RADIUS Attributes for Key Delivery
draft-zorn-radius-keywrap-06.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 January 7, 2006.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

This document defines a set of RADIUS Attributes designed to allow both the secure transmission of encryption keys and strong authentication of any RADIUS message.
1. Introduction

Many remote access deployments (for example, deployments utilizing wireless LAN technology) require the secure transmission of session keys from an authentication server to a network access point. Currently, this transfer is most often accomplished using vendor-specific RADIUS attributes [RFC2548], with the integrity of the message protected by the RADIUS Response Authenticator [RFC2865], the Request and Response Authenticators (in the cases of RADIUS Accounting [RFC2866] and Dynamic Authorization [RFC3576]) or the Message-Authenticator Attribute [RFC3579]. However, there are several issues with these techniques:

- The key transport attributes were designed for use with a specific, proprietary protocol [RFC3078] and may be inappropriate for other uses
- The security properties and strength of the encryption method used to hide the keys are unknown
- The hash function ([RFC1321]) used in the construction of the Response Authenticator is proprietary and the construct itself is weaker than more modern methods (e.g., HMAC [RFC2104])
- The Message-Authenticator Attribute is unusable in some situations where strong message authentication might be required

This document defines a set of RADIUS Attributes that can be used to securely transfer encryption keys 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 document are to be interpreted as described in [RFC2119].

3. Attributes

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

Description

This Attribute MAY be used to carry an encryption key from a RADIUS server to a client.

It MAY be sent in request messages (e.g., Access-Request, etc.), as well; if the Key Attribute is present in a request, it SHOULD be taken as a hint by the server that the client prefers this method of key delivery over others, the server is not obligated to honor the hint, however. When the Key Attribute is included in a request message the Key ID, Lifetime, IV and Key fields MAY be omitted.

Any packet that contains a Key Attribute MUST also include a Message-Authentication-Code Attribute. If the client requires the use of the Key Attribute for key delivery and it is not present in the Access-Accept or Access-Challenge message, the client MAY ignore the message in question and end the user session.

The Key Attribute MUST NOT be used to transfer long-lived keys (i.e., passwords) between RADIUS servers and clients.

A summary of the Key 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>Enc Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>App ID</td>
<td>KEK ID</td>
<td>KEK ID (cont’d)</td>
<td>KEK ID (cont’d)</td>
</tr>
<tr>
<td>KEK ID</td>
<td>KEK ID</td>
<td>KEK ID (cont’d)</td>
<td>KEK ID (cont’d)</td>
</tr>
<tr>
<td>Key ID</td>
<td>Key ID</td>
<td>Key ID (cont’d)</td>
<td>Key ID (cont’d)</td>
</tr>
<tr>
<td>Key ID</td>
<td>Key ID</td>
<td>Key ID (cont’d)</td>
<td>Key ID (cont’d)</td>
</tr>
<tr>
<td>Lifetime</td>
<td>IV</td>
<td>IV (cont’d)</td>
<td>Key Data...</td>
</tr>
</tbody>
</table>

Type

[TBD1] for Key

Length

>= 3

Reserved

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

The Enc Type field indicates the method used to encrypt the key that is carried in the Key field. This document defines only one value (decimal) for this field:

0   AES Key Wrap with 128-bit KEK [RFC3394]

Other values are to be assigned by IANA.

Implementation Note

A shared secret is used as the key-encrypting-key (KEK) for the AES key wrap algorithm. Implementations SHOULD provide a means to provision a key (cryptographically separate from the normal RADIUS shared secret) to be used exclusively as a KEK.

App ID

The App ID field is 4 octets in length and identifies the type of application for which the key is to be used. This allows for multiple keys for different purposes to be present in the same message. This document defines one value for the App ID:

0   Unspecified

Other values are to be assigned by IANA; further specification of the content of this field is outside the scope of this document.

KEK ID

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

Key ID

The Key ID field is 16 octets in length and contains an identifier for the key. The Key ID MAY be used by communicating parties to identify the key that is being transmitted. The combination of App ID and Key ID MUST uniquely identify the key between the parties consuming the
key. The Key ID is assumed to be known to the parties that derived the key. Further specification of the content of this field is outside the scope of this document.

**Lifetime**

The Lifetime field is an integer [RFC2865] representing the period of time (in seconds) for which the keying material is valid.

Note: Applications using this value SHOULD consider the beginning of the key lifetime to be the point in time when the key is first used for either encryption or decryption.

**IV**

The length of the IV field depends upon the value of the Enc Type field, but is fixed for any given value thereof. When the value of the Enc Type field is 0 (decimal), the IV field MUST be 8 octets in length (as illustrated above) and the default value for the IV field is as specified in [RFC3394].

**Key Data**

The Key Data field is variable length and contains the actual encrypted keying material.

### 3.2 Random-Nonce

**Description**

The Random-Nonce Attribute SHOULD be used in conjunction with the Message-Authentication-Code Attribute (Section 3.3) to both provide randomness to the MAC and a connection between requests and responses; it MUST be present in any message that includes both a non-random Request or Response Authenticator (e.g., Accounting-Request [RFC2866], CoA-Request [RFC3576], etc.) and an instance of the Message-Authentication-Code Attribute. The Random field MUST contain a 32 octet random number which SHOULD satisfy the requirements of [RFC1750].

**Implementation Note**

The Random field MUST be filled in before the MAC is computed.

A summary of the Random-Nonce attribute format is shown below. The fields are transmitted from left to right.
Type

[TBD2] for Random-Nonce

Length

34

Random

This field MUST contain a 32 octet random number which SHOULD satisfy the requirements of [RFC1750].

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 as 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.

If any message is received containing a 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 Random-Nonce Attribute (Section 3.2), the received Random-Nonce Attribute SHOULD be included in the computation of the MAC field, as described below.

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

[TBD3] for Message-Authentication-Code

Length

>3

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 four values for the MAC Type field:

0   HMAC-SHA-1 [FIPS.180-2.2002] [RFC2104]
1   HMAC-SHA-256 [FIPS.180-2.2002] [RFC2104]
2   HMAC-SHA-512 [FIPS.180-2.2002] [RFC2104]

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. 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 [RFC3576], Accounting-Request [RFC2866], etc.), the value of the MAC field is a hash of the entire packet, including Type, ID, Length and Request Authenticator, using a shared secret as the key, as follows.

MAC = HASH-ALG(Type, Identifier, Length, Request Authenticator, Attributes)

The Random-Nonce Attribute (Section 3.2) SHOULD be included in any request in which the Message-Authentication-Code Attribute is used; it MUST be present in any message that includes both a non-random Request or Response Authenticator (e.g., Accounting-Request [RFC2866], CoA-Request [RFC3576], etc.) and an instance of the Message-Authentication-Code Attribute. If the Random-Nonce Attribute is included, it 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 MAY ignore the contents of the
Request Authenticator.

Implementation Notes

When the hash is calculated the MAC field 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 [RFC3576], Accounting-Response [RFC2866], etc.), the value of the MAC field is a hash of the entire packet, including Type, ID, Length and Request Authenticator, using a shared secret as the key, as follows.

\[ \text{MAC} = \text{HASH-ALG} (\text{Type, Identifier, Length, Request Authenticator, Attributes}) \]

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

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

Implementation Notes

When the hash is calculated the value of the MAC field 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.
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].

5. Attribute Types

Upon publication of this document as an RFC, IANA must assign numbers to the Key [TBD1], Random-Nonce [TBD2] and Message-Authentication-Code [TBD3] Attributes.

6. Attribute Values

As defined in Section 3.1, numbers may need to be assigned for future values of the Enc Type field of the Key attribute. These numbers may be assigned by applying the "Specification Required" policy. In particular, specifications MUST define the length of the IV field for the algorithm used.

As defined in Section 3.2, 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.

As defined in Section 3.2, numbers may need to be assigned for future values of the App Type field of the Message-Authentication-Code attribute. These numbers may be assigned by applying the "First Come First Served" policy.

7. Security Considerations

It is RECOMMENDED in this memo that two new keys be shared by the RADIUS client and server. If implemented, these two keys MUST be different from each other and SHOULD NOT be based on a password. These two keys SHOULD be cryptographically independent of the RADIUS shared secret used in calculating the Response Authenticator [RFC2865], Request Authenticator [RFC2866] [RFC3576] and Message-Authentication-Attribute [RFC3579]; otherwise if the shared secret is broken, all is lost. For the same reason, if the Message-Authentication-Code Attribute is included in a RADIUS or RADIUS Accounting packet, the Message-Authenticator Attribute [RFC3579] MUST NOT be included as well.

8. Contributors

Hao Zhou, Nancy Cam-Winget and John Fossaceca all contributed to this
9. Acknowledgements

Thanks (in no particular order) to Keith McCloghrie, Kaushik Narayan, Murtaza Chiba, Bill Burr, Russ Housley and Greg Weber for useful feedback.

10. References

10.1 Normative References


10.2 Informative References


Authors’ Addresses

Glen Zorn
Cisco Systems
2901 Third Avenue, Suite 600
SEA1/5/
Seattle, WA  98121
US
Phone: +1 (425) 344 8113
Email: gwz@cisco.com

Tiebing Zhang
3e Technologies International
700 King Farm Blvd.
Rockville, MD  20850
US
Phone: +1 (301) 944-1322
Email: tzhang@3eti.com