End-to-End Security in Roaming

1. Status of this Memo

This document is an Internet-Draft. 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."

To learn the current status of any Internet-Draft, please check the "1id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.ietf.org (US East Coast), nic.nordu.net (Europe), ftp.isi.edu (US West Coast), or munnari.oz.au (Pacific Rim).

The distribution of this memo is unlimited. It is filed as <draft-ietf-roamops-roamsec-01.txt>, and expires January 8, 1999. Please send comments to the authors.

2. Abstract

As noted in Roaming Requirements, there is a need for end-to-end security in roaming, including end-to-end integrity protection, and confidentiality. In roaming implementations based on proxy chaining, packets are routed between the NAS and home server through a series of proxies. Current roaming implementations provide only hop-by-hop security, guarding only against modification of packets in transit between hops. This makes it possible for untrusted proxies to modify packets sent between a NAS and a home server without detection, as well as to decrypt PAP passwords, Tunnel passwords, and other hidden attributes which are available to it in cleartext.

This document provides a framework for end-to-end security in roaming, making it possible to provide end-to-end message integrity and attribute hiding through addition of three new attributes.

3. Introduction

As noted in [2], there is a need for end-to-end security in roaming, including end-to-end integrity protection, and confidentiality. In
roaming implementations based on proxy chaining, packets are routed between the NAS and home server through a series of proxies. Current roaming implementations, as described in [1] provide only hop-by-hop security, guarding only against modification of packets in transit between hops. This makes it possible for untrusted proxies to modify packets sent between a NAS and a home server without detection, as well as to decrypt PAP passwords, Tunnel passwords, and other hidden attributes which are available to proxies in cleartext.

This document provides a framework for end-to-end security in roaming, making it possible to provide end-to-end message integrity and attribute hiding through addition of three new attributes, Security-Parameter-Index, Hidden and End-to-End-Signature. In this document, it is assumed that a key has previously been established between the two endpoints. It is this key which is used in calculation of the message integrity check, as well as in end-to-end encryption of attributes. Future documents will discuss key exchange issues.

The Security-Parameters-Index attribute is used to identify the security association within which the End-to-End-Signature and Hidden attributes are to be evaluated. Note that in the case where an intermediate proxy implements policy, it is possible for a security association to be established between the intermediate proxy and the home server, NAS, or local proxy. For example, an intermediate proxy may immediately send an Access-Reject to the NAS in response to an Access-Request, without having first forwarded it to the home server. In this case, the intermediate proxy and NAS would need to establish a security association in order to permit verification of the authenticity of the Access-Reject.

Using the Hidden attribute, it is possible for the client or server to protect an attribute end-to-end. This is accomplished by encapsulating and then encrypting another attribute within the Hidden attribute.

Using the End-to-End-Signature attribute, it is possible for a client or server to provide a keyed MAC which will allow end-to-end integrity protection. The keyed MAC is calculated over the immutable portions of the packet header, as well as all of the attributes preceding the End-to-End-Signature attribute. This makes it possible for some or all of the attributes in the packet to be protected, at the sender’s discretion. While an intermediate proxy may add, delete or edit unprotected attributes, it MUST NOT add, delete or modify protected attributes so that the validity of the keyed MAC can be maintained.

By allowing the message integrity check to be applied to a subset of attributes selected by the sender, and by allowing attributes to be hidden individually, these extensions enable end-to-end security functionality while at the same time enabling proxies to continue to implement policy. The policy function is a central part of the roaming architecture since roaming is inherently an inter-domain activity.
4. Requirements language

In this document, the key words "MAY", "MUST", "MUST NOT", "optional", "recommended", "SHOULD", and "SHOULD NOT", are to be interpreted as described in [7].

5. Transition

Since NAS devices will not initially implement the End-to-End-Signature and Hidden attributes, it is envisaged that these attributes will initially be deployed on local proxies and home servers. In this scenario, the local proxy will be configured to employ end-to-end security for those NAS devices that do not support this. In this case, the local proxy would add End-to-End security attributes to Access-Requests destined for the home server and would process End-to-End security attributes in an Access-Accept, Access-Reject or Access-Challenge originated by the home server.

Note that if a NAS is end-to-end security enabled, then the local proxy will receive an Access-Request from the NAS with an End-to-End-Signature and will not need to add its own. As a result, a packet should include at most one End-to-End-Signature attribute. A packet may contain more than one Hidden attribute.

6. Proposed attributes

6.1. Security-Parameter-Index

Description

The purpose of the Security-Parameter-Index attribute is to identify the security association within which the End-to-End-Signature and Hidden attributes should be evaluated.

A summary of the Security-Parameter-Index attribute 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 2
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |    Length     |            Value          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Value (cont’d)          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type

? for Security-Parameter-Index

Length
Value

The Value field is four octets. This serves as an index identifying the security association established between the two end-points.

6.2. End-to-End-Signature

Description

This attribute provides for the use of a keyed-MAC to be verified end-to-end.

A summary of the End-to-End Signature attribute is shown below. The fields are transmitted from left to right.

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| +-----------------------------------------------+
| | Type | Length | Protocol | String |
| +-----------------------------------------------+
| | String... |
| +-----------------------------------------------+

Type

? for End-to-End-Signature

Length

19 (protocol 1)

Protocol

The protocol field specifies the authentication algorithm to be used in computing the message authentication code (MAC) which is placed in the string field.

If the protocol field is 1, the HMAC protocol, described in [6] is used, along with the MD5 algorithm described in [5]. All implementations MUST support HMAC-MD5. No other protocol values are defined.

String

The End-to-End-Signature is an calculated using the algorithm specified in the protocol field. The MAC is computed over the portion of the RADIUS packet from the beginning until the end of the End-to-End-Signature attribute, including Type, ID, Length and authenticator. In calculating the End-to-End-Signature, the authenticator should be considered to be filled with zeroes, and the End-to-End-Signature string should be considered to be
sixteen octets of zero.

The portion of the RADIUS packet after the End-to-End-Signature attribute is not included in the calculation of the MAC. This makes it possible for a proxy to add, modify, or delete attributes placed after the End-to-End-Signature attribute. As a result, attributes placed prior to the End-to-End-Signature attribute can be considered "protected" and those placed after the attribute can be considered to be "unprotected." Note that in order for the End-to-End-Signature to be verified, the proxy MUST maintain attribute order.

6.3. Hidden

Description

The purpose of the Hidden attribute is to make possible end-to-end encryption of individual attributes. This would make it possible for attributes such as User-Password or Tunnel-Password to be sent securely between a RADIUS client and a server, without risk of compromise by an untrusted proxy.

A summary of the Hidden attribute 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 2 3 4 5 6 7 8 9 0 1 2
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |    Length     |        String
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type

? for Hidden

Length

>=4

String

The String field includes the encapsulated ciphertext of the attribute which is to be hidden end-to-end. The hidden attribute is encrypted using a key and encryption algorithm which had previously been established between the two endstations. Note that due to the 253 octet limitation on the size of RADIUS attributes, the encapsulated attribute may be a maximum of 251 octets in length.
7. Table of Attributes

The following table provides a guide to which attributes may be found in which kind of packets.

<table>
<thead>
<tr>
<th>Request</th>
<th>Accept</th>
<th>Reject</th>
<th>Challenge</th>
<th>#</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1</td>
<td>0-1</td>
<td>0-1</td>
<td>0-1</td>
<td>?</td>
<td>End-to-End-Signature</td>
</tr>
<tr>
<td>0+</td>
<td>0+</td>
<td>0</td>
<td>0+</td>
<td>?</td>
<td>Hidden</td>
</tr>
<tr>
<td>0-1</td>
<td>0-1</td>
<td>0-1</td>
<td>0-1</td>
<td>?</td>
<td>SPI</td>
</tr>
</tbody>
</table>

8. Acknowledgments

Thanks to Glen Zorn of Microsoft for useful discussions of this problem space.

9. References


10. Authors’ Addresses

Pat R. Calhoun
Technology Development
Sun Microsystems, Inc.
15 Network Circle
Menlo Park, CA 94025

Phone: 650-786-7733