RADIUS EXTensions Working Group
Internet-Draft
Intended status: Experimental
Expires: July 4, 2012

A. Perez-Mendez
R. Marin-Lopez
F. Pereniguez-Garcia
G. Lopez-Millan
University of Murcia
D. Lopez
Telefonica I+D

Jan 2012

Fragmentation support across RADIUS packets draft-perez-radext-radius-fragmentation-00

Abstract

This document describes a mechanism providing fragmentation support of RADIUS attributes across several RADIUS packets. This is intended to support attributes that exceed the 4 KB limit per RADIUS packet.

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). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on July 4, 2012.

Table of Contents

1. Introduction

RADIUS [RFC2865] is a protocol for carrying authentication, authorization, and configuration information between a Network Access Server (NAS) which desires to authenticate its links and a shared Authentication Server (AS). Information is exchanged between the NAS and the AS through packets. Each RADIUS packet can transport several RADIUS attributes, to convey the necessary information to the other peer, up to a maximum size of 4K of total data (including RADIUS packet headers). RADIUS attributes have a maximum size of 253 bytes of payload.

RADIUS has been extensively used along the years. Along this time, the need of sending RADIUS attributes larger than 253 bytes has become a reality. An immediate alternative to overcome this issue consists in splitting the data into a group of RADIUS attributes of the same type, and then insert them into the RADIUS packet in order. At the destination, the content of these attributes is extracted and joined to rebuild the original data. This scheme is followed, for example, by RADIUS-EAP [RFC3579]. Another more general solution is given in [I-D.ietf-radext-extended-attributes].

However, there are no proposals to deal with attributes that exceed the 4K limit imposed by the maximum RADIUS packet length. As the usage of RADIUS is considered in more complex AAA scenarios, including the exchange of richer data, like SAML assertions or JWT tokens, exceeding this limit becomes more likely, thus making necessary the availability of mechanisms for dealing with this situation.

This document defines an extension to allow RADIUS peers to exchange attributes that exceed the 4 KB limit of the RADIUS packets, by fragmenting them across several packets, trying to maintain compatibility with any intra-packet fragmentation mechanisms and with the existing RADIUS deployments.

1.1. Requirements Language

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].

2. Chunked-Attribute

This document proposes the definition of a new RADIUS attribute, called Chunked-Attribute. The presence of this attribute indicates that a certain piece of information carried in one (or several)

RADIUS attribute(s) exceeds the maximum size of the RADIUS packet. The information is organized into chunks, whose characteristics are defined by the Chunked-Attribute attribute. A chunk is a collection of RADIUS attributes of the same type (fragments) that represents a portion of the original piece of information (being carried through several RADIUS packets) that is obtained by concatenating the content of each RADIUS attribute. The following figure represents the format of the Chunked-Attribute attribute.

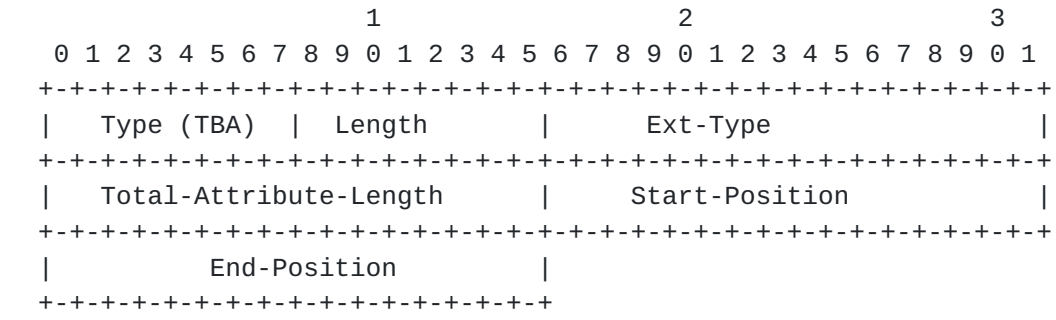

Figure 1: Chunked-Attribute

Type

To be assigned (TBA)

Length

10

Ext-Type

Type of the attribute that is being chunked. This field is encoded as an Ext-Type value, as described in [I-D.ietf-radext-extended-attributes], to be compatible with extended attributes.

Total-Attribute-Length

The total length of the attribute data, before being chunked.

Start-Position

The position in the original attribute data (in bytes) where this chunk starts (inclusive). It can take values from 0 to Total-Attribute-Length - 1.

End-Position

The position in the original attribute data (in bytes) where this chunk ends (inclusive). It can take values from 0 to Total-Attribute-Length - 1. It MUST be always >= Start-Position.

A Chunked-Attribute can be included in either an Access-Request or an Access-Challenge packet. It can also appear in an Access-Accept or Access-Reject packet, but only if it refers to the last chunk of a sequence (i.e End-Position = Total-Attribute-Length - 1).

3. Use in Access-Request packets

When the NAS desires to send an attribute that is too large to be completely included into an Access-Request packet, the attribute can be split into several chunks and sent over different Access-Request packets. This fact is indicated by including a Chunked-Attribute attribute. The process is described in detail using the following example. In this example, the intra-packet fragmentation has been performed through the extension of the attribute value into several attributes of the same type, as done in RADIUS-EAP [RFC3579]. If other intra-packet fragmentation model is used, calculations may differ due to the difference in the effective payload size.

- The NAS wants to send a RADIUS attribute of type X and an associated value size of 2000 bytes to the AS. After fragmenting the attribute, it results into 8 RADIUS attributes of type X. The first 7 have a size of 255 bytes (253 of payload and 2 of header), while the last one has a size of 231 (229 of payload and 2 of header).

- Due to the presence of other RADIUS attributes, let us suppose that the free space in the Access-Request packet is only 1000 bytes. Hence, only 3 of these attributes can be included without exceeding the limit. Thus, the NAS includes a Chunked-Attribute in the packet, indicating the Ext-Type of the attribute to be included (X), the Total-Attribute-Length (2000), the Start-Position(0), and the End-Position (3 * 253 = 759).

- After the Chunked-Attribute, the NAS includes the first 3 attributes of type X, and sends the packet to the AS.

- When the AS sees the Chunked-Attribute, it knows that an incomplete attribute is present in the packet. It allocates enough space to hold the complete attribute (based on Total-Attribute-Length) and starts filling it with the received data (759 bytes of payload). Consistency between the values indicated in the Chunked-Attribute and the actually received data is checked. The AS will delay the processing of the received RADIUS

attributes (even those placed in the packet before the Chunked-Attribute) until all the remaining parts have been received.

- To obtain the remaining chunks, the AS sends an Access-Challenge packet to the NAS, in response to the Access-Request. This packet MUST only include an State attribute, that MUST be sent back by the NAS in the next Access-Request packet, to link the packets that belong to the same conversation.

- When the NAS receives the Access-Challenge packet, it replies with an Access-Request packet including the received State attribute, a new Chunked-Attribute, and the remaining attributes of type X. Specifically, the Chunked-Attribute indicates the Ext-Type of the attribute (X), the Total-Attribute-Length (2000), the Start-Position (760) and the End-Position (1999). After that, the remaining 5 attributes of type X are included, making a total of 1241 bytes of payload.

- When the AS receives the Access-Request packet, it concatenates the different RADIUS attributes to complete the data of length 2000 associated to the attribute of type X. Once the chunked attribute is completed, the AS can process the received packet as if all the attributes had been received at once. The AS will generate a response according to the content of those attributes, as usual.

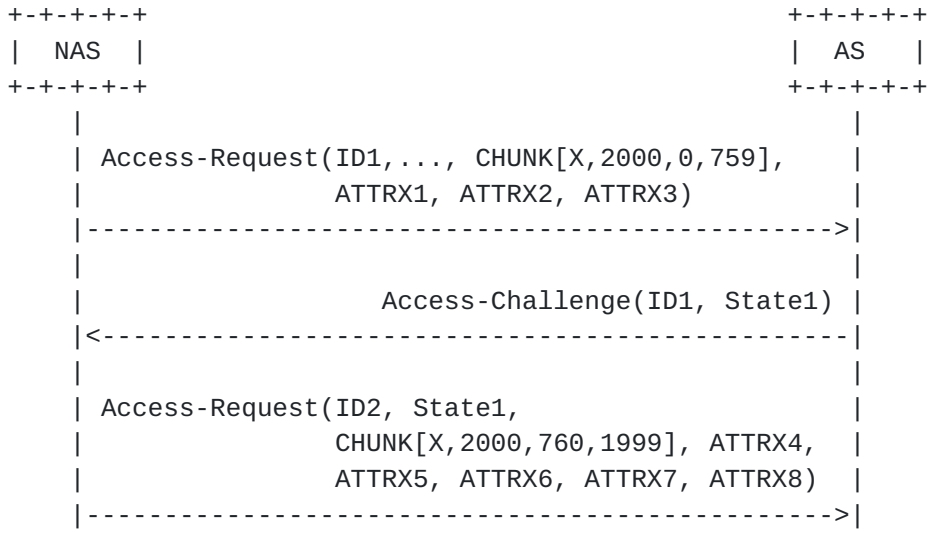

Figure 2: Access-Request chunked attributes

4. Use in Access-Challenge packets

When the AS desires to send a RADIUS attribute that does not fit into the available free space of an Access-Challenge packet, the attribute is split into chunks in a similar way as described in Section 3. The following example describes the process step by step.

- The AS wants to send a RADIUS attribute of type X and size 2000 bytes to the NAS. After fragmenting the attribute, it results into 8 RADIUS attributes of type X. The first 7 have a size of 255 bytes (253 of payload and 2 of header), while the last one has a size of 231 (229 of payload and 2 of header).

- There are 1000 bytes of free space in the Access-Challenge packet, thus 3 of these attributes can be included without exceeding the limit. The AS includes a Chunked-Attribute in the packet, indicating the Ext-Type of the attribute to be included (X), the Total-Attribute-Length (2000), the Start-Position(0), and the End-Position (3 * 253 = 759).

- After the Chunked-Attribute, the NAS includes the first 3 attributes of type X.

- The AS also includes a State attribute that MUST be sent back by the NAS in the next Access-Request packet. This is required to tie together all the packets belonging to the same conversation. Then the AS sends the Access-Challenge packet to the NAS.

- When the NAS receives the Chunked-Attribute, it knows that an incomplete attribute is present in the packet. It allocates enough space to hold the complete attribute (based on Total-Attribute-Length) and starts filling it with the received data (759 bytes of payload). Consistency between the values indicated in the Chunked-Attribute and the actual received data is checked. The NAS will delay the processing of the received RADIUS attributes (even those placed in the packet before the Chunked-Attribute) until all the remaining parts have been received.

- To obtain the remaining chunks, the NAS sends an Access-Request packet to the AS. This packet MUST include the State attribute obtained from the previous Access-Challenge packet.

- When the AS receives the Access-Request packet it replies with another Access-Challenge, including a new Chunked-Attribute and the remaining attributes of type X. If it were not possible to include all the remaining attributes in this packet, a new State attribute would be sent to the NAS. In this example, the Chunked-Attribute indicates the Ext-Type of the attribute (X), the Total-

Attribute-Length (2000), the Start-Position (760) and the End-Position (1999). After that, the remaining 5 attributes of type X are sent, making a total of 1241 bytes of payload.

Note well that the AS MAY send an Access-Accept or Access-Reject packet instead of an Access-Challenge at this point, if it has already completed the process originally required by the NAS.

- When the NAS receives this packet, it completes the data associated to the attribute of type X. Once the attribute is completed, the NAS can process all the received attributes (including those that were not chunked), as if all of them had been received in the same packet. The NAS can continue with the RADIUS protocol as determined by the processing of the received attributes, as usual.

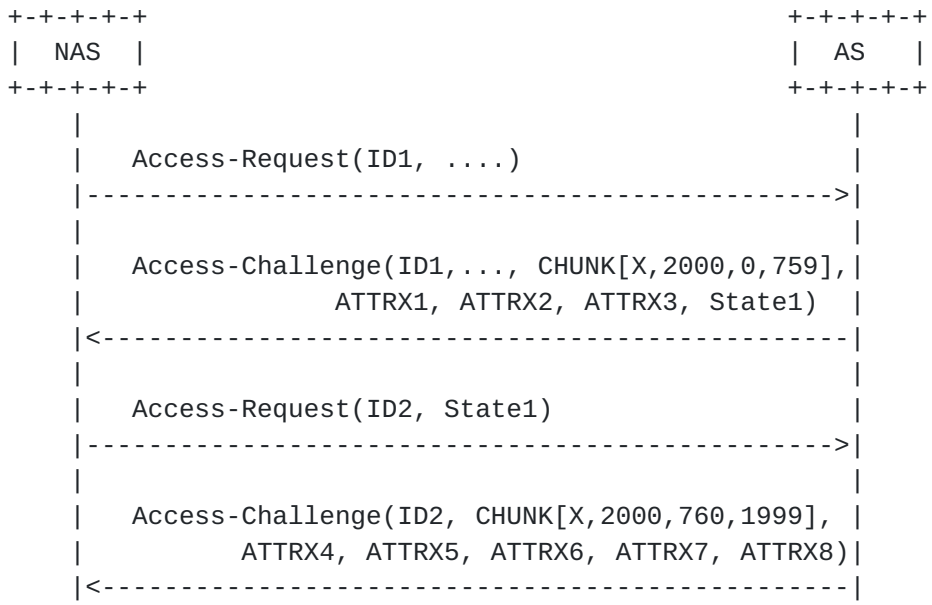

Figure 3: Access-Challenge chunked attributes

5. Security Considerations

6. IANA Considerations

This document has no actions for IANA.

7. Normative References

[I-D.ietf-radext-extended-attributes]
Li, Y., Lior, A., and G. Zorn, "Extended Remote Authentication Dial In User Service (RADIUS) Attributes", draft-ietf-radext-extended-attributes-09 (work in progress), May 2010.

[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. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

[RFC3579] Aboba, B. and P. Calhoun, "RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)", RFC 3579, September 2003.

Authors' Addresses

Alejandro Perez-Mendez (Ed.)
University of Murcia
Campus de Espinardo S/N, Faculty of Computer Science
Murcia, 30100
Spain

Phone: +34 868 88 46 44
Email: alex@um.es

Rafa Marin-Lopez
University of Murcia
Campus de Espinardo S/N, Faculty of Computer Science
Murcia, 30100
Spain

Phone: +34 868 88 85 01
Email: rafa@um.es

Fernando Pereniguez-Garcia
University of Murcia
Campus de Espinardo S/N, Faculty of Computer Science
Murcia, 30100
Spain

Phone: +34 868 88 78 82
Email: pereniguez@um.es

Gabriel Lopez-Millan
University of Murcia
Campus de Espinardo S/N, Faculty of Computer Science
Murcia, 30100
Spain

Phone: +34 868 88 85 04
Email: gabilm@um.es

Diego R. Lopez
Telefonica I+D
Don Ramon de la Cruz, 84
Madrid, 28006
Spain

Phone: +34 913 129 041
Email: diego@tid.es