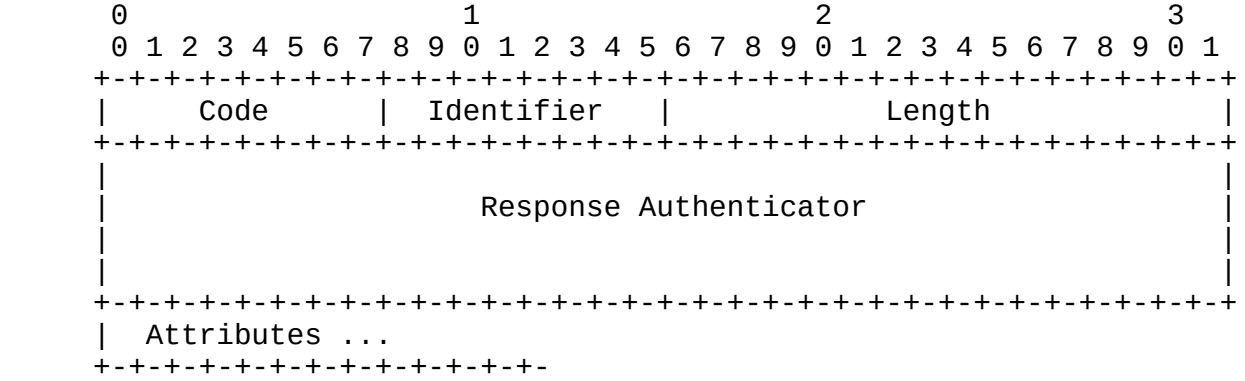
The Attribute field is variable in length, and contains the list of Attributes that are required for the type of service, as well as any desired optional Attributes.

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. Additionally, 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.



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.

Response Authenticator

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

Attributes

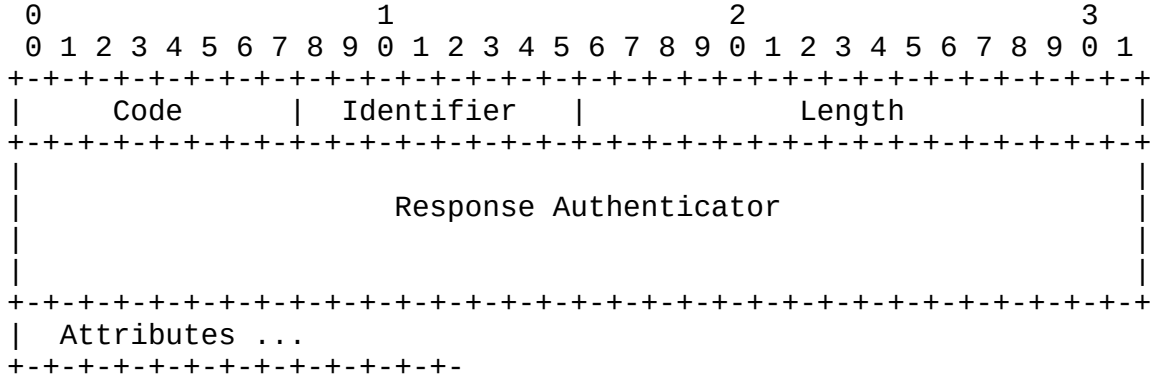
The Attribute field is variable in length, and contains a list of zero or more Attributes.

4.3. 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.

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



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

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. No other Attributes are permitted in an Access-Challenge.

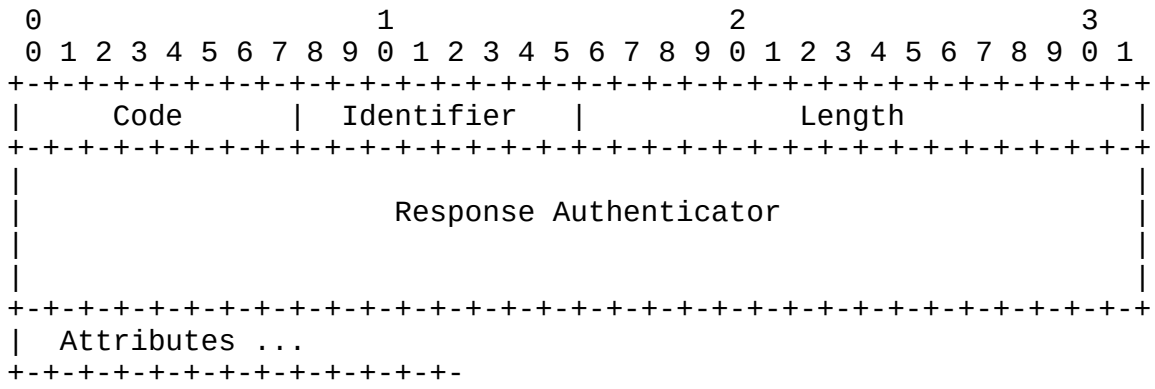
On receipt of an Access-Challenge, the Identifier field is matched with a pending Access-Request. Additionally, the Response Authenticator field MUST contain the correct response for the pending Access-Request. Invalid packets are silently discarded.

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 dialin 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 should 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.



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.

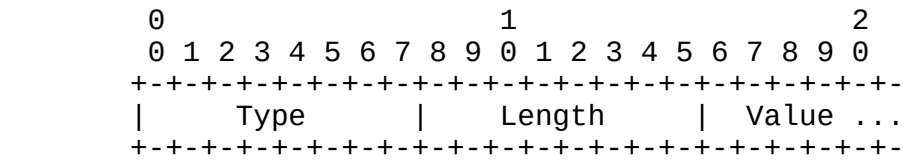
5. Attributes

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

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

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

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



Type

The Type field is one octet. Up-to-date values of the RADIUS Type field are specified in the most recent "Assigned Numbers" RFC [3]. 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:

A RADIUS server MAY ignore Attributes with an unknown Type.

A RADIUS client MAY ignore Attributes with an unknown Type.

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. 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 a "string" in RADIUS does not require termination by an ASCII NUL because the Attribute already has a length field.

The format of the value field is one of four data types.

string	0-253 octets
address	32 bit value, most significant octet first.
integer	32 bit value, most significant octet first.

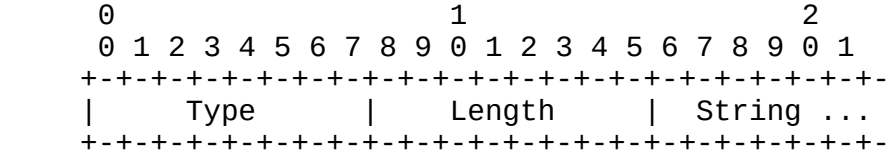
time 32 bit value, most significant octet first -- seconds since 00:00:00 GMT, January 1, 1970. The standard Attributes do not use this data type but it is presented here for possible use within Vendor-Specific attributes.

5.1. User-Name

Description

This Attribute indicates the name of the user to be authenticated. It is only used in Access-Request packets.

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



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:

monolithic Consisting only of alphanumeric characters. This simple form might be used to locally manage a NAS.

simple Consisting only of printable ASCII characters.

name@fqdn SMTP address. The Fully Qualified Domain Name (with or without trailing dot) indicates the realm in which the name part applies.

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 [4] pages 109-110. A more precise explanation of the method follows:

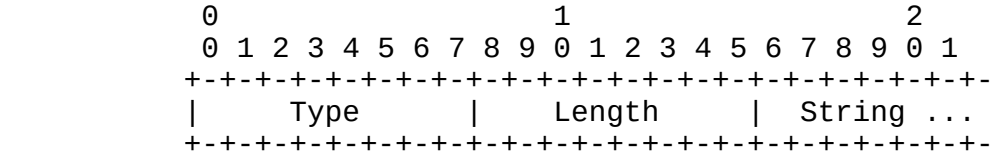
Call the shared secret S and the pseudo-random 128-bit Request Authenticator RA. Break the password into 16-octet chunks p1, p2, etc. with the last one padded at the end with nulls to a 16-octet boundary. Call the ciphertext blocks c(1), c(2), etc. We'll need intermediate values b1, b2, etc.

```
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.



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.

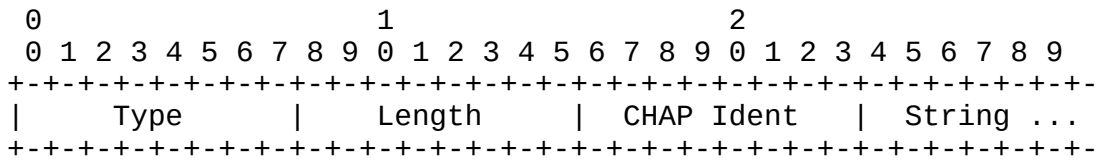
5.3. CHAP-Password

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.



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

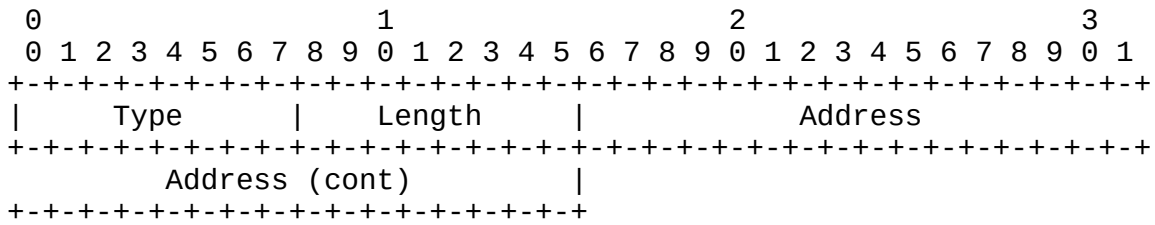
The String field is 16 octets, and contains the CHAP Response from the user.

5.4. NAS-IP-Address

Description

This Attribute indicates the identifying IP Address of the NAS which is requesting authentication of the user. It is only used in Access-Request packets. Either NAS-IP-Address or NAS-Identifier SHOULD be present in an Access-Request packet.

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



Type

4 for NAS-IP-Address.

Length

6

Address

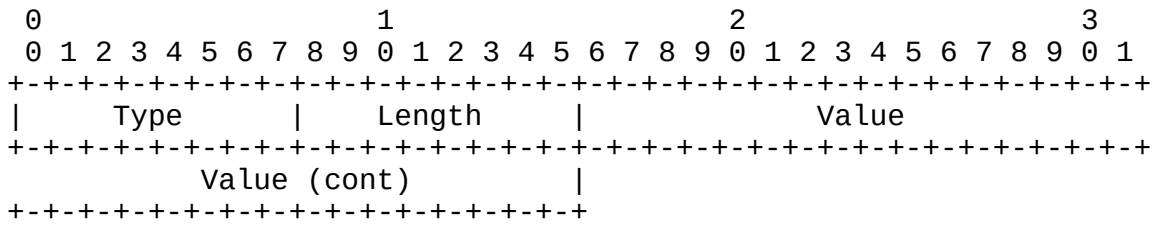
The Address field is four octets.

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.



Type

5 for NAS-Port.

Length

6

Value

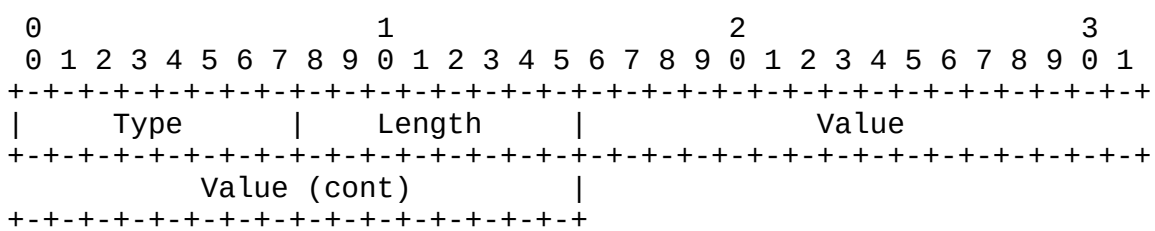
The Value field is four octets. Despite the size of the field, values range from 0 to 65535.

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.



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
```

The service types are defined as follows when used in an Access-Accept. When used in an Access-Request, they should 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.

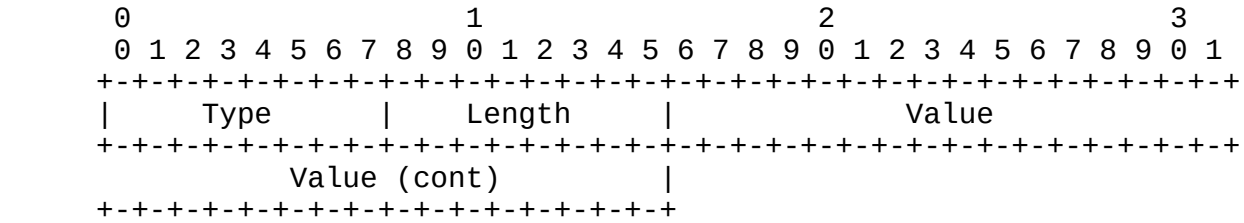
NAS Prompt	The user should be provided a command prompt on the NAS from which non-privileged commands can be executed.
Authenticate Only	Only Authentication is requested, and no authorization information needs to be returned in the Access-Accept (typically used by proxy servers rather than the NAS itself).
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.

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.



Type

7 for Framed-Protocol.

Length

6

Value

The Value field is four octets.

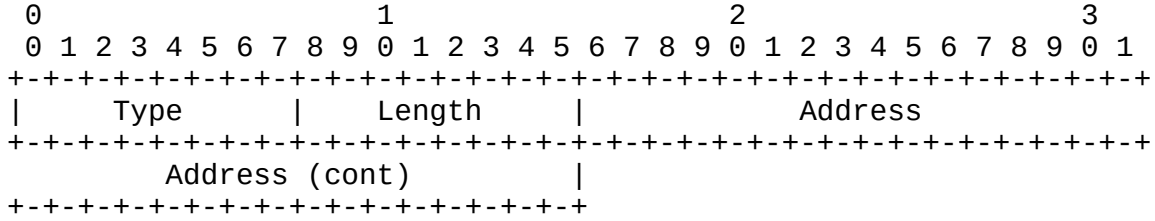
1	PPP
2	SLIP
3	AppleTalk Remote Access Protocol (ARAP)
4	Gandalf proprietary SingleLink/MultiLink protocol
5	Xylogics proprietary IPX/SLIP

5.8. Framed-IP-Address

Description

This Attribute indicates the address to be configured for the user. 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 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.



Type

8 for Framed-IP-Address.

Length

6

Address

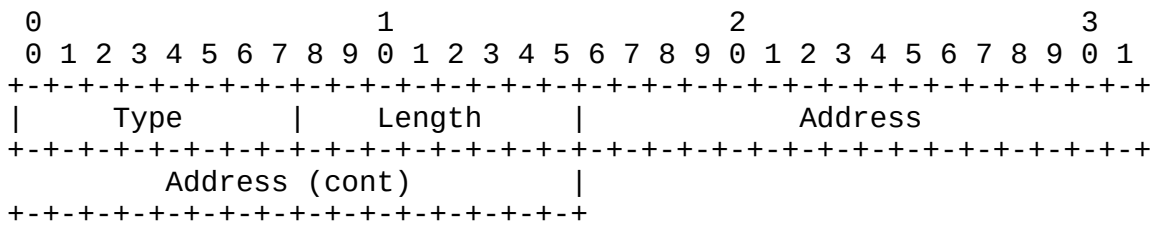
The Address field is four octets. The value 0xFFFFFFFF indicates that the NAS should allow the user to select an address (e.g. Negotiated). The value 0xFFFFFFFE indicates that the NAS should select an address for the user (e.g. Assigned from a pool of addresses kept by the NAS). Other valid values indicate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

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.



Type

9 for Framed-IP-Netmask.

Length

6

Address

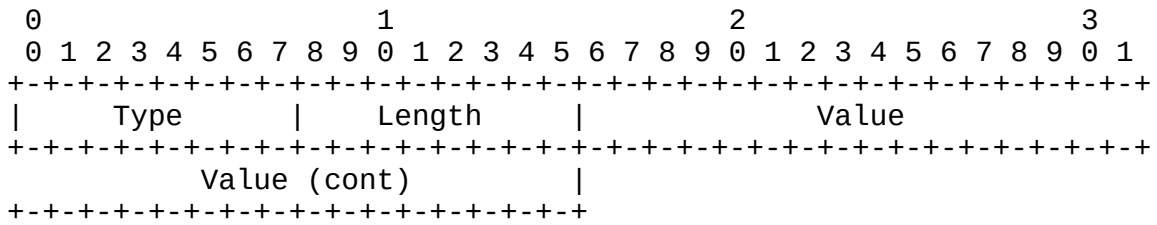
The Address field is four octets specifying the IP netmask of the user.

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.



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

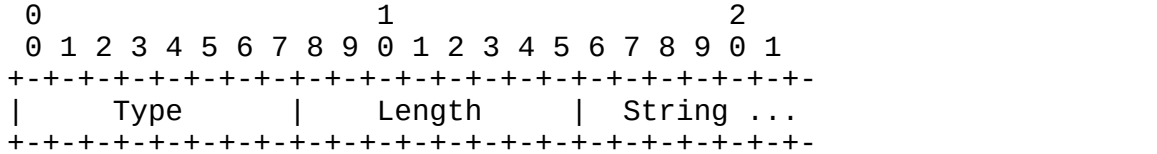
5.11. Filter-Id

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.



Type

11 for Filter-Id.

Length

>= 3

String

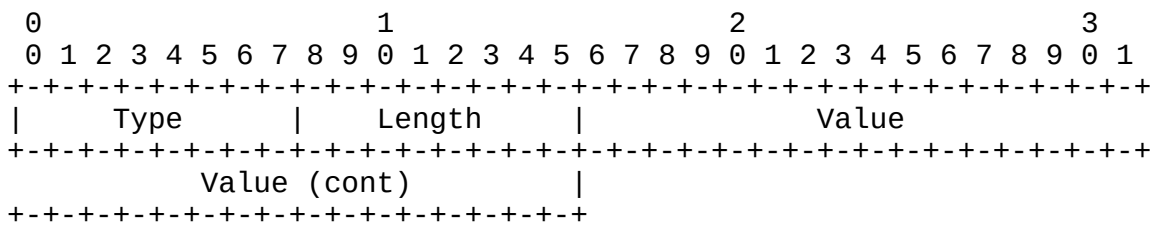
The String field is one or more octets, and its contents are implementation dependent. It is intended to be human readable and MUST NOT affect operation of the protocol. It is recommended that the message contain displayable ASCII characters from the range 32 through 126 decimal.

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 is only used in Access-Accept packets.

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



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.

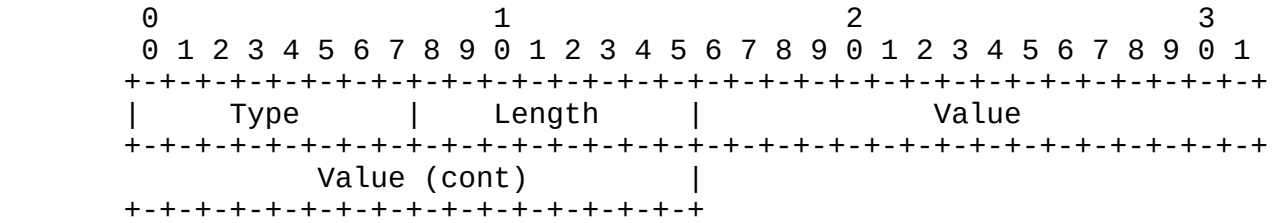
5.13. Framed-Compression

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.



Type

13 for Framed-Compression.

Length

6

Value

The Value field is four octets.

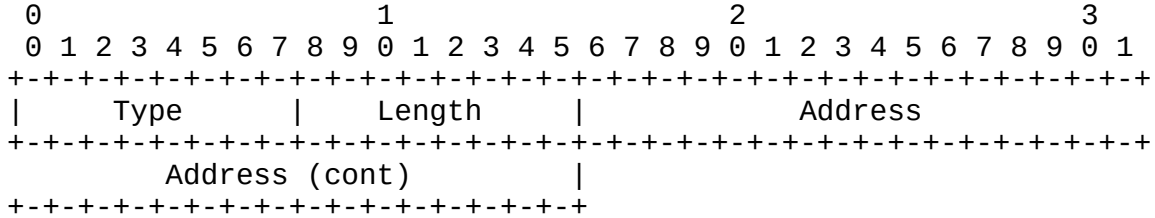
0	None
1	VJ TCP/IP header compression [5]
2	IPX header compression

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.



Type

14 for Login-IP-Host.

Length

6

Address

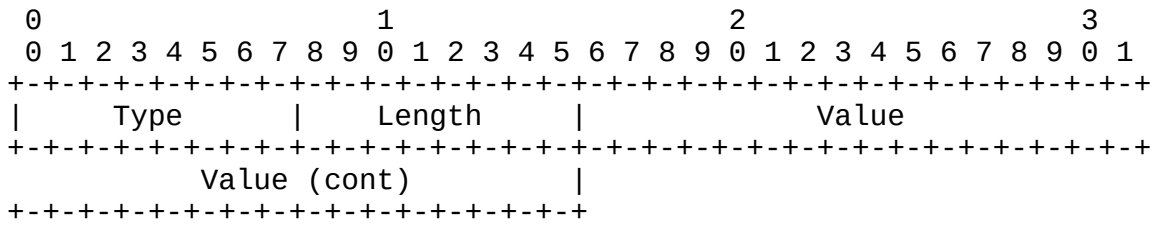
The Address field is four octets. The value 0xFFFFFFFF indicates that the NAS SHOULD allow the user to select an address. The value 0 indicates that the NAS SHOULD select a host to connect the user to. Other values indicate the address the NAS SHOULD connect the user to.

5.15. Login-Service

Description

This Attribute indicates the service which should be used 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.



Type

15 for Login-Service.

Length

6

Value

The Value field is four octets.

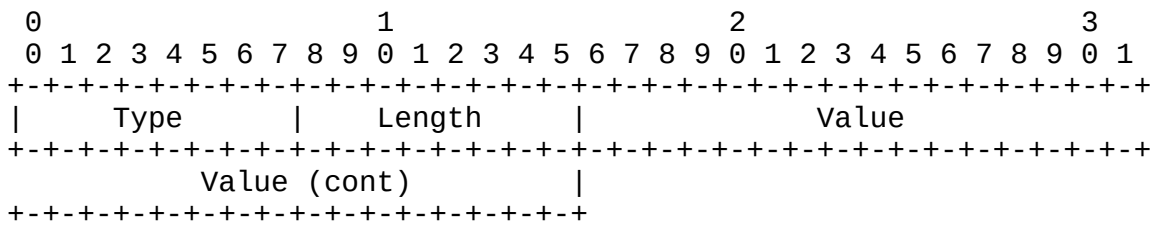
0	Telnet
1	Rlogin
2	TCP Clear
3	PortMaster (proprietary)
4	LAT

5.16. 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.



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.

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.

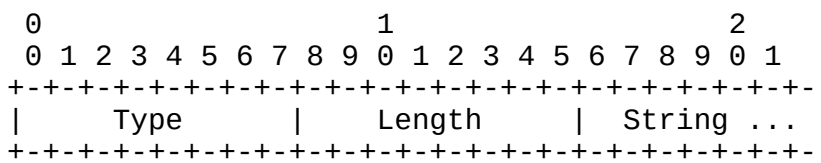
When used in an Access-Accept, it is the success message.

When used in an Access-Reject, it is the failure message. It MAY indicate a dialog message to prompt the user before another Access-Request attempt.

When used in an 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.

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



Type

18 for Reply-Message.

Length

>= 3

String

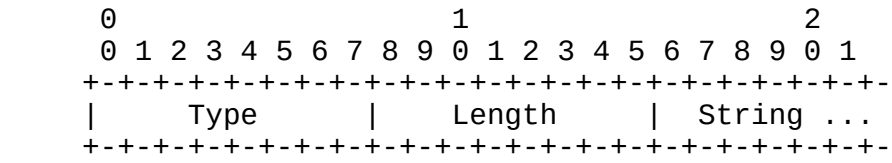
The String field is one or more octets, and its contents are implementation dependent. It is intended to be human readable, and MUST NOT affect operation of the protocol. It is recommended that the message contain displayable ASCII characters from the range 10, 13, and 32 through 126 decimal. Mechanisms for extension to other character sets are beyond the scope of this specification.

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.



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.

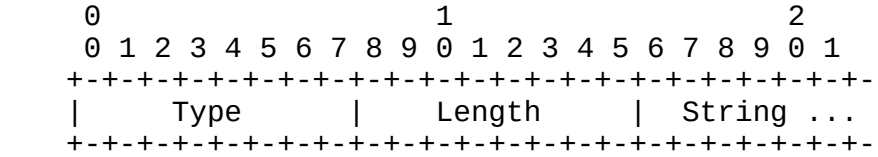
The codification of the range of allowed usage of this field is outside the scope of this specification.

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.



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.

5.21. (unassigned)

Description

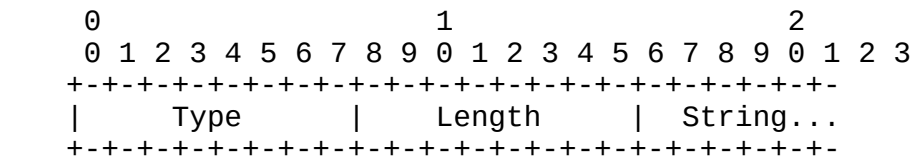
ATTRIBUTE TYPE 21 HAS NOT BEEN ASSIGNED.

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.



Type

22 for Framed-Route.

Length

>= 3

String

The String 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 displayable ASCII characters from the range 32 through 126 decimal.

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 should be used. 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 should default 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".

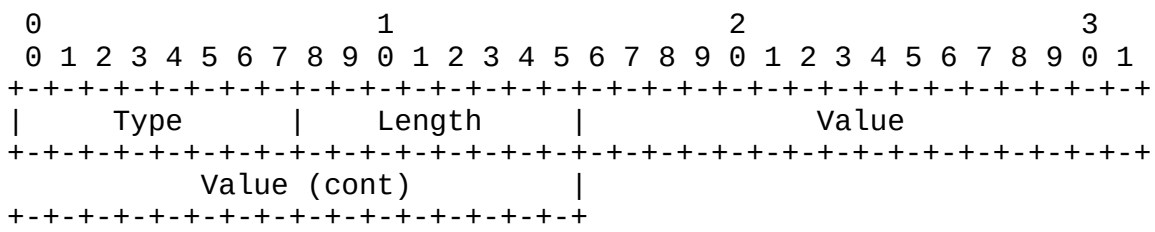
Whenever the gateway address is specified as "0.0.0.0" the IP address of the user SHOULD be used as the gateway address.

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.



Type

23 for Framed-IPX-Network.

Length

6

Value

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.

5.24. State

Description

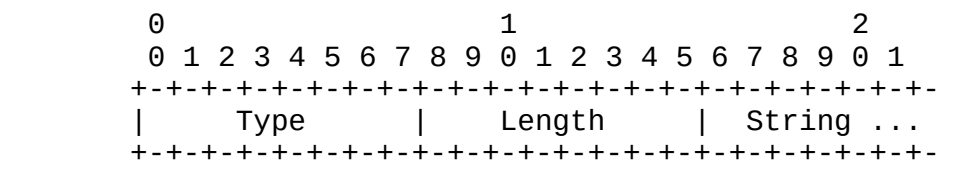
This Attribute is available to be sent by the server to the client in an Access-Challenge and MUST be sent unmodified from the client to the server in the new Access-Request reply to that challenge, if any.

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, no interpretation by the client should be made. A packet may have only 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.



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.

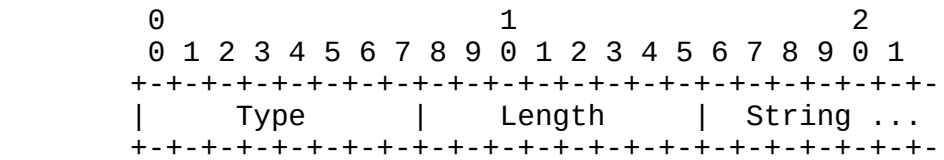
The codification of the range of allowed usage of this field is outside the scope of this specification.

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. No interpretation by the client should be made.

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



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.

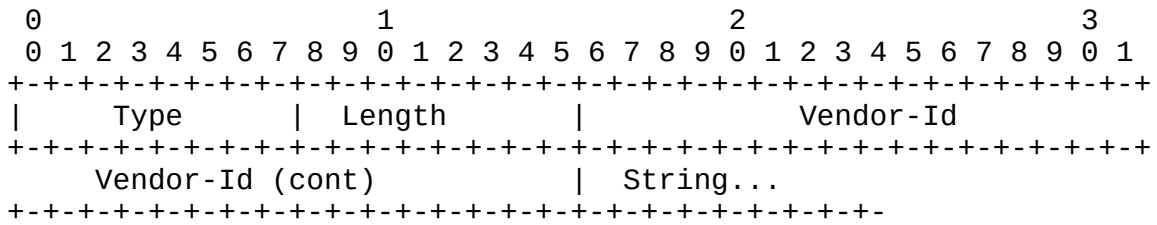
5.26. Vendor-Specific

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.



Type

26 for Vendor-Specific.

Length

>= 7

Vendor-Id

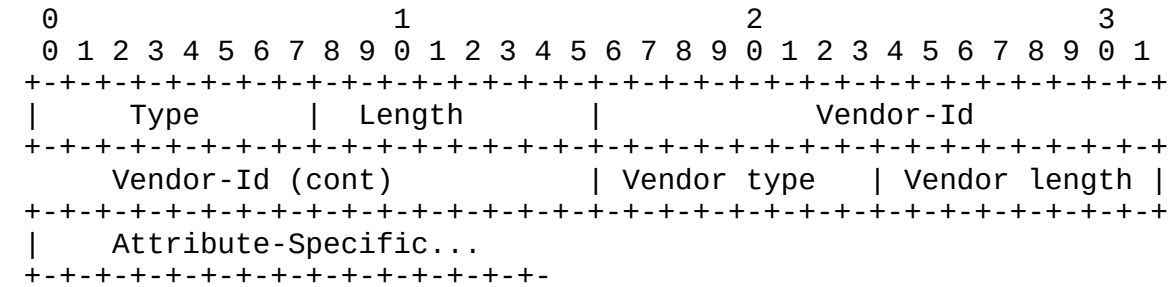
The high-order octet is 0 and the low-order 3 octets are the SMI Network Management Private Enterprise Code of the Vendor in network byte order, as defined in the Assigned Numbers RFC [2].

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:

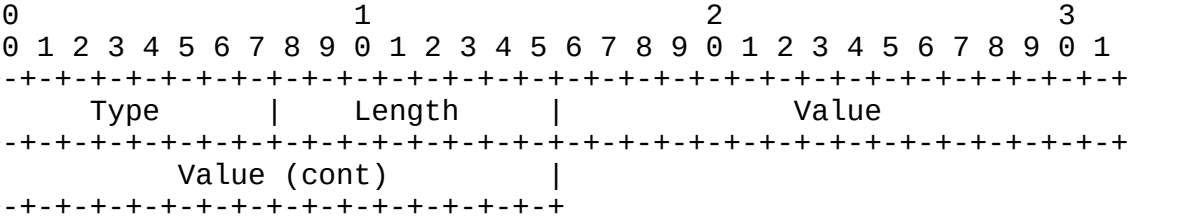


5.27. Session-Timeout

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.



Type

27 for Session-Timeout.

Length

6

Value

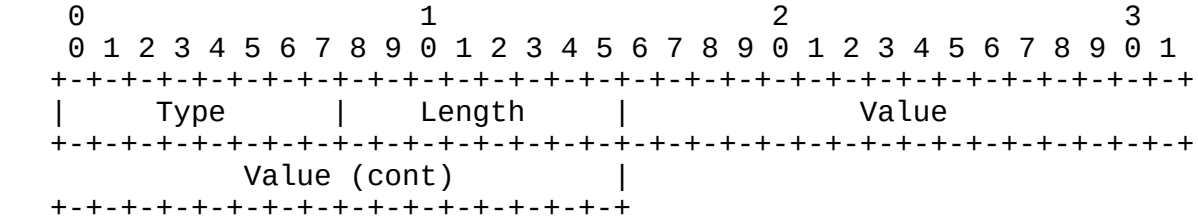
The field is 4 octets, containing a 32-bit unsigned integer with the maximum number of seconds this user should be allowed to remain connected by the NAS.

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.



Type

28 for Idle-Timeout.

Length

6

Value

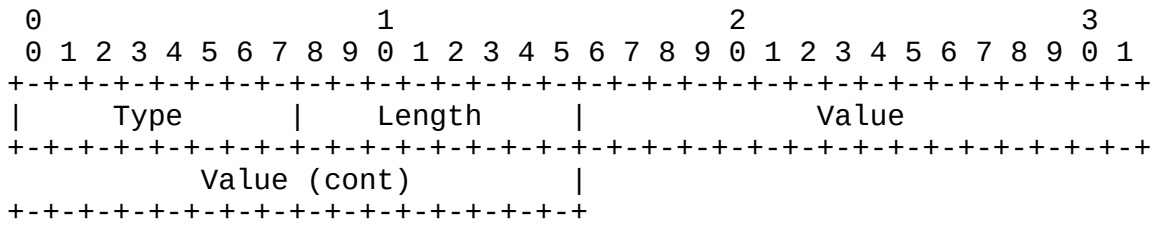
The field is 4 octets, containing a 32-bit unsigned integer with the maximum number of consecutive seconds of idle time this user should be permitted before being disconnected by the NAS.

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.



Type

29 for Termination-Action.

Length

6

Value

The Value field is four octets.

0	Default
1	RADIUS-Request

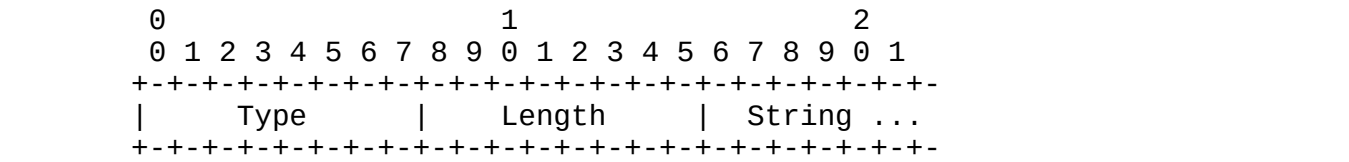
If the Value is set to RADIUS-Request, upon termination of the specified service the NAS MAY send a new Access-Request to the RADIUS server, including the State attribute if any.

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.



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. Printable ASCII is recommended, but a robust implementation SHOULD support the field as undistinguished octets.

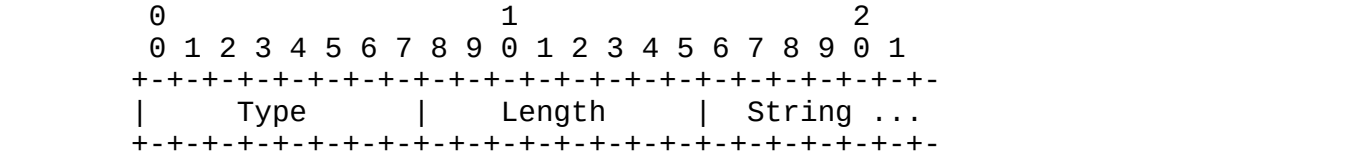
The codification of the range of allowed usage of this field is outside the scope of this specification.

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.



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. Printable ASCII is recommended, but a robust implementation SHOULD support the field as undistinguished octets.

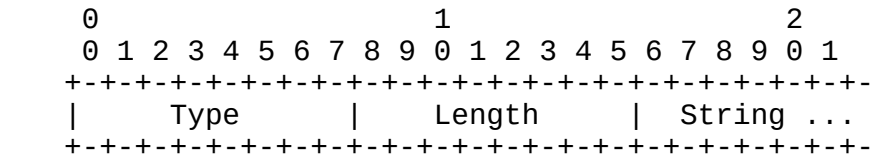
The codification of the range of allowed usage of this field is outside the scope of this specification.

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 SHOULD be present in an Access-Request packet.

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



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.

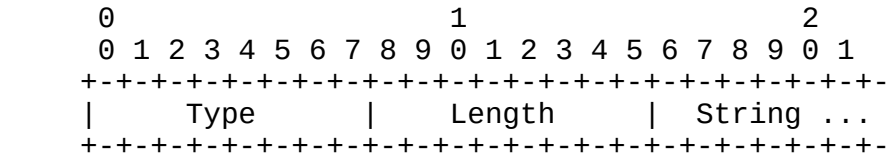
5.33. Proxy-State

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. This attribute should be removed by the proxy server before the response is forwarded to the NAS.

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.



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.

5.34. Login-LAT-Service

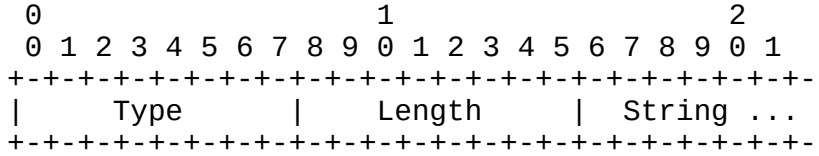
Description

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.



Type

34 for Login-LAT-Service.

Length

>= 3

String

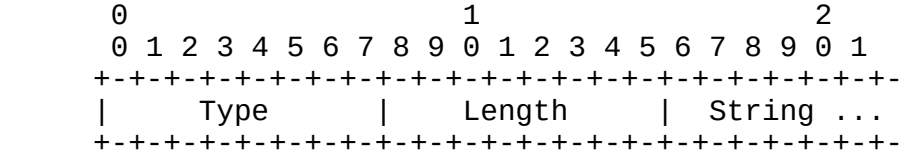
The String field is one or more octets, and contains the identity of the LAT service to use. The LAT Architecture allows this string to contain $ (dollar), - (hyphen), . (period), _ (underscore), numerics, upper and lower case alphabetics, and the ISO Latin-1 character set extension [6]. All LAT string comparisons are case insensitive.

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.



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.

5.36. Login-LAT-Group

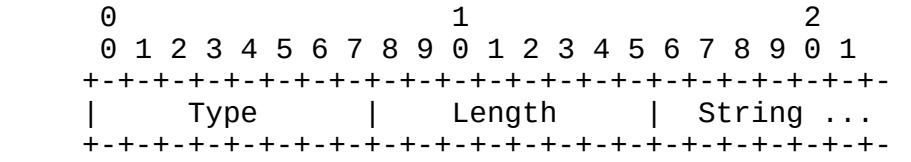
Description

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.



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.

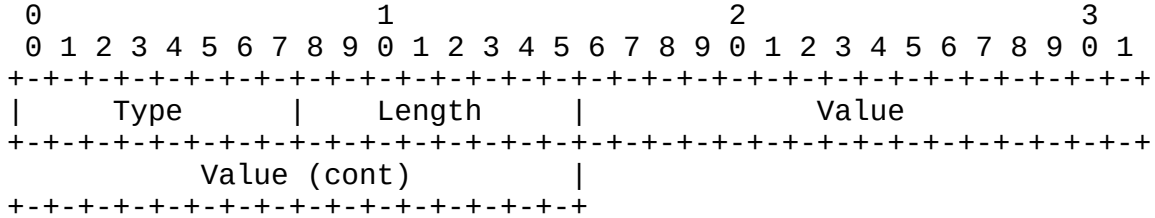
The codification of the range of allowed usage of this field is outside the scope of this specification.

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.



Type

37 for Framed-AppleTalk-Link.

Length

6

Value

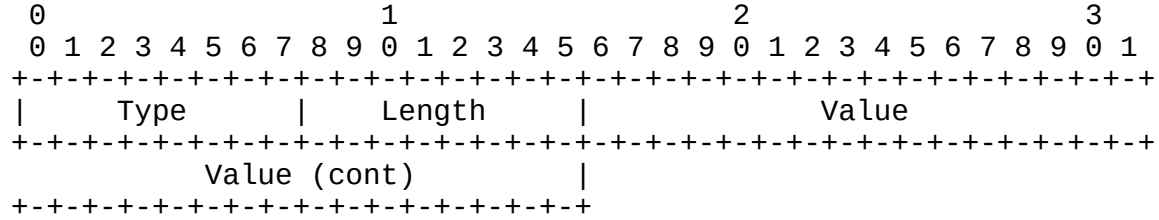
The Value field is four octets. Despite the size of the field, values range from 0 to 65535. The special value of 0 indicates that this is an unnumbered serial link. A value of 1-65535 means that the serial line between the NAS and the user should be assigned that value as an AppleTalk network number.

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.



Type

38 for Framed-AppleTalk-Network.

Length

6

Value

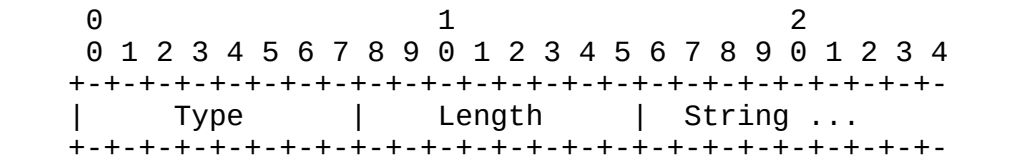
The Value field is four octets. Despite the size of the field, values range from 0 to 65535. The special value 0 indicates that the NAS should assign a network for the user, using its default cable range. A value between 1 and 65535 (inclusive) indicates the AppleTalk Network the NAS should probe to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



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.

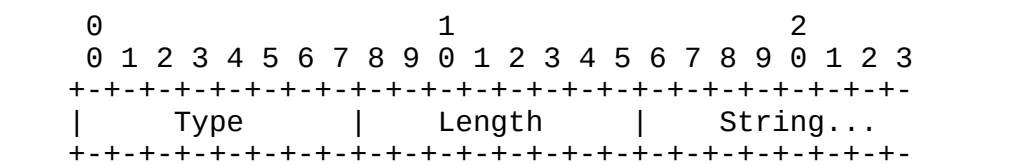
5.40. CHAP-Challenge

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.



Type

60 for CHAP-Challenge.

Length

>= 7

String

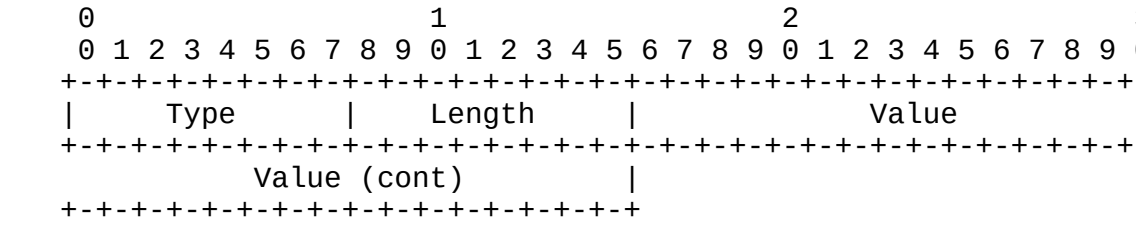
The String field contains the CHAP Challenge.

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.



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.

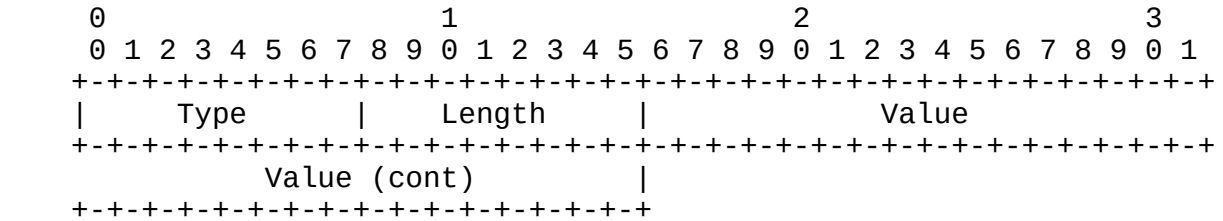
0	Async
1	Sync
2	ISDN Sync
3	ISDN Async V.120
4	ISDN Async V.110
5	Virtual

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 [7] 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.



Type

62 for Port-Limit.

Length

6

Value

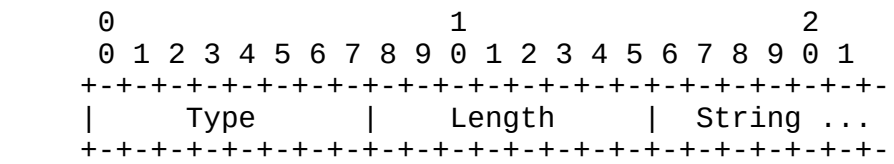
The field is 4 octets, containing a 32-bit unsigned integer with the maximum number of ports this user should be allowed to connect to on the NAS.

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.



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.

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
1	0	0	0	1	User-Name
0-1	0	0	0	2	User-Password [Note 1]
0-1	0	0	0	3	CHAP-Password [Note 1]
0-1	0	0	0	4	NAS-IP-Address
0-1	0	0	0	5	NAS-Port
0-1	0-1	0	0	6	Service-Type
0-1	0-1	0	0	7	Framed-Protocol
0-1	0-1	0	0	8	Framed-IP-Address
0-1	0-1	0	0	9	Framed-IP-Netmask
0	0-1	0	0	10	Framed-Routing
0	0+	0	0	11	Filter-Id
0	0-1	0	0	12	Framed-MTU
0+	0+	0	0	13	Framed-Compression
0+	0+	0	0	14	Login-IP-Host
0	0-1	0	0	15	Login-Service
0	0-1	0	0	16	Login-TCP-Port
0	0+	0+	0+	18	Reply-Message
0-1	0-1	0	0	19	Callback-Number
0	0-1	0	0	20	Callback-Id
0	0+	0	0	22	Framed-Route
0	0-1	0	0	23	Framed-IPX-Network
0-1	0-1	0	0-1	24	State
0	0+	0	0	25	Class
0+	0+	0	0+	26	Vendor-Specific
0	0-1	0	0-1	27	Session-Timeout
0	0-1	0	0-1	28	Idle-Timeout
0	0-1	0	0	29	Termination-Action
0-1	0	0	0	30	Called-Station-Id
0-1	0	0	0	31	Calling-Station-Id
0-1	0	0	0	32	NAS-Identifier
0+	0+	0+	0+	33	Proxy-State
0-1	0-1	0	0	34	Login-LAT-Service
0-1	0-1	0	0	35	Login-LAT-Node
0-1	0-1	0	0	36	Login-LAT-Group
0	0-1	0	0	37	Framed-AppleTalk-Link
0	0+	0	0	38	Framed-AppleTalk-Network
0	0-1	0	0	39	Framed-AppleTalk-Zone
0-1	0	0	0	60	CHAP-Challenge
0-1	0	0	0	61	NAS-Port-Type
0-1	0-1	0	0	62	Port-Limit
0-1	0-1	0	0	63	Login-LAT-Port
Request	Accept	Reject	Challenge	#	Attribute

[Note 1] An Access-Request MUST contain either a User-Password or a CHAP-Password, and MUST NOT contain both.

The following table defines the meaning of the above table entries.

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

6. 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.

6.1. 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.

```
Code = 1        (Access-Request)
ID = 0
Request Authenticator = {16 octet random number}
Attributes:
    User-Name = "nemo"
    User-Password = {16 octets of Password padded at end with nulls,
                XORed with MD5(shared secret|Request Authenticator)}
    NAS-IP-Address = 192.168.1.16
    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.

```
Code = 2        (Access-Accept)
ID = 0          (same as in Access-Request)
Response Authenticator = {16-octet MD-5 checksum of the code (2),
                id (0), the Request Authenticator from above, the
                attributes in this reply, and the shared secret}
Attributes:
    Service-Type = Login-User
    Login-Service = Telnet
    Login-Host = 192.168.1.3
```

6.2. 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.

```
Code = 1        (Access-Request)
ID = 1
Request Authenticator = {16 octet random number also used as
                         CHAP challenge}
Attributes:
    User-Name = "flopsy"
    CHAP-Password = {1 octet CHAP ID followed by 16 octet
                     CHAP response}
    NAS-IP-Address = 192.168.1.16
    NAS-Port = 20
    Service-Type = Framed-User
    Framed-Protocol = PPP
```

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.

```
Code = 2        (Access-Accept)
ID = 1          (same as in Access-Request)
Response Authenticator = {16-octet MD-5 checksum of the code (2),
                id (1), the Request Authenticator from above, the
                attributes in this reply, and the shared secret}
Attributes:
    Service-Type = Framed-User
    Framed-Protocol = PPP
    Framed-IP-Address = 255.255.255.254
    Framed-Routing = None
    Framed-Compression = 1      (VJ TCP/IP Header Compression)
    Framed-MTU = 1500
```

6.3. User with Challenge-Response card

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.

```
Code = 1        (Access-Request)
ID = 2
Request Authenticator = {16 octet random number}
Attributes:
    User-Name = "mopsy"
    User-Password = {16 octets of Password padded at end with nulls,
                XORed with MD5(shared secret|Request Authenticator)}
    NAS-IP-Address = 192.168.1.16
    NAS-Port = 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.

```
Code = 11       (Access-Challenge}
ID = 2          (same as in Access-Request)
Response Authenticator = {16-octet MD-5 checksum of the code (11),
                id (2), the Request Authenticator from above, the
                attributes in this reply, and the shared secret}
Attributes:
    Reply-Message = "Challenge 32769430.  Enter response at prompt."
    State =     {Magic Cookie to be returned along with user's response}
```

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

```
Code = 1        (Access-Request)
ID = 3          (Note that this changes)
Request Authenticator = {NEW 16 octet random number}
Attributes:
    User-Name = "mopsy"
    User-Password = {16 octets of Response padded at end with
                nulls, XORed with MD5 checksum of shared secret
                plus above Request Authenticator}
    NAS-IP-Address = 192.168.1.16
    NAS-Port = 7
    State =     {Magic Cookie from Access-Challenge packet, unchanged}
```

The Response was incorrect, so the RADIUS server tells the NAS to reject the login attempt.

```
Code = 3        (Access-Reject)
ID = 3          (same as in Access-Request)
Response Authenticator = {16-octet MD-5 checksum of the code (3),
                id (3), the Request Authenticator from above, the
                attributes in this reply if any, and the shared
                 secret}
Attributes:
        (none, although a Reply-Message could be sent)
```

Security Considerations

Security issues are the primary topic of this document.

In practice, within or associated with each RADIUS server, there is a database which associates "user" names with authentication information ("secrets"). 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 [8], but such a mechanism is outside the scope of this specification.

Other distribution methods are currently undergoing research and experimentation. The SNMP Security document [8] also has an excellent overview of threats to network protocols.

References

[1] Rivest, R., and S. Dusse, "The MD5 Message-Digest Algorithm", RFC 1321, MIT Laboratory for Computer Science, RSA Data Security Inc., April 1992.

[2] Postel, J., "User Datagram Protocol", STD 6, RFC 768, USC/Information Sciences Institute, August 1980.

[3] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1700, USC/Information Sciences Institute, October 1994.

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

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

[6] ISO 8859. International Standard -- Information Processing -- 8-bit Single-Byte Coded Graphic Character Sets -- Part 1: Latin Alphabet No. 1, ISO 8859-1:1987. <URL:http://www.iso.ch/cate/d16338.html>

[7] Sklower, K., Lloyd, B., McGregor, G., and Carr, D., "The PPP Multilink Protocol (MP)", RFC 1717, University of California Berkeley, Lloyd Internetworking, Newbridge Networks Corporation, November 1994.

[8] Galvin, J., McCloghrie, K., and J. Davin, "SNMP Security Protocols", RFC 1352, Trusted Information Systems, Inc., Hughes LAN Systems, Inc., MIT Laboratory for Computer Science, July 1992.

[9] Rigney, C., "RADIUS Accounting", RFC 2059, January 1997.

Acknowledgments

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

Chair's Address

The working group can be contacted via the current chair:

Carl Rigney
Livingston Enterprises
6920 Koll Center Parkway, Suite 220
Pleasanton, California 94566

Phone: +1 510 426 0770
EMail: cdr@livingston.com

Authors' Addresses

Questions about this memo can also be directed to:

Carl Rigney
Livingston Enterprises
6920 Koll Center Parkway, Suite 220
Pleasanton, California 94566

Phone: +1 510 426 0770
EMail: cdr@livingston.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
6920 Koll Center Parkway, Suite 220
Pleasanton, California 94566

EMail: steve@livingston.com