Network Working Group
Internet Draft
Intended status: Experimental
Expires: November 2013

P. Deacon
IEA Software, Inc

May 1, 2013

RADIUS Extended Request
draft-deacon-radext-extended-request-01.txt

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on August 1, 2013.

Copyright Notice

Abstract

This document describes methods for RADIUS servers to communicate optional extended abilities to RADIUS clients. The abilities

described provide for exchange of RADIUS packets where total packet length exceeds 4096 bytes.

Table of Contents

1. Introduction

Historically RADIUS packets transporting Authentication Authorization and Accounting (AAA) information required a small fraction of 4096 byte message limit allotted by [RFC2865]. Today need for larger packets driven by progressively complex security and configuration requirements has increased pressure for RADIUS beyond 4096 bytes.

This text describes methods for enabling RADIUS clients and servers to effectively exchange large RADIUS packets above limits prescribed by [RFC2865].

To maintain compatibility with existing RADIUS infrastructure a protocol is defined such that large packets shall be permitted by RADIUS server or client only after support for large packets has been established. This is achieved automatically using the protocol defined in this text or by non-default administrative settings.

Two methods for supporting RADIUS packets beyond 4096 bytes are described.

Switching to TCP - Clients normally using UDP may elect to use TCP [RFC6614] always or only while large packets shall be exchanged. Since RADIUS over TCP is also limited to 4096 bytes procedures are described for establishing availability of TCP to UDP clients as well as TCP support for large packets to UDP and TCP clients. TCP is the recommended method for transmitting large RADIUS packets.

Fragmentation - Intended for UDP interoperability with TCP. This approach is based on fragmenting large packets into a series of smaller RADIUS packets, transmitting and finally reassembling all packet fragments. Procedures to signal support for fragmentation to client and server are described.

1.1. Terminology

This document uses these terms:

Head Attribute Attribute immediately following header of an Extended-Request packet.

Outer Packet	When a packet encapsulates another packet the encapsulating packet is known as the outer packet.
Inner Packet	When a packet encapsulates another packet the encapsulated packet is known as the inner packet.
RADIUS Client	For purposes of this document a RADIUS client is that which initiates a RADIUS request packet.
RADIUS Server	For purposes of this document a RADIUS server is that which responds to a RADIUS request packet.
Attribute	Usage of the word attribute in this document does not include "long attributes" or similar concepts where a logical attribute is created by aggregation of multiple Type-Length-Value fields.

2. Conventions used in this document

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 [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

In this document, the characters ">>" preceding an indented line(s) indicates a compliance requirement statement using the key words listed above. This convention aids reviewers in quickly identifying or finding the explicit compliance requirements of this RFC.

3. Extended-Request

An Extended-Request packet is sent to the RADIUS server requesting an action whose purpose is determined by an attribute present immediately after RADIUS header within Extended-Request packet. This attribute is known as the "head attribute". All subsequent attributes are evaluated exclusively within defined context of the head attribute. Subsequent attributes have no meaning or purpose outside of those explicitly defined within the head attributes specification. Section 4 provides descriptions of each head attribute defined by this text.

In response to an Extended-Request packet sent to RADIUS server an Extended-Response or Extended-Reject packet is returned to the client indicating result of Extended-Request.

Extended-Response packets may include one or more response attributes. Extended-Reject packets indicate failure. Extended-Reject packets include failure attributes to communicate diagnostic information to the RADIUS client.

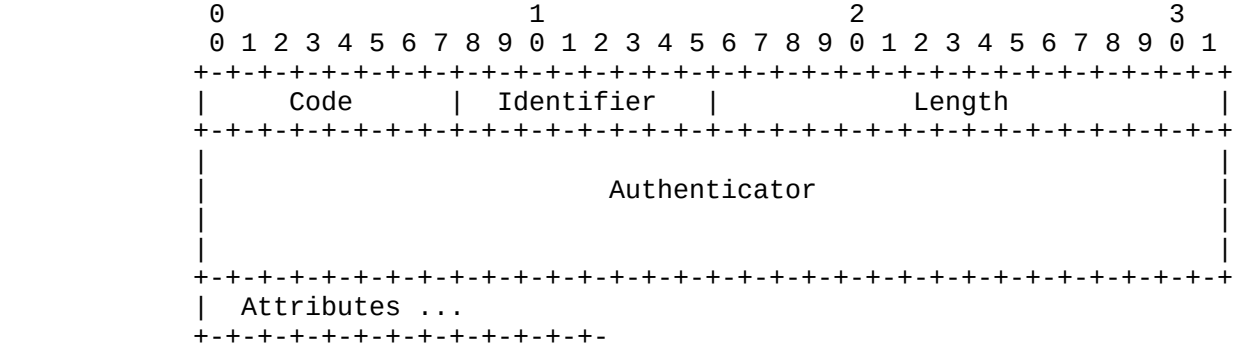
A RADIUS server not supporting a head attribute responds with Extended-Reject containing attribute Error-Cause = "Unsupported Extension" [RFC5176]. RADIUS servers MUST NOT make any attempt to use or interpret attributes subsequent to the head attribute in the event head attribute is unknown or not supported. All such attributes MUST be ignored.

Should RADIUS server receive an Extended-Request packet with no attributes or receive a request containing Missing attributes per the specification of a supported head attribute an Extended-Reject message is returned containing attribute Error-Cause = "Missing Attribute"

3.1. Packet Format

Packet format consists of the fields: Code, Identifier, Length, Authenticator and optional Attributes. All fields hold the same meaning as those described in RADIUS [RFC2865].

Authenticator field is calculated using same method specified for Accounting-Request and Accounting-Response packets [RFC2866].



Code

The Code field is one octet, and identifies the type of RADIUS packet. RADIUS codes described in this document are assigned as follows:

TBD - Extended-Request
TBD - Extended-Response
TBD - Extended-Reject

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 when transmitted over UDP.

Authenticator

The Authenticator field is sixteen (16) octets. This value is used to authenticate packets between the RADIUS server and client.

Request Authenticator

The Authenticator field in a Request packet (e.g. Extended-Request) is called the Request Authenticator. The Request Authenticator is calculated the same way as for an Accounting-Request packet specified in [RFC2866].

Response Authenticator

The Authenticator field in a Response packet (e.g. Extended-Response or Extended-Reject) is called the Response Authenticator. This field is calculated the same way as for an Accounting-Response packet specified in [RFC2866].

Attributes

Attributes may have none or multiple instances.

4. Extended Request head attributes

Head attributes always appear immediately after RADIUS header within an Extended-Request and conditionally within Extended-Response packets. Each head attribute defines purpose and usage of Extended-Request and Extended-Response packets.

4.1. Fragment-Data

Fragment-Data head attribute describes a method for encapsulating a large RADIUS packet within series of smaller Extended-Request packets and later reassembling the encapsulated packet.

In this section "outer packet" refers to the Extended-Request or Extended-Response RADIUS packet acting as an envelope for the encapsulated RADIUS packet. "Inner packet" refers to the encapsulated packet.

4.1.1. Client to server packet fragmentation

This section describes method for RADIUS client to fragment and transmit request packets (e.g. Access-Request) to server.

4.1.1.1. Prerequisites

RADIUS client SHALL meet one or more of the following requirements before sending a fragmented request:

1. Administrative setting indicating RADIUS server support for fragments. If client has previously received successful Fragment-Inquire response either omitting Fragment-Limit or consisting of value <= 4096 bytes the administrative setting is ignored, a fragmented request MUST NOT be sent.

2. Fragment-Inquire response containing Fragment-Limit having value > 4096.

4.1.1.2. Preparing inner request packet

RADIUS client generates a complete RADIUS request packet in a storage buffer rather than transmitting to RADIUS server. Prior to generating request packet following changes to normal processing SHALL be observed:

1. Identifier field of RADIUS request is set 0. This field is unused while inner packet is encapsulated in outer packet.

2. 4096 byte maximum packet length [RFC2865] limit is replaced with lower of following three constraints:

   A. Maximum Length RADIUS header field can accommodate. (e.g. 65535 bytes)

   B. Value of Fragment-Limit obtained by prior Fragment-Inquire request.

   C. Administrative limit.

4.1.1.3. Generating outer request packet

Once inner packet is generated it is fragmented and transmitted to RADIUS server in a series of outer Extension-Request/Extension-Response packet exchanges.

For each fragment following method is used to construct outer packet sent to the RADIUS server:

1. Fragment-Data head attribute added to Extension-Request packet.

2. Inner packet header authenticator field copied to Fragment-Data "Inner Authenticator" field. See section 4.1.5 for Fragment-Data format.

3. Inner packets header code field copied to Fragment-Data "Inner Code" field.

4. Fragment-Data Sequence field is incremented. First fragment starts at 1.

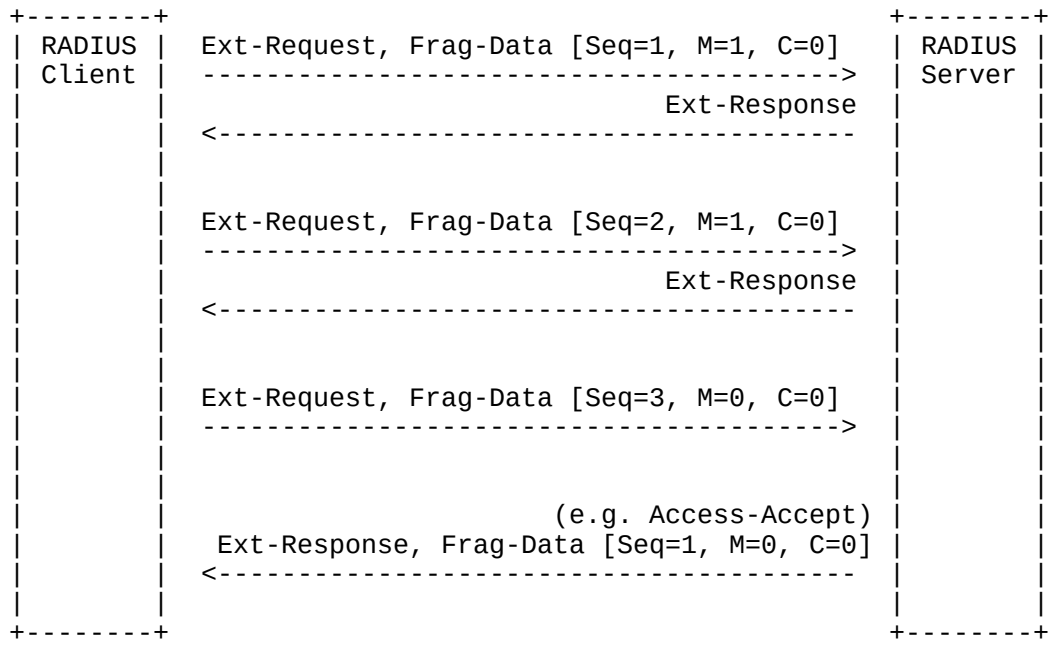
5. If there are more fragments to transmit Fragment-Data M bit is set 1. M bit is set 0 if current fragment is last.

6. Attributes are appended to outer packet from inner packet until either per-fragment length limit is reached or there are no more attributes remaining within inner packet. Attributes are copied in order they appear within inner packet without modification. An attribute is copied to outer packet only while there is enough space remaining to accommodate the full attribute. If sufficient space is unavailable attribute is sent with the next fragment.

7. Request authenticator of outer packet calculated normally per [RFC2866]

4.1.1.4. Transmitting fragments to RADIUS server

Once an outer packet is sent server responds with Extended-Response acknowledging receipt. The process is repeated until all fragments are delivered. Receipt of final fragment is acknowledged by RADIUS Servers response to assembled inner packet rather than a final Extended-Response.



4.1.1.5. Server state and fragment validation

RADIUS servers processing fragmented requests from clients SHALL perform the following validation steps:

1. Request authenticator of outer packet validated per [RFC2866]. If validation fails the outer packet is silently discarded.

2. Attributes contained within outer packet must be well formed otherwise outer packet is silently discarded.

3. Upon receipt of first fragment Fragment-Data head attribute sequence field shall be 1. In this case server allocates state necessary to uniquely track and assemble this and all subsequent fragments using Fragment-Data excluding Sequence and Flag fields as a unique identifier. If a request is received

in which sequence number is greater than 1 and for which no state exists the outer packet responds with Extended-Reject packet containing attribute Error-Cause = "Invalid Request"

4. If Continuation bit is 1 or Sequence number is 0 within Fragment-Data head attribute outer packet is silently discarded.

5. If "Inner code" field is Extended-Request, Extended-Response or Extended-Reject the outer packet is silently discarded.

6. If a request is received in which Fragment-Data sequence number is less than sequence of previously accepted fragment or does not contain the next expected sequential value the outer packet is silently discarded.

7. If outer packet is less than 400 bytes and More bit is 1 or if storing this fragment would cause total inner packet length to exceed the set inner packet limit then state for this request is removed and outer packet responds with Extended-Reject packet containing attribute Error-Cause = "Administratively Prohibited"

In any case described above where outer packet is silently discarded this MUST NOT trigger release of stored state for fragment assembly.

4.1.2. Server to client packet fragmentation

This section describes method for RADIUS server to fragment and transmit response packets (e.g. Access-Accept) to client.

4.1.2.1. Prerequisites

A RADIUS server SHALL meet one or more of the following requirements before sending a fragmented response.

1. Administrative option indicating client support for fragments.

2. Request packet from RADIUS client delivered using fragments.

3. Request contained Fragment-Reply-Supported attribute having value of Yes (4000).

4.1.2.2. Preparing inner response packet

RADIUS server generates a complete RADIUS response packet in a storage buffer rather than transmitting to RADIUS client. Prior to

generating response packet following changes to normal processing SHALL be observed:

1. Identifier field of RADIUS response is set 0. This field is unused while inner packet is encapsulated in outer packet.

2. 4096 byte maximum packet length [RFC2865] limit is replaced with lower of following two constraints:

   A. Maximum Length RADIUS header field can accommodate. (e.g. 65535 bytes)

   B. Administrative limit.

4.1.2.3. Generating outer response packet

Once inner packet is generated it is fragmented and transmitted to RADIUS client in a series of outer Extension-Response/Extension-Request packet exchanges.

For each fragment following method is used to construct outer packet sent to the RADIUS client:

1. Fragment-Data head attribute added to Extension-Response packet.

2. Inner packet header authenticator field copied to Fragment-Data "Inner Authenticator" field. See section 4.1.5 for Fragment-Data format.

3. Inner packets header code field copied to Fragment-Data "Inner Code" field.

4. Fragment-Data Sequence field is incremented. First fragment starts at 1.

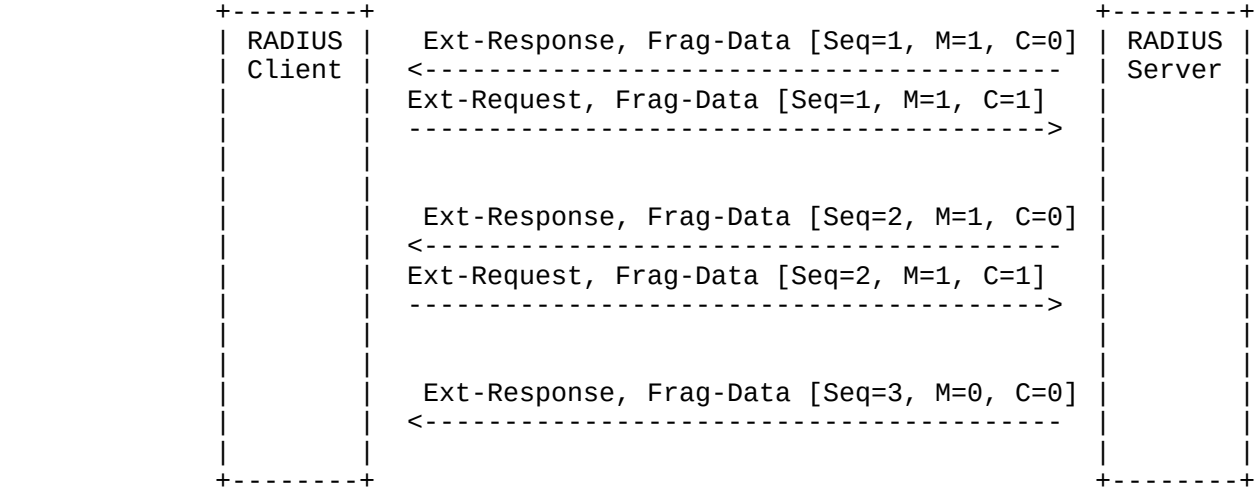
5. If there are more fragments to transmit Fragment-Data M bit is set 1. M bit is set 0 if current fragment is last.

6. Attributes are appended to outer packet from inner packet until either per-fragment length limit is reached or there are no more attributes remaining within inner packet. Attributes are copied in order they appear within inner packet without modification. An attribute is copied to outer packet only while there is enough space remaining to accommodate the full attribute. If sufficient space is unavailable attribute is sent with the next fragment.

7. Response authenticator of outer packet calculated normally per [RFC2866]

4.1.2.4. Transmitting fragments to RADIUS client

Once an outer packet (Extended-Response) is sent RADIUS client responds by issuing an Extended-Request containing Fragment-Data head attribute copied from response with Continuation bit set to 1. The process is repeated until all fragments are delivered. Receipt of final fragment is unacknowledged.



4.1.2.5. Client state and fragment validation

RADIUS clients processing fragmented responses from servers SHALL perform the following validation steps:

1. Response authenticator of outer packet validated per [RFC2866]. If validation fails outer packet is silently discarded.

2. Attributes contained within outer packet must be well formed otherwise outer packet is silently discarded.

3. Upon receipt of first response fragment Fragment-Data head attribute sequence field shall be 1 otherwise the outer packet is silently discarded.

4. If Continuation bit is 1 within Fragment-Data head attribute outer packet is silently discarded.

5. If "Inner code" field is Extended-Request, Extended-Response or Extended-Reject the outer packet is silently discarded.

6. If a response is received in which Fragment-Data sequence number is less than sequence of previously accepted fragment or does not contain the next expected sequence value outer packet is silently discarded.

4.1.3. Inner packet reassembly

Once all fragments are received then inner packet is assembled by reversing fragmentation process.

Inner packet RADIUS header is generated as follows:

1. Code = Fragment-Data "Inner Code" field

2. Identifier = 0

3. Length = RADIUS header length (e.g. 20) + sum of all attributes within all fragments excluding Fragment-Data head attribute of each fragment.

4. Authenticator = Fragment-Data "Inner Authenticator" field

Attributes are appended to packet in order received excluding Fragment-Data head attribute of each fragment.

Fully assembled inner packet is processed normally as if packet were received from the network including validation of applicable authenticator fields.

4.1.4. Request response options

RADIUS clients supporting fragmentation MUST be capable of accepting a fragmented response in reply to a non-fragmented request.

If a request is fragmented response SHALL also be fragmented.

4.1.4.1. Fragmented response to non-fragmented request

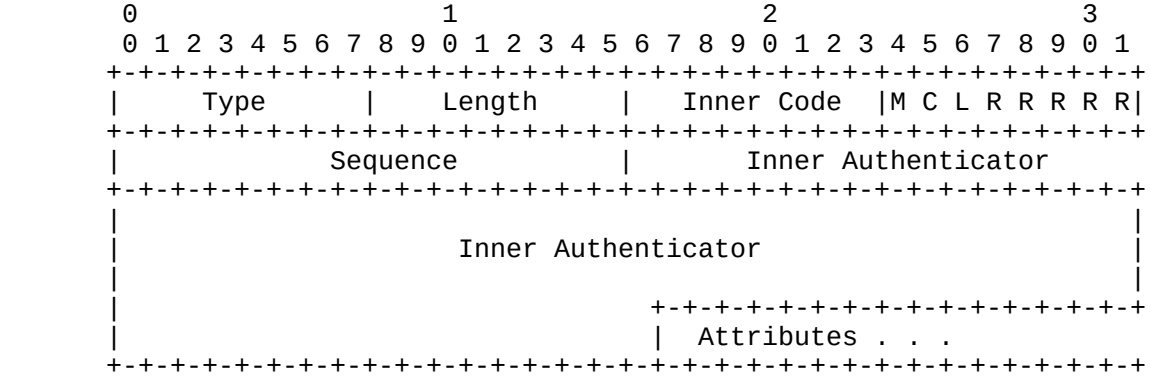
The non-fragmented requests authenticator is used both during production of the inner RADIUS response packet and production of the first outer packet carrying first fragmented response.

When client reassembles fragmented response to obtain inner packet original request authenticator from non-fragmented request is used to validate the inner packets response authenticator.

4.1.4.2. Fragmented response to fragmented request

Inner and outer packets authenticators are separate.

4.1.5. Fragment-Data head attribute format



Type

TBD for Fragment-Data

Length

24

Inner Code

Represents inner packet header code field

Flags

M - More

1 = Additional fragments pending
0 = Final fragment

C - Continuation

1 = RADIUS client requests next response fragment
0 = Packet contains inner packet fragments

L - Reserved

This field is reserved for future use and MUST be set 0.

R - Reserved

These fields are reserved for future use and MUST be set 0.

Sequence

The 16-bit unsigned fragment sequence number in network byte order.

Inner Authenticator

Represents inner packet header authenticator field

4.1.6. Server state management

RADIUS servers are required to keep state to facilitate assembly of fragments into completed inner packets. This process occurs between client and server and cannot be proxied. RADIUS servers may do no processing of a request until after the completed inner packet has been transmitted. These constraints allow selection of a low threshold for retention of state to account for worse case round trip delays, server delay and client retransmission policy. It is RECOMMENDED state be kept for no longer than 30 seconds after last successfully received fragment.

It is recommended RADIUS servers develop a robust policy for managing state to guard against resource exhaustion. One method of accomplishing this is to alter state retention period proportionally to pressure on state resources until a lower threshold is reached. Beyond this if state is exhausted the RADIUS server MAY respond to new fragment requests by sending Extended-Reject containing attribute Error-Cause = "Resources Unavailable". Upon receipt clients MAY attempt to contact an alternate RADIUS server if available.

4.1.7. Per-fragment and inner packet sizing

Minimum length of an outer packet shall be 400 bytes when the More bit is set 1.

Maximum acceptable length of an outer packet varies between 400 and 4096 bytes inclusive for UDP or 65535 bytes for TCP. Using Fragment-Inquire request described in section 4.2 clients MAY obtain

Framed-MTU hint from RADIUS server to size request fragments on IP packet boundaries.

For clients to signal MTU to server client includes Framed-MTU hint in fragmented or non-fragmented Access-Request packet.

RADIUS packets SHOULD only be fragmented if they would exceed 4096 bytes in length. It is not recommended MTU hinting be used as a threshold to fragment attributes.

It is RECOMMENDED by default servers support inner packets up to 65535 bytes in length. Administrative settings SHOULD be available to allow operators to change the limit.

4.2. Fragment-Inquire

Fragment-Inquire requests optional fragment related capabilities and parameters from the RADIUS server.

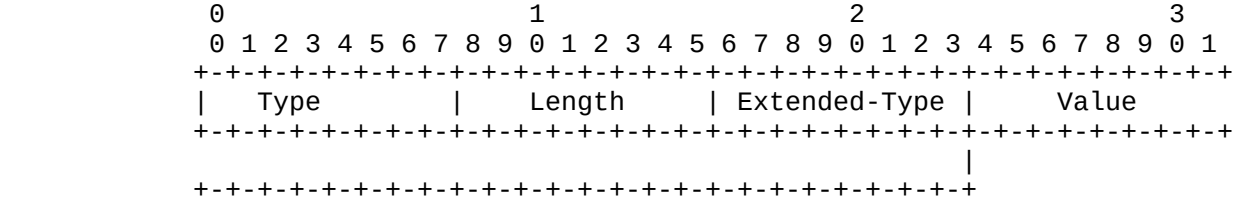
4.2.1. Request

When RADIUS is used over a connection based transport (e.g. TLS/DTLS) Fragment-Inquire requests are sent upon initial connection. When used over a connectionless transport (e.g. UDP) Fragment-Inquire requests are sent upon startup or before issuing first RADIUS request.

If RADIUS is used over a connection based transport following additional attributes MAY be included after the Fragment-Inquire head attribute.

```
  Fragment-Stream-Limit    (Section 6.4)
  Fragment-Reply-Supported (Section 6.1)
```

If RADIUS is used over a connectionless transport no additional attributes SHALL be sent.



Type

TBD

Length

7

Extended-Type

TBD

Value (Integer)

1

4.2.2. Response

In response to Fragment-Inquire the following attributes MAY be present within Extended-Response packet.

Fragment-Reply-Allowed	Section 6.2
Fragment-Stream-Limit	Section 6.3
Fragment-Limit	Section 6.4
Fragment-Inquire-Interval	Section 6.5
Framed-MTU	Section 6.6 [RFC2865]
Event-Timestamp	Section 6.7 [RFC2869]

5. Compatibility

While procedures described in this text provide a means to manage compatibility amongst servers and clients capable systems are still required to exchange RADIUS packets beyond 4096 bytes. Clients and Servers may not send or process large packets unless both client and server support large packets. Likewise proxy systems may not process large packets unless they are capable of processing them.

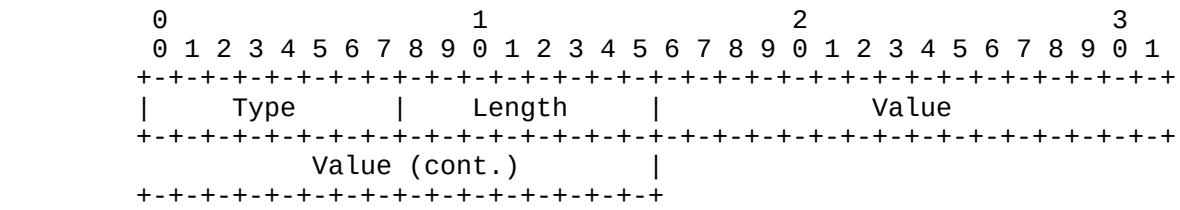
If a RADIUS server is part of a proxy network without support for packets larger than 4096 bytes it SHOULD NOT advertise support for fragmentation or TCP large packet support via Fragment-Inquire. In this case clients and servers should seek to minimize the need for large packets by other means (e.g. limiting capability, out-of-band signals or compression) until such time remaining systems can be upgraded.

6. Attributes

6.1. Fragment-Reply-Supported

When sent within an Access-Request packet this attribute signals client support for fragmented response of packets beyond 4096 bytes. (e.g. Access-Accept).

This attribute MUST only be transmitted from client after a successful Fragment-Inquire request contains attribute Fragment-Reply-Allowed=1 within its response. This attribute MUST NOT be administratively configured nor SHALL it be forwarded for proxy unless the same procedure is followed.



Type

TBD

Length

6

Value (Integer)

4000 = Yes
All other values = No

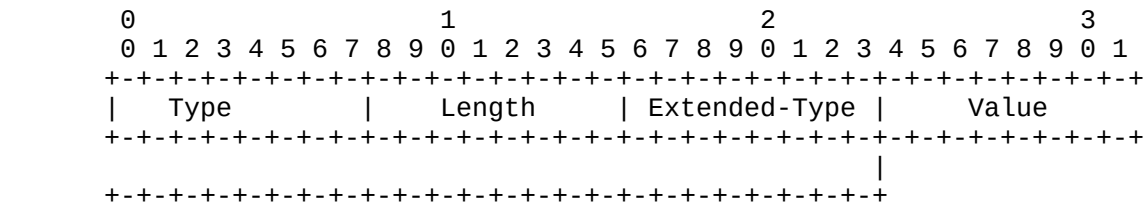
6.2. Fragment-Reply-Allowed

If RADIUS server supports Fragment-Reply-Supported attribute described in section 6.1 then Fragment-Reply-Allowed is sent within Extended-Response to Fragment-Inquire head attribute.

The server MUST meet following requirements before this attribute can be sent.

1. Server supports Fragment-Data head attribute or is using TCP transport supporting RADIUS packets exceeding 4096 bytes.

2. Server will not forward Fragment-Reply-Supported attribute for proxy unless same client procedure described in section 6.1 has been followed toward forward proxy destination.



Type

TBD

Length

7

Extended-Type

TBD

Value (Integer)

1

6.3. Fragment-Stream-Limit

When sent in response to Fragment-Inquire head attribute this attribute signals to RADIUS clients maximum (non-fragmented) RADIUS packet length in bytes accepted by server over a stream (e.g. TCP) transport.

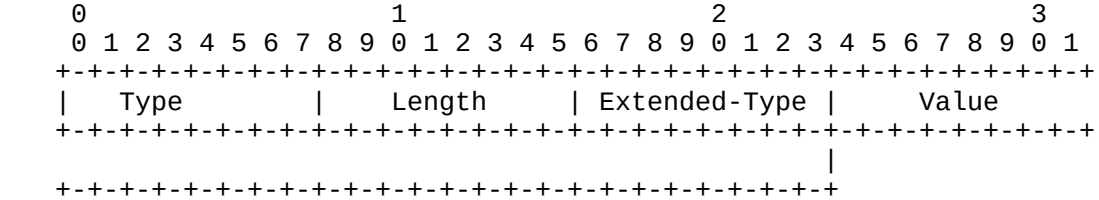
This attribute MUST NOT be sent if RADIUS server does not support any stream transports or is otherwise unable to offer support to the client. Attribute MUST NOT be sent if maximum packet length supported by RADIUS server over stream transport is 4096 bytes or less.

When sent to UDP clients this provides a hint client MAY establish a stream connection while there is a need to send requests larger than 4096 bytes or where a large response is anticipated.

When sent to non-UDP clients the client MUST NOT attempt to send a request packet larger than indicated by Fragment-Stream-Limit.

A RADIUS client SHOULD only switch from UDP if protocol being switched to offers the same or higher level of security.

When sent within a Fragment-Inquire head attribute this signals to server that client is capable of receiving response packets up to length indicated by Fragment-Stream-Limit. If this attribute was sent by a UDP client it MUST be ignored.



Type

TBD

Length

7

Extended-Type

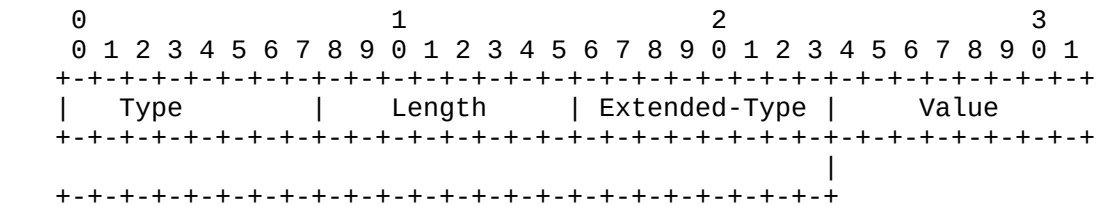
TBD

Value (Integer)

> 4096 (bytes)

6.4. Fragment-Limit

When sent in response to Fragment-Inquire head attribute this signals maximum fragmented inner packet length in bytes server is willing to accept and advertises to clients that fragmentation is supported. If Fragment-Limit is sent it MUST be greater than 4096 bytes.



Type

TBD

Length

7

Extended-Type

TBD

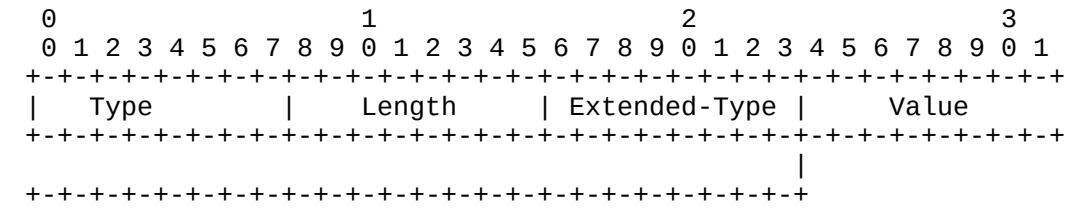
Value (Integer)

>4096 bytes

6.5. Fragment-Inquire-Interval

When sent in response to Fragment-Inquire head attribute this attribute controls interval in seconds at which additional Fragment-Inquire requests should be sent in order for clients to be made aware of configuration changes. If the attribute is not set clients are not expected to check periodically for configuration change.

Fragment-Inquire-Interval SHOULD be no less than 60 seconds.



Type

TBD

Length

7

Extended-Type

TBD

Value (Integer)

>= 60

6.6. Framed-MTU

When sent in response to Fragment-Inquire head attribute Framed-MTU [RFC2865] provides client a hint as to server MTU. (See also section 4.1.7)

6.7. Event-Timestamp

When sent in response to Fragment-Inquire head attribute Event-Timestamp [RFC2869] reflects time at which server generated Extended-Response packet.

7. Table of Attributes

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

	Auth			Acct			Disconnect			CoA			Extended-Req		
	REQ	ACK	NAK	REQ	ACK		REQ	ACK	NAK	REQ	ACK	NAK	REQ	ACK	NAK
Fragment-Data	0	0	0	0	0		0	0	0	0	0	0	0-1	0-1	0
Fragment-Inquire	0	0	0	0	0		0	0	0	0	0	0	0-1	0	0
Fragment-Reply-Supported	0-1	0	0	0	0		0	0	0	0	0	0	0-1	0	0
Fragment-Reply-Allowed	0	0	0	0	0		0	0	0	0	0	0	0	0-1	0
Fragment-Stream-Limit	0	0	0	0	0		0	0	0	0	0	0	0-1	0-1	0
Fragment-Limit	0	0	0	0	0		0	0	0	0	0	0	0	0-1	0
Fragment-Inquire-Interval	0	0	0	0	0		0	0	0	0	0	0	0	0-1	0

0 This attribute MUST NOT be present in packet.

0-1 Zero or one instance of this attribute MAY be present in packet.

8. Security Considerations

An attacker can collect and replay previous fragment exchanges in a bid to overwhelm server resources or cause previous packets to be reprocessed.

All security considerations which apply to confidentiality, integrity and availability of RADIUS Accounting [RFC2866] packets apply to this document.

9. IANA Considerations

This document defines following RADIUS attribute types and packet codes.

Protocols / RADIUS Types / RADIUS Packet Type Codes

- TBD - Extended-Request
- TBD - Extended-Response
- TBD - Extended-Reject

Protocols / RADIUS Types / RADIUS Attribute Types (Standard Space)

- TBD - Fragment-Data
- TBD - Fragment-Reply-Supported

Protocols / RADIUS Types / RADIUS Attribute Types (Extended Space)

- TBD - Fragment-Inquire
- TBD - Fragment-Reply-Allowed
- TBD - Fragment-Stream-Limit
- TBD - Fragment-Limit
- TBD - Fragment-Inquire-Interval

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W.

[RFC2866] Rigney, C., "RADIUS Accounting", RFC 2866, June 2000.

[RFC2869] RADIUS Extensions C. Rigney, W. Willats, P. Calhoun June 2000

[RFC5176] Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS) M. Chiba, G. Dommety, M. Eklund, D. Mitton, B. Aboba, January 2008

[RFC6613] RADIUS over TCP A. DeKok, May 2012

[RFC6614] Transport Layer Security (TLS) Encryption for RADIUS S. Winter, M. McCauley, S. Venaas, K. Wierenga, May 2012

[RFC6929] Remote Authentication Dial In User Service (RADIUS) Protocol Extensions. A. DeKok, A. Lior. April 2013.

11. Acknowledgments

Author would like to thank Sam Hartman for valuable ideas.

Authors' Addresses

Peter Deacon
IEA Software, Inc.
P.O. Box 1170
Veradale, WA 99037
USA

Email: peterd@iea-software.com