Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: November 1, 2010

Y. Li
A. Lior
BWS
G. Zorn, Ed.
Network Zen
May 13, 2010

Extended Remote Authentication Dial In User Service (RADIUS) Attributes

draft-ietf-radext-extended-attributes-09.txt

Abstract

For the Remote Authentication Dial In User Service (RADIUS) protocol to continue to support new applications, the RADIUS attribute type space must be extended beyond the current limit of 255 possible attribute types while maintaining backwards compatibility with the existing protocol. This document defines a mechanism to accomplish that task, along with standard methods to group together related attributes and to encode values that don't fit into 253 octets.

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 November 11, 2010.

Table of Contents

1. Introduction

The Remote Authentication Dial In User Service (RADIUS) Protocol [RFC2865] defines two classes of attributes: standard and vendor-specific.

Vendor-specific Attributes (VSAs) allow vendors (including Standards Development Organizations (SDOs)) to define their own Attributes, which may not be suitable for general usage; on the other hand, the attributes that belong to the standard RADIUS space are controlled by the IETF and are intended to be of general utility. These attributes are defined in RFCs and are assigned type codes by the Internet Assigned Number Authority (IANA)[IANA].

The standard RADIUS attribute type code is 8 bits in length; hence RADIUS is limited to 255 attribute types. Of these 255 attribute types, approximately 101 have been assigned as of this writing. According to RFC 3575 [RFC3575], types 192-223 are reserved for experimental use; types 224-240 are reserved for implementation-specific use; and values 241-255 are reserved and should not be used. Therefore, as of this writing there are approximately 90 type codes that can be allocated to new attributes.

RADIUS evolution must not be hindered by the inability to define new standard RADIUS attributes. This document defines a mechanism to extend the standard RADIUS Attribute space by defining a new scheme to allocate attribute type codes. In addition, mechanisms are defined to support both the grouping of related attributes and the encoding of attribute values the length of which exceed the current limit of 253 octets.

2. Terminology

Extended Attribute
: The term used for the new RADIUS attributes that are defined in this document

Extended Type
: The type code assigned to an Extended Attribute

2.1. Requirements Language

In this document, several words are used to signify the requirements of the specification. These words are often capitalized. The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

An implementation is not compliant if it fails to satisfy one or more of the must or must not requirements for the protocols it implements. An implementation that satisfies all the MUST, MUST NOT, SHOULD, and SHOULD NOT requirements for its protocols is said to be "unconditionally compliant"; one that satisfies all the MUST and MUST NOT requirements but not all the SHOULD or SHOULD NOT requirements for its protocols is said to be "conditionally compliant".

3. Problem Statement

A fundamental requirement for extending the RADIUS attribute space is the maintenance of backwards compatibility. This means that RADIUS servers and proxies must be able to continue to decode and encode messages even though they may not need to understand an attribute that has been extended. More specifically, the scheme MUST be compliant with the various RADIUS RFCs such as [RFC2865] and RADIUS Accounting [RFC2866], etc.

The scheme SHOULD ensure that the size of the standard type space extension is large enough that it will not be quickly exhausted or is extensible in the event that it is.

Furthermore, the scheme SHOULD align with the Diameter NASReq Application [RFC4005], thereby allowing the two AAA standards to interoperate.

A need to group related RADIUS attributes together has become prevalent in current work. Therefore, the proposed scheme SHOULD provide a mechanism to group related attributes together.

In recent years, attribute sizes have been pushing the current limit of 253 octets. Fragmentation of RADIUS attributes has always been possible by extending the value into another attribute of the same type; however, this approach does not always work (for example, if more than one instance of an attribute occurs in the same RADIUS packet). The proposed scheme SHOULD enable the transmission of attributes longer than 253 octets.

4. RADIUS Type Extension

The solution described in this document takes the recommended VSA format [RFC2865] as a basis for the RADIUS Extended Attributes.

We allocate RADIUS the Vendor-Id of zero (0). In essence we are assigning the IETF a Vendor-Id which is what other SDOs have done in registering their own Vendor-Id.

Extended Attributes consist of an attribute header similar to that recommended by RFC 2865 [RFC2865] for Vendor Specific Attributes followed by a non-empty sequence of Type-Length-Value (TLV) triples (see below). If an Extended Attribute contains more than one TLV then all of the encapsulated TLVs MUST fit completely within the Extended Attribute.

The Extended Attribute header is 7 octets in length and is encoded as follows:

- The first octet contains the Type which is always Vendor-Specific (26)
- The second octet contains the length (in octets) of the entire Extended Attribute, including the Extended Attribute header and all encapsulated TLVs
- The next 4 octets contain the Vendor-Id (0)
- The final octet of the header contains the More flag and Tag field. If the one-bit More flag is set (1) this indicates that the encapsulated TLV is continued in the following Extended Attribute; if the More flag is clear (0) then all of the encapsulated TLVs fit into the current Extended Attribute. The More flag MUST NOT be set if the Extended Attribute contains more than one TLV. The Tag field is used to combine sets of related Extended Attributes into simple, one level groups.
- The Data field is an abstract container for TLVs; the Data field MUST contain at least one TLV.

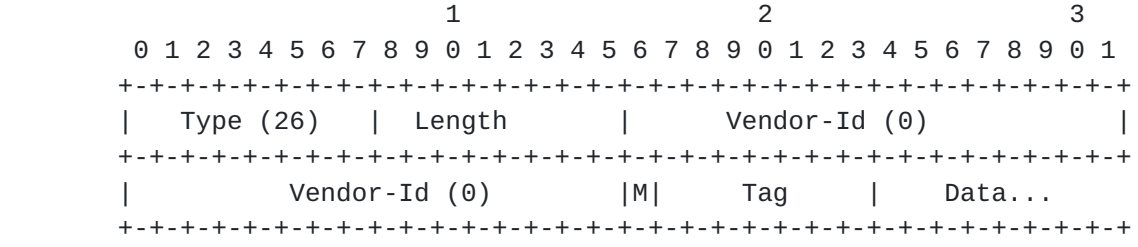
TLVs are encoded as follows:

- The first two octets contain the Ext-Type field
- The next octet is the Ext-Len field, representing the length in octets of the entire TLV, including the length of the Ext-Type field (2 octets), the length of the Ext-Len field itself (1 octet) and the length of the Value field (1 or more octets)
- The Value field consists of one or more octets comprising the actual data to be transmitted

5. Formal Syntax

This section describes the encoding scheme used for RADIUS Extended Attributes. The basis of this encoding is the format recommended for

Vendor Specific Attributes in RFC 2865 [RFC2865].



Type

26 for Vendor-Specific

Length

> 10

Vendor ID

The Vendor Id field is 4 octets in length and MUST be zero (0x0000), signifying an extended IETF RADIUS attribute

M (More)

The More Flag is one (1) bit in length and MUST be present. When a value to be transmitted exceeds 245 octets in length it is fragmented over two or more Extended Attributes. If the More Flag is set (1), this indicates that the Value field of the Extended Attribute contains a fragment of a larger value, which MUST be continued in the next Extended Attribute of the same Ext-Type. When the More Flag is clear (0), the final (or only) fragment of the value is contained in the Extended Attribute. The More Flag MUST NOT be set if the Length is less than 255. Any Extended Attributes containing multiple fragments of the same value MUST be in order and MUST be consecutive attributes in the packet.

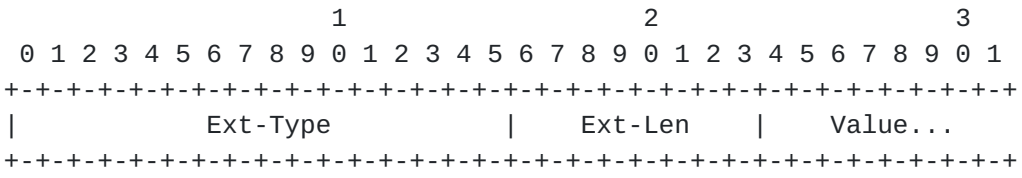
Tag

The Tag field is 7 bits long and MUST be present. It is used to group Extended Attributes. Extended Attributes with the same non-zero value in the Tag field belong to the same group. A Tag value of zero (0) indicates that the attribute is not grouped. A Tag value of all ones (0x7F) is reserved.

Data

The Data field is >= 4 octets in length. It consists of 1 or more TLVs.

TLVs have the following syntax:



Ext-Type

Two (2) octets. Up-to-date values of the Ext-Type field are specified in the most recent "Assigned Numbers" [IANA]. Values 64512-65535 (0xFC00-0xFFFF) are reserved.

Ext-Len

> 3. The length of the Type-Length-Value tuple in octets, including the Ext-Type, Ext-Len and Value fields.

Value

One or more octets.

6. Examples

Consider an attribute called Foo of type String. Foo has been allocated an Extended-Type of 257 by IANA. The following figure illustrates the encoding of the string "Hello":

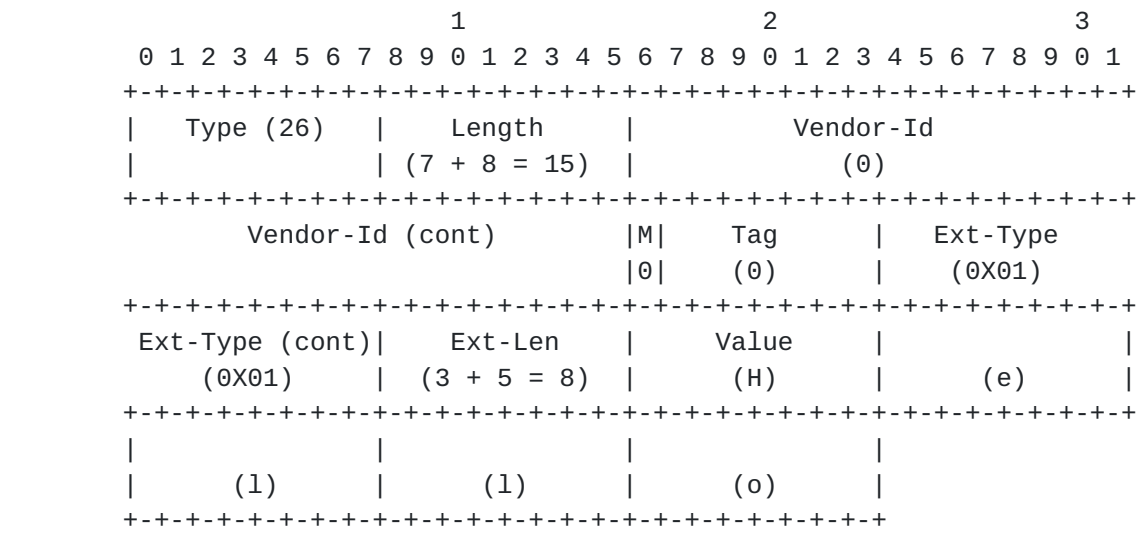


Figure 1

Now consider another instantiation of the Foo Extended Attribute, this one with a length of 251 octets. In this case the value is fragmented over two Extended Attributes. The first 245 octets are included in the first fragment which has the More bit set and the remaining 6 octets appear in the second attribute. Figure 2 below illustrates the encoding of the first 7 octets of the first Extended Attribute, while Figure 3 shows how the second attribute (containing the string "e end.") is encoded.

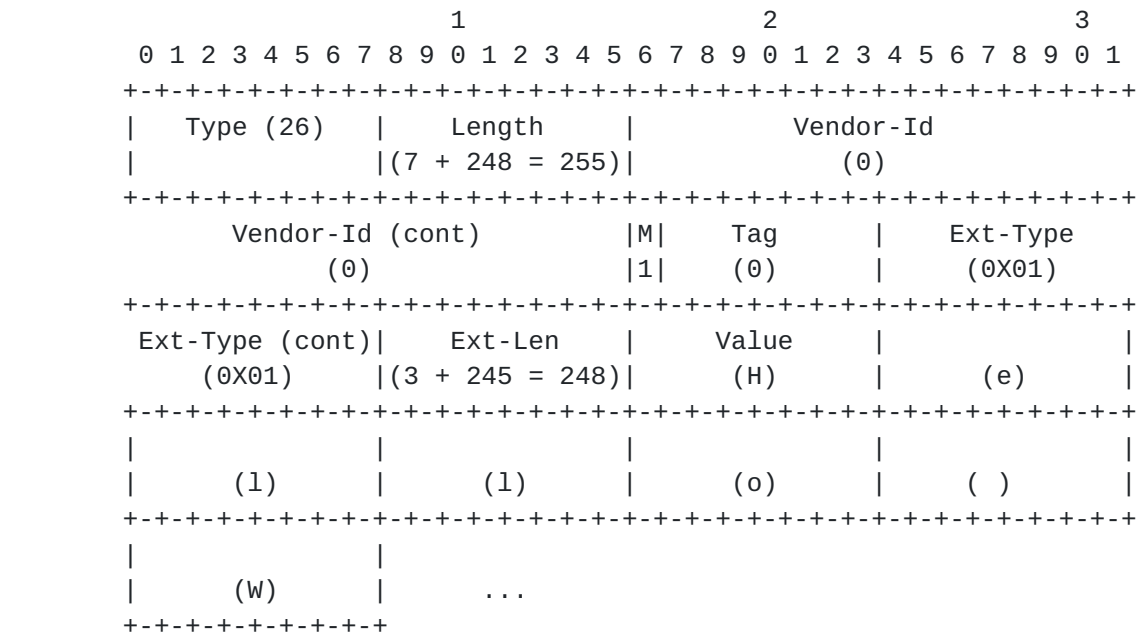


Figure 2

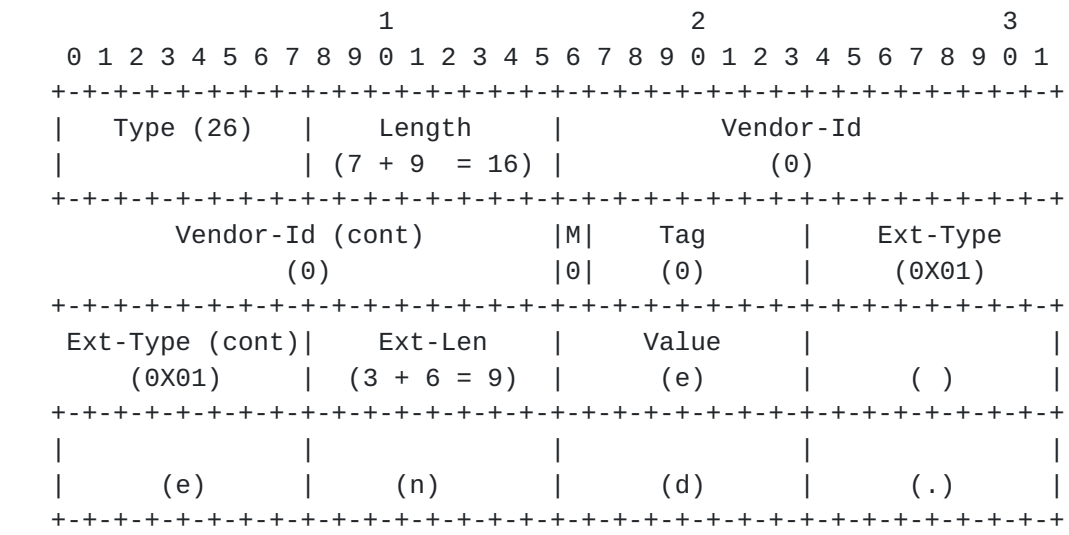


Figure 3

The next example illustrates several of the features of Extended Attributes:

o encapsulation of values greater than 253 octets in length

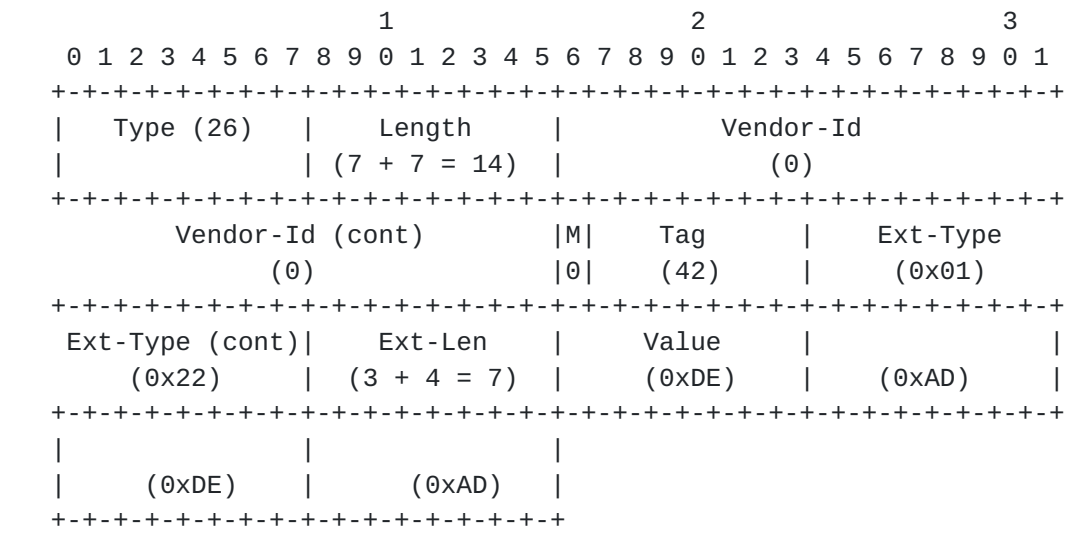
o grouping of related Extended Attributes using tags

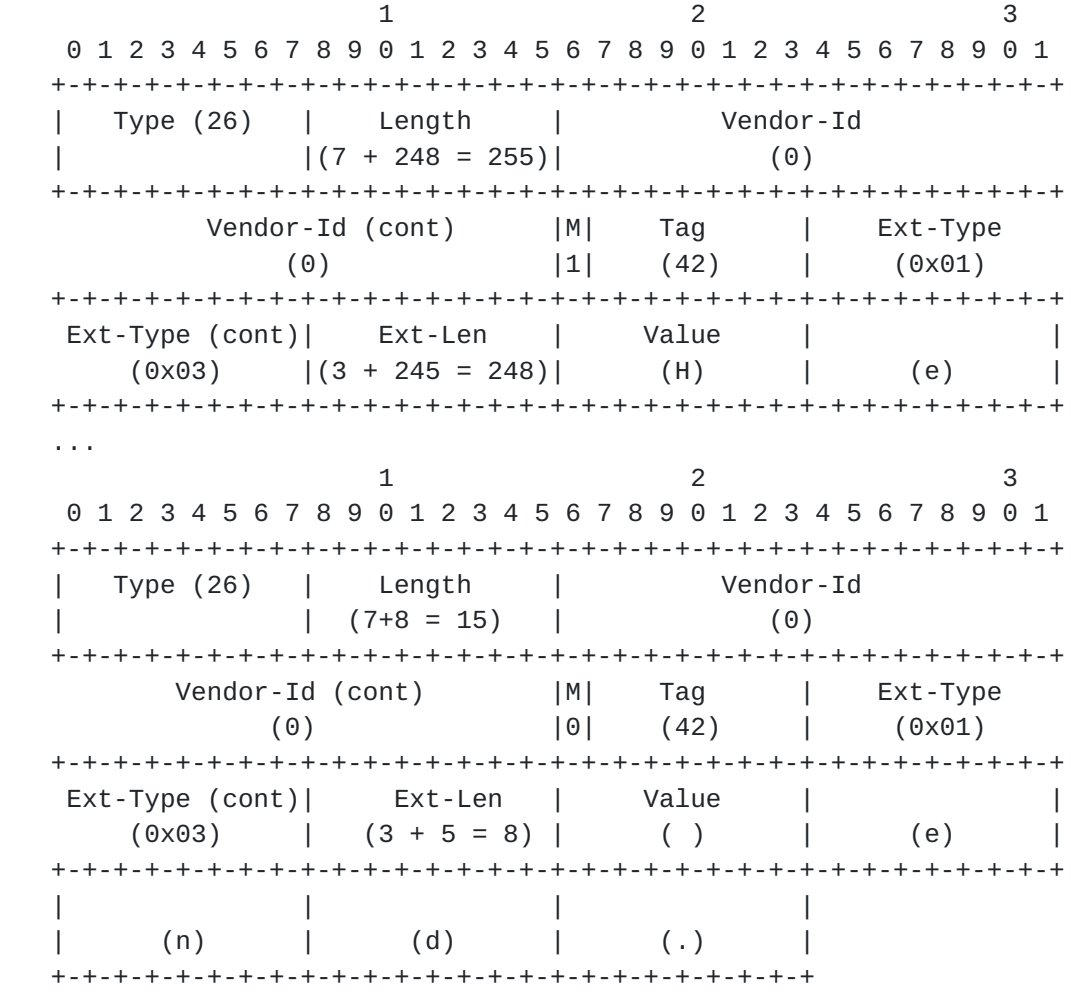
o encapsulation of more than one TLV in a single Extended Attribute

Consider the following structure:

```
struct
  Integer a;
  String  b;
  Integer c;
endStruct
```

Element 'a' is assigned an Extended Type of 290 (0x0122). Element 'b' is assigned an Extended Type of 259 (0x0103) and element 'c' is assigned an Extended Type of 271 (0x010F). The following figure illustrates the encoding where the value of 'a' contains 0xDEADDEAD, the first two octets of 'b' contain the string "He", octets 243-250 of 'b' contain "The end." and the value of 'c' is 0x12345678. The attributes are grouped together with TAG=42. Note that this encoding is only one out of several possibilities since there is no strict order in attribute marshalling; for the sake of brevity, octets 3-241 of the value of 'b' are omitted from the diagram.


1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Type (26)
Length
(7 + 7 = 14)
Vendor-Id
(0)
Vendor-Id (cont)
(0)
M
0
Tag
(42)
Ext-Type
(0x01)
Ext-Type (cont)
(0x22)
Ext-Len
(3 + 4 = 7)
Value
(0xDE)
(0xAD)
(0xDE)
(0xAD)


1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Type (26)
Length
(7 + 248 = 255)
Vendor-Id
(0)
Vendor-Id (cont)
(0)
M
1
Tag
(42)
Ext-Type
(0x01)
Ext-Type (cont)
(0x03)
Ext-Len
(3 + 245 = 248)
Value
(H)
(e)
...
1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Type (26)
Length
(7+8 = 15)
Vendor-Id
(0)
Vendor-Id (cont)
(0)
M
0
Tag
(42)
Ext-Type
(0x01)
Ext-Type (cont)
(0x03)
Ext-Len
(3 + 5 = 8)
Value
( )
(e)
(n)
(d)
(.)

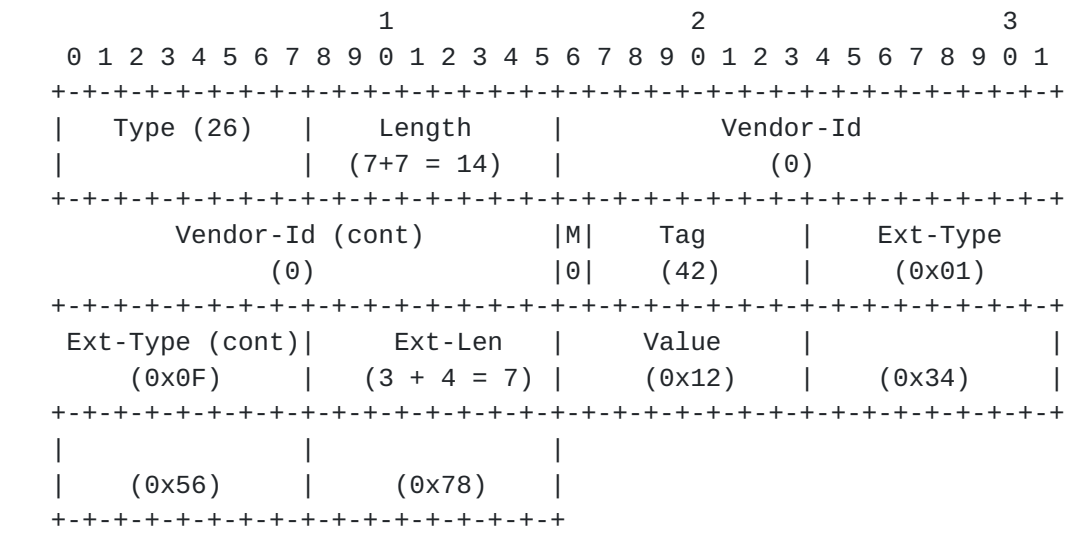


Figure 4

7. Diameter Considerations

Since the Extended Attributes are encoded as Vendor-Specific RADIUS Attributes (see [IANA]), no special handling is required by Diameter [RFC3588] entities; see [RFC4005] for details on the Diameter treatment of RADIUS VSAs.

8. Security Considerations

This document does not introduce any new security issues into the RADIUS protocol; for known security problems with RADIUS, see [RFC2865], [RFC2869] and [RFC2607].

9. IANA Considerations

This standard requires that the Vendor-Id of zero be allocated to the IETF.

It also requires that IANA set up a new registry for the RADIUS Extended Types, reserving the value ranges 0-255 (0x0000-0x00FF) and 64512-65535 (0xFC00-0xFFFF) for future purposes. Values in this registry should be allocated using the "IETF Review" policy [RFC5226].

10. References

10.1. Normative References

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

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

10.2. Informative References

[IANA] Internet Assigned Number Authority, "RADIUS TYPES", August 2008, <http://www.iana.org/assignments/radius-types>.

[RFC2607] Aboba, B. and J. Vollbrecht, "Proxy Chaining and Policy Implementation in Roaming", RFC 2607, June 1999.

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

[RFC2869] Rigney, C., Willats, W., and P. Calhoun, "RADIUS Extensions", RFC 2869, June 2000.

[RFC3575] Aboba, B., "IANA Considerations for RADIUS (Remote Authentication Dial In User Service)", RFC 3575, July 2003.

[RFC3588] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and J. Arkko, "Diameter Base Protocol", RFC 3588, September 2003.

[RFC4005] Calhoun, P., Zorn, G., Spence, D., and D. Mitton, "Diameter Network Access Server Application", RFC 4005, August 2005.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

Authors' Addresses

Yong Li
Bridgewater Systems Corporation
303 Terry Fox Drive
Suite 100
Ottawa, Ontario K2K 3J1
Canada

Phone: +1 (613) 591-6655
Email: yongli@bridgewatersystems.com
URI: http://www.bridgewatersystems.com/

Avi Lior
Bridgewater Systems Corporation
303 Terry Fox Drive
Suite 100
Ottawa, Ontario K2K 3J1
Canada

Phone: +1 (613) 591-6655
Email: avi@bridgewatersystems.com
URI: http://www.bridgewatersystems.com/

Glen Zorn
Network Zen
1310 East Thomas Street
Seattle, Washington 98102
US

Phone: +1 (206) 377-9035
Email: gwz@net-zen.net