Transmitting Confidential Data in RADIUS
draft-zorn-radius-encattr-10.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of 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 29, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2008).

Abstract

This document defines a set of RADIUS Attributes designed to allow the secure transmission of sensitive or confidential data between RADIUS clients and servers and the strong authentication of any RADIUS message.
Table of Contents

1. Introduction .................................................. 3
2. Specification of Requirements .............................. 3
3. Attributes .................................................... 4
   3.1. Crypto-Params ........................................... 4
   3.2. Encrypted-Attribute ................................. 6
   3.3. Message-Authentication-Code ........................ 7
   3.4. MAC-Randomizer ........................................ 11
4. IANA Considerations .......................................... 12
   4.1. Attribute Types ....................................... 12
   4.2. Attribute Values ...................................... 12
5. Security Considerations ..................................... 12
6. Acknowledgements ............................................ 13
7. References ................................................... 13
   7.1. Normative References ................................ 13
   7.2. Informative References .............................. 14
Authors’ Addresses ............................................. 14
Intellectual Property and Copyright Statements .............. 16
1. Introduction

Certain applications of the RADIUS protocol [RFC2865] require that the content (at least, if not the type and length) of one or more Attributes in a message be encrypted. For example, an application enabling the interception of certain packets by law enforcement agencies might require that it be impossible for an observer to distinguish between sessions which are under surveillance and those that are not. If packet interception is enabled and disabled using RADIUS (via the Access-Accept [RFC2865] or CoA-Request [RFC5176] messages, for example) then the Attributes used to signal this must be encrypted; however, it might be acceptable for the remainder of the Attributes to be sent in cleartext.

Currently, this type of transfer is usually accomplished using either the Tunnel-Password Attribute [RFC2868] or vendor-specific RADIUS attributes. However, there are several issues with these techniques:

- The Tunnel-Password Attribute was not designed to carry entire RADIUS Attributes and it is not large enough to hold an Attribute of the maximum size
- The security properties and strength of the encryption method used to hide the contents of the Tunnel-Password Attribute are unknown
- Due to its dependency upon the random Request Authenticator in the Access-Request message [RFC2865], the Tunnel-Password Attribute cannot be used in messages other than Access-Accept
- Although vendor-specific Attributes may not share the problems outlined above, a profusion of different attributes used for the same purpose entails considerable multiplication of effort and makes interoperability difficult to achieve

This document defines RADIUS Attributes that can be used to encapsulate and confidentially transfer one or more RADIUS Attributes using non-proprietary techniques with well-understood security properties. In addition, the Message-Authentication-Code Attribute may be used to provide strong authentication for any RADIUS message, including those used for accounting and dynamic authorization.

Discussion of this draft may be directed to the authors.

2. Specification of Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
3. Attributes

The following subsections describe the Attributes defined by this document. This specification concerns the following values:

<TBD1> Crypto-Params
<TBD2> Encrypted-Attribute
<TBD3> Message-Authentication-Code
<TBD4> MAC-Randomizer

3.1. Crypto-Params

Description

This Attribute is used to carry data used during the encryption and decryption of the Attribute(s) encapsulated in the Encrypted-Attribute Attribute, specifically the initialization vector and algorithm identifier.

Any packet that contains an Crypto-Params Attribute MUST also contain a Message-Authentication-Code Attribute (Section 3.3) and SHOULD contain one or more instances of the Encrypted-Attribute Attribute (Section 3.2).

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

```
0                   1                   2                   3
+-------------------------------+-------------------------------+-------------------------------+-------------------------------+
|     Type    |    Length     |    Enc Type   |     Key ID
|            |              |              |     Key ID (cont’d)
|            |              |              |     Key ID (cont’d)
|            |              |              |     Key ID (cont’d)
+-------------------------------+-------------------------------+-------------------------------+-------------------------------+
                  |      IV...                |      IV...                |      IV...
```
Encrypted Attributes

Type

<TBD1> for Crypto-Params

Length

>=19

Enc Type

The Enc Type field is 1 octet in length and serves to identify the encryption algorithm in use. This document defines the following decimal values for this field:

0 NULL

1 AES-CBC-128 [FIPS-197-2001]

2 AES-CBC-192 [FIPS-197-2001]

3 AES-CBC-256 [FIPS-197-2001]

Implementations MUST support Enc Types 0 and 1 (NULL and AES-CBC-128). Other values are to be assigned by IANA.

Key ID

The Key ID field is 16 octets in length and contains an identifier for the key used to encrypt and decrypt the String field of the Encrypted-Attribute Attribute (Section 3.2). Further specification of the content of this field is outside the scope of this document.

IV

The IV field is variable length and contains the initialization vector. The length of the IV field depends upon the algorithm specified in the Enc Type field (above); for the algorithms defined in this document, the length of the IV field is 16 octets. If no initialization vector is required by the algorithm specified by the Enc Type field, this field MAY be omitted.
3.2. Encrypted-Attribute

Description

This Attribute MAY be used to carry one or more encrypted Attributes in a RADIUS message.

Any packet that contains an Encrypted-Attribute Attribute MUST also include both a Crypto-Params Attribute (Section 3.1) and a Message-Authentication-Code (Section 3.3) Attribute.

The encryption of the Attribute(s) MUST be performed by the sender according to the following algorithm:

Concatenate the Attributes to be encrypted. If the algorithm specified by the Enc Type field of the Crypto-Params Attribute is a block cipher and the length in octets of the result is not an even multiple of the algorithm’s block size, pad the result of the concatenation on the right with enough zero-value octets to make the resulting string an even multiple of the block size in length. Encrypt the result using the algorithm specified by the Enc Type field and the initialization vector contained in the IV field of the Crypto-Params Attribute (Section 3.1). Split the resulting ciphertext into one or more chunks, each <= 253 octets in length. Encapsulate each chunk in a separate instance of the Encrypted-Attribute Attribute.

The receiver MUST recover the plaintext Attribute(s) using the following algorithm:

Concatenate the String fields of the received Encrypted-Attribute Attributes in order of reception. Decrypt the result using the algorithm specified in the Enc Type field and the initialization vector contained in the IV field of the Crypto-Params Attribute (Section 3.1). Split the resulting cleartext in to Attributes, discarding the padding (if any).

A subset of the attributes in a message can be authenticated and integrity-protected by setting the Enc Type field in the Crypto-Params Attribute (Section 3.1) to NULL and concatenating an instance of the Message-Authentication-Code Attribute to the set of attributes to be protected. When used in this way, only the attributes to be protected are to be used as input in the creation of the Message-Authentication-Code Attribute.

The Encrypted-Attribute Attribute MUST NOT be used to transfer keys between RADIUS servers and clients.
A summary of the Encrypted-Attribute attribute format is shown below. The fields are transmitted from left to right.

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

**Type**

<TBD2> for Encrypted-Attribute

**Length**

>=3

**String**

The String field is variable length and contains the actual encrypted Attributes (see above).

### 3.3. Message-Authentication-Code

**Description**

This Attribute MAY be used to "sign" messages to prevent spoofing. If it is present in a request, the receiver should take this a hint that the sender prefers the use of this Attribute for message authentication; the receiver is not obligated to do so, however.

The Message-Authentication-Code Attribute MUST be included in any message that contains a Key attribute.

Any packet that contains an instance of the Message-Authentication-Code Attribute SHOULD NOT contain an instance of the Message-Authenticator Attribute [RFC3579]. If both attributes are to be included in a message (e.g., for backward compatibility in a network containing both old and new clients), the value of the Message-Authentication-Code Attribute MUST be computed first.

If any message is received containing an instance of the Message-Authentication-Code Attribute, the receiver MUST calculate the correct value of the Message-Authentication-Code and silently discard the packet if the computed value does not match the value
received.

If a received message contains an instance of the MAC-Randomizer Attribute (Section 3.4), the received MAC-Randomizer Attribute SHOULD be included in the computation of the Message-Authentication-Code Attribute sent in the response, as described below.

A summary of the Message-Authentication-Code attribute format is shown below. The fields are transmitted from left to right.

```
<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
<th>Reserved</th>
<th>MAC Type</th>
</tr>
</thead>
</table>
```

Type

<TBD3> for Message-Authentication-Code

Length

>20

Reserved

This field is reserved for future usage and MUST be zero-filled.

MAC Type

The MAC Type field specifies the algorithm used to create the value in the MAC field. This document defines six values for the MAC Type field:

Implementations MUST support MAC Type 0 (HMAC-SHA-1); other values are to be assigned by IANA.

MAC Key ID

The MAC Key ID field is 16 octets in length and contains an identifier for the key. The MAC Key ID MUST refer to a key of a type and length appropriate for use with the algorithm specified by the MAC Type field (see above). Further specification of the content of this field is outside the scope of this document.

MAC

Both the length and value of the MAC field depend upon the algorithm specified by the value of the MAC Type field. If the algorithm specified is HMAC-SHA-1, HMAC-SHA-256 or HMAC-SHA-512, the MAC field MUST be 20, 32 or 64 octets in length, respectively. If the algorithm specified is CMAC-AES-128, CMAC-AES-192 or CMAC-AES-256, the MAC field SHOULD be 64 octets in length. The derivation of the MAC field value for all the algorithms specified in this document is identical, except for the algorithm used. There are differences, however, depending upon whether the MAC is being computed for a request message or a response. These differences are detailed below, with the free variable HASH-ALG representing the actual algorithm used.

Request Messages

For requests (e.g., CoA-Request [RFC5176], Accounting-Request [RFC2866], etc.), the value of the MAC field is a hash of the entire packet except the Request Authenticator in the header of the RADIUS packet, using a shared secret as the key, as follows.
MAC = MAC-ALG(Key, Type + Identifier + Length + Attributes)
where ‘+’ represents concatenation

The MAC-Randomizer Attribute (Section 3.4) MUST be included in any request in which the Message-Authentication-Code Attribute is used. The Random field of the MAC-Randomizer Attribute MUST be filled in before the value of the MAC field is computed.

If the Message-Authenticator-Code Attribute is included in a client request, the server SHOULD ignore the contents of the Request Authenticator.

Implementation Notes

When the hash is calculated, both the MAC field of the Message-Authenticator-Code attribute and the String field of the Message-Authenticator Attribute (if any) MUST be considered to be zero-filled.

Implementations SHOULD provide a means to provision a key (cryptographically separate from the normal RADIUS shared secret) to be used exclusively in the generation of the Message-Authentication-Code.

Response Messages

For responses (e.g., CoA-ACK [RFC5176], Accounting-Response [RFC2866], etc.), the value of the MAC field is a hash of the entire packet except the Response Authenticator in the header of the RADIUS packet using a shared secret as the key, as follows.

MAC = HASH-ALG(Key, Type + Identifier + Length + Attributes)
where ‘+’ represents concatenation

If the request contained an instance of the MAC-Randomizer Attribute and the responder wishes to include an instance of the Message-Authentication-Code Attribute in the corresponding response, then the MAC-Randomizer Attribute from the request MUST be included in the response.

If the Message-Authenticator-Code Attribute is included in a server response, the client SHOULD ignore the contents of the Response Authenticator.
Implementation Notes

When the hash is calculated, both the MAC field of the Message-Authenticator-Code attribute and the String field of the Message-Authenticator Attribute (if any) MUST be considered to be zero-filled.

The Message-Authentication-Code Attribute MUST be created and inserted in the packet before the Response Authenticator is calculated.

Implementations SHOULD provide a means to provision a key (cryptographically separate from the normal RADIUS shared secret) to be used exclusively in the generation of the Message-Authentication-Code.

3.4. MAC-Randomizer

Description

The MAC-Randomizer Attribute MUST be present in any message that includes an instance of the Message-Authentication-Code Attribute. The Random field MUST contain a 32 octet random number which SHOULD satisfy the requirements of [RFC4086].

Implementation Note

The Random field MUST be filled in before the MAC is computed. The MAC-Randomizer Attribute SHOULD be placed at the beginning of the RADIUS message if possible.

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

```
+---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |           Random...
+---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type

>TBD4< for MAC-Randomizer
4. IANA Considerations

This section explains the criteria to be used by the IANA for assignment of numbers within namespaces defined within this document. The "Specification Required" policy is used here with the meaning defined in BCP 26 [RFC2434].

4.1. Attribute Types

Upon publication of this document as an RFC, IANA must assign numbers to the Crypto-Params, Encrypted-Attribute, MAC-Randomizer and Message-Authentication-Code Attributes.

4.2. Attribute Values

As defined in Section 3.1, numbers may need to be assigned for future values of the Enc Type field of the Crypto-Params attribute. These numbers may be assigned by applying the "Specification Required" policy.

As defined in Section 3.3, numbers may need to be assigned for future values of the MAC Type field of the Message-Authentication-Code attribute. These numbers may be assigned by applying the "Specification Required" policy.

5. Security Considerations

Although the encryption algorithms specified in this document are believed to be strong, ultimately the confidentiality of the encrypted attributes depends upon the strength of the keys used to encrypt them. For this reason, implementations SHOULD use keys with entropy equal to or greater than the strength of the algorithm used (e.g., 128 bits of entropy for AES-CBC-128, etc.).

Given that the secret shared between RADIUS clients and servers typically has relatively weak entropy, it is NOT RECOMMENDED that implementations use the shared secret (or a derivative thereof) as a
key for attribute encryption.

To avoid the possibility of collisions, the same MAC key SHOULD NOT be used with more than $2^{n/2}$ messages, where ‘n’ is the length of the MAC value in octets.

6. Acknowledgements

Thanks to David McGrew for review and useful feedback.

7. References

7.1. Normative References


7.2. Informative References


Authors’ Addresses

Glen Zorn
Aruba Networks
1322 Crossman Avenue
Sunnyvale, CA  94089-1113
USA

Email: gwz@arubanetworks.com

Hao Zhou
Cisco Systems
4125 Highlander Parkway
REQ01/3/
Richfield, OH  44286
US

Phone: +1 (330) 523-2132
Email: hzhou@cisco.com
Joseph Salowey
Cisco Systems
2901 Third Avenue
SEA1/6/
Seattle, WA 98121
US

Phone: +1 (206) 256-380
Email: jsalowey@cisco.com