RADIUS Extension for Digest Authentication
draft-sterman-aaa-sip-02.txt

Status of this Memo

By submitting this Internet-Draft, I certify that any applicable patent or other IPR claims of which I am aware have been disclosed, and any of which I become aware will be disclosed, in accordance with RFC 3668.

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 December 10, 2004.

Copyright Notice

Copyright (C) The Internet Society (2004). All Rights Reserved.

Abstract

Basic and Digest authentication schemes are widely used in protocols such as SIP and HTTP. RADIUS is a protocol for back end
authentication. RADIUS supports Basic authentication natively, as well as several other authentication schemes, such as CHAP, but does not support Digest authentication scheme. This document describes an extension to RADIUS for Digest authentication and provides a scenario of Digest user authentication.

Table of Contents

1. Introduction ........................................... 3
   1.1 Terminology ........................................ 3
   1.2 Motivation ......................................... 3
   1.3 Scenario 1, RADIUS client chooses nonces .......... 4
   1.4 Scenario 2, RADIUS server chooses nonces .......... 5
   1.5 Approach ........................................... 7
2. New RADIUS attributes .................................. 8
   2.1 Digest-Response attribute .......................... 8
   2.2 Digest-Realm attribute .............................. 9
   2.3 Digest-Nonce attribute .............................. 9
   2.4 Digest-Method attribute ............................. 10
   2.5 Digest-URI attribute ................................ 11
   2.6 Digest-QoP attribute ................................ 11
   2.7 Digest-Algorithm attribute ......................... 12
   2.8 Digest-Body attribute ................................ 12
   2.9 Digest-CN nonce attribute ......................... 13
   2.10 Digest-Nonce-Count attribute ....................... 13
   2.11 Digest-Username attribute ......................... 14
   2.12 Digest-Opaque attribute ........................... 15
   2.13 Digest-AKA-Auts attribute ......................... 15
   2.14 Digest-Domain attribute ............................ 16
   2.15 Digest-Domain attribute ............................ 16
   2.16 Digest-Stale attribute ............................. 17
3. Detailed Description .................................... 19
   3.1 RADIUS Client Behaviour ............................ 19
   3.2 RADIUS Server Behaviour ............................ 20
4. Migration Path to DIAMETER ............................... 22
5. IANA Considerations .................................... 24
6. Security Considerations ................................ 25
7. Example .................................................. 26
8. Changes from -01 ....................................... 30
9. References ............................................. 31
   9.1 Normative References ................................. 31
   9.2 Informative References ............................... 31
   Authors’ Addresses ..................................... 32
A. Acknowledgements ..................................... 33
   Intellectual Property and Copyright Statements ....... 34
1. Introduction

1.1 Terminology

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].

1.2 Motivation

Digest authentication is a simple authentication mechanism for HTTP and SIP. While it was not too successful in HTTP environments, it is the only SIP authentication mechanism that has been widely adopted. Due to the weaknesses of Digest authentication (see Section 6), PKI-based authentication and encryption mechanisms have been introduced into SIP: TLS [RFC2246] and S/MIME [RFC2633]. However, most SIP user agents that support TLS don’t send client certificates. SIP with S/MIME is lacking support, too: even two years after the inclusion of S/MIME into SIP, almost no implementations exist.

SIP service providers wishing to authenticate their clients have the following options: they can

- build a PKI and wait for interoperable S/MIME capable SIP implementations,
- build a PKI and wait for SIP implementations supporting TLS with client-side certificates,
- replace their existing RADIUS infrastructure with DIAMETER [RFC3588], when DIAMETER supports HTTP Digest authentication,
- use TLS for server authentication and plaintext passwords (Basic) for client authentication, which can be done with standard RADIUS,
- upgrade their existing RADIUS servers with the functionality described in this document.

PKI solutions only cover authentication, not authorization (EAP could change this, but its use with SIP is not standardized). TLS / Basic authentication works only with the limited number of SIP devices that implement TLS. Basic authentication has been deprecated for SIP in [RFC3261].

Current RADIUS-based AAA infrastructures have been built and debugged over years. Deficiencies of RADIUS have been mitigated with proprietary (ie costly) extensions. Operators are therefore reluctant to replace their RADIUS infrastructure in order to enable a single new authentication mechanism.

Given the complexity of S/MIME, simple clients will continue to support HTTP digest authentication only. Its interoperability with a back-end authentication protocol such as RADIUS is needed.
Operators that are about to replace their RADIUS-based AAA infrastructure are strongly recommended to use DIAMETER.

1.3 Scenario 1, RADIUS client chooses nonces

Figure 1 depicts the basic scenario that is relevant for this document. It shows a generic case where entities A and B communicate in the front-end using protocols such as HTTP/SIP, while entities B and C communicate in the back-end using RADIUS.

#### Figure 1: Overview of basic operation

The roles played by the entities in this scenario are as follows:

A: HTTP client / SIP UA

B: (HTTP server / HTTP proxy server / SIP proxy server / SIP UAS)
acting also as a RADIUS NAS

C: RADIUS server

The relevant order of messages sent in this scenario is as follows:
A sends B an HTTP/SIP request without authorization header (step 1). B challenges A sending an HTTP/SIP "(Proxy) Authorization required" response containing a locally generated nonce (step 2). A sends B an HTTP/SIP request with authorization header (step 3). B sends C a RADIUS Access-Request with attributes described in this document (step 4). C responds to B with a RADIUS Access-Accept/Access-Reject response (step 5). If credentials were accepted B receives an Access-Accept response and the message sent from A is considered authentic. If B receives an Access-Reject response, however, B then responds to A with a "(Proxy) Authorization required" response (step 6).

1.4 Scenario 2, RADIUS server chooses nonces

Figure 2 depicts an alternative scenario, where the RADIUS server generates nonces. It shows a generic case where entities A and B communicate in the front-end using protocols such as HTTP/SIP, while entities B and C communicate in the back-end using RADIUS.
The roles played by the entities in this scenario are as follows:

A: HTTP client / SIP UA

B: {HTTP server / HTTP proxy server / SIP proxy server / SIP UAS) acting also as a RADIUS NAS

C: RADIUS server

The relevant order of messages sent in this scenario is as follows:

A sends B an HTTP/SIP request without authorization header (step 1). B sends an Access-Request message with the newly defined Digest-Method and Digest-URI attributes but without a Digest-Nonce attribute to the RADIUS server, C (step 2). C chooses a nonce and responds with an Access-Accept (step 3). This Access-Accept contains Digest attributes, from which B takes values to construct a HTTP/SIP "(Proxy) Authorization required" response. The remaining steps are identical with scenario 1 (Section 1.3): B sends this response to A (step 4). A resends its request with its credentials (step 5). B sends an Access-Request to C (step 6). C checks the credentials and replies with Access-Accept or Access-Reject (step 7). Dependent on the C’s result, B processes A’s request or rejects it with a "(Proxy)
Authorization required" response (step 8).

1.5 Approach

The approach taken here is to extend RADIUS to support Digest authentication by mimicking its native support for CHAP authentication. According to [RFC2865], the RADIUS server distinguishes between different authentication schemes by looking at the presence of an attribute specific for that scheme. For the three natively supported authentication schemes, these attributes are: User-Password for PAP (or any other clear-text password scheme), CHAP-Password for CHAP, and State + User-Password for challenge-response scheme. This document adds another attribute to be used in this role: Digest-Response. Also according to [RFC2865], "An Access-Request packet MUST contain either a User-Password or a CHAP-Password or a State. It MUST NOT contain both a User-Password and a CHAP-Password. If future extensions allow other kinds of authentication information to be conveyed, the attribute for that can be used instead of User-Password or CHAP-Password." The Digest-Response introduced here therefore can be used instead of User-Password or CHAP-Password.

The HTTP Authentication parameters found in the Proxy-Authorization or Authorization request header are mapped into newly defined RADIUS attributes. These new RADIUS attributes are defined in the document together with some other information required for calculating the correct digest response on the RADIUS server with exception of the password, which the RADIUS server is assumed to be able to retrieve from a data store given the username.

In most cases, the operation outlined in Section 1.3 is sufficient. It reduces the load on the RADIUS server to a minimum. However, in some cases the RADIUS server is better off with pre-computed hashes. Section 1.4 describes a mechanism that enables this style of authentication.
2. New RADIUS attributes

DIG-RES, DIG-REALM, DIG-NONCE, DIG-METHOD, DIG-URI, DIG-QOP, DIG-ALG, DIG-BODY, DIG-CNONCE, DIG-NC, DIG-USER, DIG-OPAQUE, DIG-AUTHP, DIG-AUTS, DIG-DOMAIN and DIG-STALE are placeholders for values that will be assigned by IANA, if this specification becomes a working group document.

The term ‘HTTP-style’ denotes any protocol that uses HTTP-like headers and uses HTTP digest authentication as described in [RFC2617]. Examples are HTTP and SIP.

2.1 Digest-Response attribute

If this attribute is present, the RADIUS server SHOULD view the Access-Request as a Digest one. The following paragraphs apply for RADIUS servers implementing this specification. Access-Request packets MUST contain an Digest-Response attribute. In Access-Request packets, this attribute contains the digest taken from request-digest field in Digest (Proxy)Authorization header, as received from the HTTP or SIP client.

Access-Accept packets MUST contain a Digest-Response attribute. In Access-Accept packets, this attribute contains a digest that can be used for generating Authentication-Info headers. The calculation of this digest is described in [RFC2617], section 3.2.3. A summary of the Digest-Response attribute format is shown below. The fields are transmitted from left to right.

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

Type
DIG-RES for Digest-Response. Early implementations have used the experimental type 206.
Length
34
String
In Access-Requests, this string proves the user knows a password. The String field is 32 octets long and contains hexadecimal representation of 16 octet digest value as it was calculated by the authenticated client. The String field SHOULD be copied from request-digest of digest-response ([RFC2617]). In Access-Accepts, this string proves the RADIUS server knows the password. The RADIUS server calculates a digest according to section 3.2.3 of [RFC2617] and copies the result into this string.

2.2 Digest-Realm attribute

This string attribute describes a protection space of the RADIUS server. See [RFC2617] 1.2 for details.

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

Type
DIG-REALM
Length
>=3
String
In Access-Requests, the RADIUS client takes the value of the realm directive (realm-value) from the HTTP-style request it wants to authenticate. In Access-Accept messages, the RADIUS server puts the expected realm value into this attribute, if the RADIUS client asked for a nonce.

2.3 Digest-Nonce attribute

This attribute holds a random nonce to be used in the HTTP Digest calculation.
2.4 Digest-Method attribute

This attribute holds the method string to be used in the HTTP Digest calculation.

In Access-Requests, the RADIUS client takes the value of the request method from the HTTP-style request it wants to authenticate. This attribute MUST only be used in Access-Request messages.
2.5 Digest-URI attribute

This attribute holds the URI string to be used in the HTTP Digest calculation.

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
<th>String...</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIG-URI</td>
<td>&gt;=3</td>
<td></td>
</tr>
</tbody>
</table>

In Access-Requests, the RADIUS client takes the value of the request URI (digest-uri-value) from the HTTP-style request it wants to authenticate. The attribute MUST only be used in Access-Request messages.

2.6 Digest-QoP attribute

This attribute holds the Quality of Protection parameter that influences the HTTP Digest calculation.

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
<th>String...</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIG-QOP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In Access-Requests, the RADIUS client takes the value of the qop directive (qop-value) from the HTTP-style request it wants to authenticate. The attribute MUST only be used in Access-Request messages.

2.7 Digest-Algorithm attribute

This attribute holds the algorithm parameter that influences the HTTP Digest calculation.

2.8 Digest-Body attribute

When using the qop level 'auth-int', the contents of the message body are required for digest calculation. Instead of sending the complete body of the message, only its hash value is sent. This hash value can be used directly in the digest calculation.
Type DIG-BODY
Length 34
String String, hexadecimal representation of a digest calculated over entity-body of HTTP/SIP request ([RFC2616], [RFC3261]). Computed by entity B in figure Figure 1. This attribute is not part of the HTTP Digest response. See [RFC2617] section 3.2.2.3. This attribute MUST only be sent in Access-Request packets.

2.9 Digest-CNonce attribute

This attribute holds the client nonce parameter that is used in the HTTP Digest calculation.

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>+---------------------------+---------------------------+---------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type</td>
<td>Length</td>
</tr>
</tbody>
</table>

Type DIG-CNONCE
Length >=3
String In Access-Requests, the RADIUS client takes the value of the cnonce directive (cnonce-value) from the HTTP-style request it wants to authenticate. The attribute is never used in Access-Response messages.

2.10 Digest-Nonce-Count attribute

This attribute holds the nonce count parameter that is used to detect replay attacks.
Type DIG-NC
Length 9
String
In Access-Requests, the RADIUS client takes the value of the nc directive (nc-value) from the HTTP-style request it wants to authenticate. The attribute MUST only be used in Access-Request messages.

2.11 Digest-Username attribute

This attribute holds the user name parameter that is used in the HTTP digest calculation.

Type DIG-USER
Length >= 3
String
In Access-Requests, the RADIUS client takes the value of the username directive (username-value) from the HTTP-style request it wants to authenticate. The RADIUS server SHOULD NOT use this value for password finding, but only for digest calculation purpose. In order to find the user record containing the password, the RADIUS server SHOULD use the value of the (RFC2865) -User-Name attribute. This attribute MUST only be sent in Access-Request packets.
2.12 Digest-Opaque attribute

This attribute holds the opaque parameter that is passed to the HTTP-style client. The HTTP-style client passes this value back to the server (i.e., the RADIUS client) without modification.

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
<th>String...</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIG-OPAQUE</td>
<td>&gt;=3</td>
<td></td>
</tr>
</tbody>
</table>

This attribute is only used when the RADIUS server chooses nonces. In Access-Requests, the RADIUS client takes the value of the opaque directive from the HTTP-style request it wants to authenticate and puts it into this attribute. In Access-Accepts that convey a nonce, the RADIUS server MAY include this attribute.

2.13 Digest-Auth-Param attribute

This attribute is a placeholder for future extensions.
DIG-AUTHP
Length
>=3
String
This attribute is for future extensions. Any extension parameter in the digest-response can be put into a Digest-Auth-Param attribute. The string consists of the whole parameter, including its name and the equal (‘=’) sign. RADIUS servers that do not implement a parameter contained in an Digest-Auth-Param attribute MUST respond with an Access-Reject message. RADIUS clients that do not implement a parameter contained in an Digest-Auth-Param attribute MUST reject the original HTTP-style request.

2.14 Digest-AKA-Auts attribute

This attribute holds the auts parameter that is used in the AKA Digest ([RFC3310]) calculation.

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

Type
DIG-AUTS
Length
>=3
String
In Access-Requests, the RADIUS client takes the value of the auts directive from the HTTP-style request it wants to authenticate. It is only used if the algorithm of the digest-response denotes a version of AKA digest [RFC3310]. RADIUS servers that do not implement AKA digest MUST respond with an Access-Reject message. The attribute MUST only be used in Access-Request messages.

2.15 Digest-Domain attribute

When a RADIUS client has asked for a nonce, the RADIUS server MAY add one or more Digest-Domain attributes to its Access-Accept message. The RADIUS client puts them into the quoted, space-separated list of
URIs of the 'domain' directive of a WWW-Authenticate header. The URIs in the list define the protection space (see [RFC2617], section 3.2.1).

2.16 Digest-Stale attribute

If this attribute is present, the RADIUS server did not accept the nonce value.
The string consists of a single character ‘l’. The contents of this string MAY be ignored by an implementation. The presence of the attribute itself tells the RADIUS client that the nonce was considered stale. The attribute MUST only be used in Access-Accept messages.
3. Detailed Description

3.1 RADIUS Client Behaviour

A RADIUS client without an encrypted or otherwise secured connection to its RADIUS server only accepts unsecured connections from its HTTP-style clients (or else the clients would have a false sense of security).

The RADIUS client examines the (Proxy-)Authorization header of an incoming HTTP-style request message. If this header is present and contains HTTP digest information, the RADIUS client checks the ‘nonce’ parameter. If the ‘nonce’ value has not been sent by the RADIUS client, it responds with a 401 (Unauthorized) or 407 (Proxy Authentication Required) to the HTTP-style client. In this error response, the RADIUS client sends a new nonce.

If the RADIUS client recognizes the nonce, it takes the header parameters and puts them into a RADIUS Access-Request message. It puts the ‘response’ parameter into a Digest-Response attribute and the realm / nonce / qop / algorithm / cnonce / nc / username into the respective Digest-Realm / Digest-Nonce / Digest-QoP / Digest-Algorithm / Digest-CNnonce / Digest-Nonce-Count / Digest-Username attributes. The request URI and the request method are put into the Digest-URI and Digest-Method attributes. Now, the RADIUS client sends the Access-Request message to the RADIUS server.

The RADIUS client has three ways to obtain nonces: it generates them locally, it has received one in a Digest-Nonce attribute of a previously received Access-Accept message, or it asks the RADIUS server for one. To do the latter, it sends an Access-Request containing a Digest-Method and Digest-URI attribute but without a Digest-Nonce attribute. The RADIUS server chooses a nonce and responds with an Access-Accept containing a Digest-Nonce attribute. If the RADIUS server responds with an Access-Reject, the RADIUS client MAY generate a nonce locally. If the RADIUS client does not generate nonces locally, the authentication has failed.

The RADIUS server processes the message and responds with an Access-Accept or an Access-Reject. If the RADIUS client receives an Access-Accept, it examines the Digest attributes contained in the message.

If the Digest-Stale attribute is present, the RADIUS client sends an error (401 or 407) response containing WWW-/Proxy-Authenticate header with a stale directive.

If the Digest-Stale attribute is not present, the RADIUS client
constructs an Authentication-Info header and puts the contents of Digest-Response into the ‘rspauth’ directive. The RADIUS server MAY have added a Digest-Nonce attribute. If the RADIUS client discovers this, it puts the contents of this attribute into a ‘nextnonce’ directive. Now it can send a HTTP-style response.

If the RADIUS client did not receive a (Proxy-)Authorization header from its HTTP-style client, it obtains a new nonce and sends an error response (401 or 407) containing a (Proxy-)Authenticate header.

If the RADIUS client receives an Access-Reject or no response from the RADIUS server, it sends an error response to the HTTP-style request it has received.

3.2 RADIUS Server Behaviour

If the RADIUS server receives an Access-Request message with a Digest-Method and a Digest-URI attribute but without a Digest-Nonce attribute, it chooses a nonce. It puts the nonce into a Digest-Nonce attribute and sends it in an Access-Accept message to the RADIUS client. If the server cannot choose a nonce, it replies with an Access-Reject message.

If the RADIUS server receives an Access-Request message containing a Digest-Response attribute, it looks for the following attributes: Digest-Realm, Digest-Nonce, Digest-Method, Digest-URI, Digest-QoP, Digest-Algorithm, Digest-Username. Depending on the content of Digest-Algorithm and Digest-QoP, it looks for Digest-Body, Digest-CNonce and Digest-AKA-Auts, too. See [RFC2617] and [RFC3310] for details. If it has issued a Digest-Opaque attribute along with the present nonce, the Access-Request MUST have a matching Digest-Opaque attribute.

If it does not find these attributes, it responds with an Access-Reject message. If the attributes are present, the RADIUS server calculates the digest response as described in [RFC2617]. To look up the password, the RADIUS server uses the RADIUS User-Name attribute. All other values are taken from the Digest attributes described in this document. If the calculated digest response equals the string received in the Digest-Response attribute, the authentication was successful. If not, the RADIUS server responds with an Access-Reject.

If the authentication was successful, the RADIUS server calculates a Digest-Response attribute that can be used by the RADIUS client to construct an Authentication-Info header. The calculation of this response is described in [RFC2617], section 3.2.3. RADIUS servers issuing nonces MAY construct a Digest-Nonce attribute. This is
useful to limit the lifetime of a nonce and to save a round-trip in future requests (see nextnonce discussion in [RFC2617], section 3.2.3). The Digest-Response attribute and the optional Digest-Nonce attribute are send to the RADIUS client in an Access-Accept message.
4. Migration Path to DIAMETER

The following table gives an overview of the mapping between RADIUS attributes defined here and the corresponding DIAMETER AVPs described in [I-D.ietf-aaa-diameter-sip-app]:

<table>
<thead>
<tr>
<th>RADIUS</th>
<th>DIAMETER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digest-Realm</td>
<td>Digest-Realm</td>
</tr>
<tr>
<td>Digest-Nonce</td>
<td>Digest-Nonce</td>
</tr>
<tr>
<td>Digest-URI</td>
<td>Digest-URI</td>
</tr>
<tr>
<td>Digest-Domain</td>
<td>Digest-Domain</td>
</tr>
<tr>
<td>Digest-QoP</td>
<td>Digest-Qop</td>
</tr>
<tr>
<td>Digest-Algorithm</td>
<td>Digest-Algorith</td>
</tr>
<tr>
<td>Digest-CNonce</td>
<td>Digest-Cnonce</td>
</tr>
<tr>
<td>Digest-Nonce-Count</td>
<td>Digest-Nonce-Count</td>
</tr>
<tr>
<td>Digest-Method</td>
<td>SIP-Method AVP</td>
</tr>
<tr>
<td>Digest-Username</td>
<td>Digest-Username AVP</td>
</tr>
<tr>
<td>Digest-Body</td>
<td>SIP-Entity-Body-Hash AVP</td>
</tr>
<tr>
<td>Access-Request Digest-Response</td>
<td>SIP-Authorization</td>
</tr>
<tr>
<td>Access-Accept Digest-Response</td>
<td>Digest-Response</td>
</tr>
<tr>
<td>Digest-Opaque</td>
<td>Digest-Opaque AVP</td>
</tr>
<tr>
<td>Digest-Auth-Param</td>
<td>Digest-Auth-Param</td>
</tr>
<tr>
<td>Digest-AKA-Auts</td>
<td>Digest-AKA-Auts</td>
</tr>
<tr>
<td>Digest-Response</td>
<td>Digest-Response AVP</td>
</tr>
<tr>
<td>Digest-Stale</td>
<td>Digest-Stale AVP</td>
</tr>
</tbody>
</table>
Access-Requests containing a Digest-Response attribute and other Digest attributes are mapped to a DIAMETER SIP-Authorization AVP. The RADIUS/DIAMETER gateway constructs a MAR and sends it to the DIAMETER server.

If the authentication was successful, the DIAMETER server replies with a MAA containing a SIP-Authentication-Info and a Digest-Response AVP. The gateway converts these to the corresponding RADIUS attributes and constructs a RADIUS message. If the Result-Code AVP is DIAMETER_SUCCESS or a Digest-Stale AVP is present, an Access-Accept is sent. In all other cases, an Access-Reject is sent.

<table>
<thead>
<tr>
<th>RADIUS</th>
<th>DIAMETER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digest-URI</td>
<td>SIP-AOR</td>
</tr>
<tr>
<td>Digest-Method</td>
<td>SIP-Method</td>
</tr>
</tbody>
</table>

Table 2

If an Access-Request contains a Digest-Method and a Digest-URI attribute but no Digest-Nonce attribute, the gateway maps the RADIUS attributes to DIAMETER according to Table 2. The gateway constructs a MAR message and sends it to the DIAMETER server.

If the MAA contains a SIP-Authenticate-AVP, the gateway maps the contained values to RADIUS attributes, according to Table 1. If the Result-Code AVP is DIAMETER_MULTI_ROUND_AUTH, the gateway constructs an Access-Accept and sends it.
5. IANA Considerations

DIG-RES, DIG-REALM, DIG-NONCE, DIG-METHOD, DIG-URI, DIG-QOP, DIG-ALG, DIG-BODY, DIG-CNONCE, DIG-NC, DIG-USER, DIG-OPAQUE, DIG-AUTHP, DIG-AUTS, DIG-DOMAIN and DIG-STALE are placeholders for values that require IANA registration. They must be assigned from the RADIUS attribute type number space, if this specification becomes a working group document.
6. Security Considerations

The RADIUS extensions described in this document make RADIUS a transport protocol for the data that is required to perform a digest calculation. It adds the vulnerabilities of HTTP Digest (see [RFC2617], section 4) to those of RADIUS (see [RFC2865], section 8 or <http://www.untruth.org/~josh/security/radius/radius-auth.html>)).

If an attacker gets access to a RADIUS client, it can perform man-in-the-middle attacks even if the connections between A, B and B, C (Figure 1) have been secured with TLS or IPSec.

SIP or HTTP requests occur much more frequently than dial-in requests. RADIUS servers implementing this specification must meet that additional performance requirements. An attacker could try to overload the RADIUS infrastructure by excessively sending SIP or HTTP requests. This kind of attack was more difficult when RADIUS was just used for dial-in authentication: the attacker could be identified by the DSL / Cable interface or with some help of the PSTN provider.

To make simple denial of service attacks more difficult, RADIUS clients MUST check if nonces received from a client have been issued by them. This SHOULD be done statelessly. For example, a nonce could consist of a cryptographically random part and some kind of signature of the RADIUS client, as described in [RFC2617], section 3.2.1.

HTTP-style clients can use TLS with server side certificates together with HTTP-Digest authentication. IPSec can be used in a similar way. TLS or IPSec secure the connection while Digest Authentication authenticates the user. If a RADIUS client accepts such connections, it MUST have a secure connection to the RADIUS server.
7. Example

This is an example sniffed from the traffic between HearMe softphone (A), Cisco Systems Proxy Server (B) and deltathree RADIUS server (C) (The communication between Cisco Systems Proxy Server and a SIP PSTN gateway is omitted for brevity):

A->B

INVITE sip:97226491335@213.137.69.38 SIP/2.0
Via: SIP/2.0/UDP 213.137.67.67:5061
From: <sip:12345678@213.137.67.67>;tag=216ae97f
To: sip:97226491335@213.137.69.38
Contact: sip:12345678@213.137.67.67:5061
Call-ID: da591c98-f056-4803-a751-0bd296170875@213.137.67.67
CSeg: 2544265 INVITE
Content-Length: 150
Content-Type: application/sdp
User-Agent: HearMe SoftPHONE

v=0
o=HearMe 2544265 2544265 IN IP4 213.137.67.67
s=HearMe
c=IN IP4 213.137.67.67
t=0 0
m=audio 8000 RTP/AVP 0 4
a=ptime:20
a=x-ssrc:009aa330

B->A

SIP/2.0 100 Trying
Via: SIP/2.0/UDP 213.137.67.67:5061
Call-ID: da591c98-f056-4803-a751-0bd296170875@213.137.67.67
From: <sip:12345678@213.137.67.67>;tag=216ae97f
To: sip:97226491335@213.137.69.38
CSeg: 2544265 INVITE
Content-Length: 0

B->A

SIP/2.0 407 Proxy Authentication Required
Via: SIP/2.0/UDP 213.137.67.67:5061
Call-ID: da591c98-f056-4803-a751-0bd296170875@213.137.67.67
From: <sip:12345678@213.137.67.67>;tag=216ae97f
To: sip:97226491335@213.137.69.38;tag=3f5611de-22a007dc
CSeq: 2544265 INVITE
Proxy-Authenticate: DIGEST realm="deltathree"
   ,nonce="3badala0", algorithm="md5"
Content-Length: 0

A->B

ACK sip:97226491335@213.137.69.38 SIP/2.0
Via: SIP/2.0/UDP 213.137.67.67:5061
From: <sip:12345678@213.137.67.67>;tag=216ae97f
To: sip:97226491335@213.137.69.38;tag=3f5611de-22a007dc
Call-ID: da591c98-f056-4803-a751-0bd2961708758213.137.67.67
CSeq: 2544265 ACK
Content-Length: 0

A->B

INVITE sip:97226491335@213.137.69.38 SIP/2.0
Via: SIP/2.0/UDP 213.137.67.67:5061
From: <sip:12345678@213.137.67.67>;tag=29e97f
To: sip:97226491335@213.137.69.38
Contact: sip:12345678@213.137.67.67:5061
Call-ID: b0f48f7c9-04a0-4108-a5a3-580ecbaf0e24@213.137.67.67
CSeq: 2544266 INVITE
Content-Length: 150
Content-Type: application/sdp
User-Agent: HearMe SoftPHONE
Proxy-Authorization: DIGEST algorithm="md5",nonce="3badala0"
   ,opaque="",realm="deltathree"
   ,response="2ae133421cda65d7dc50d12ba0eb9bc"
   ,uri="sip:97226491335@213.137.69.38",username="12345678"

v=0
o=HearMe 2544265 2544265 IN IP4 213.137.67.67
s=HearMe
p=IN IP4 213.137.67.67
m=audio 8000 RTP/AVP 0 4
a=ptime:20
a=x-ssrc:009aa330

B->C
Code = 1 (Access-Request)
Identifier = 1
Length = 162
Authenticator = 56 7b e6 9a 8e 43 cf b6 fb a6 c0 f0 9a 92 6f 0e
Attributes:
  NAS-IP-Address = d5 89 45 26 (213.137.69.38)
  NAS-Port-Type = 5 (Virtual)
  User-Name = "12345678"
  Digest-Response (DIG-RES) = "2ae133421cda65d67dc50d13ba0eb9bc"
  Digest-Realm (DIG-REALM) = "deltathree"
  Digest-Nonce (DIG-NONCE) = "3bada1a0"
  Digest-Method (DIG-METHOD) = "INVITE"
  Digest-URI (DIG-URI) = "sip:97226491335@213.137.69.38"
  Digest-Algorithm (DIG-ALG) = "md5"
  Digest-Username (DIG-USER) = "12345678"

C->B

Code = 2 (Access-Accept)
Identifier = 1
Length = 20
Authenticator = 6d 76 53 ce aa 07 9a f7 ac b4 b0 e2 96 2f c4 0d
Attributes:
  Digest-Response (206) = "6303c41b0e2c3e524e413cafe8cce954"

B->A

SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 213.137.67.67:5061
From: <sip:12345678@213.137.67.67>;tag=29e97f
To: sip:97226491335@213.137.69.38;tag=7BF5248C-177E
Date: Tue, 25 Jan 2000 03:41:00 gmt
Call-ID: b0f487c9-04a0-4108-a5a3-580ecbaf0e248213.137.67.67
Server: Cisco-SIPGateway/IOS-12.x
Record-Route: <sip:97226491335@213.137.69.38>;maddr=213.137.69.38>
CSeq: 2544266 INVITE
Content-Length: 0

B->A

SIP/2.0 200 OK
Via: SIP/2.0/UDP 213.137.67.67:5061
From: <sip:12345678@213.137.67.67>;tag=29e97f
To: sip:97226491335@213.137.69.38;tag=7BF5248C-177E
Date: Tue, 25 Jan 2000 03:41:00 gmt
Call-ID: b0f487c9-04a0-4108-a5a3-580ecbaf0e24@213.137.137.67.67
Authentication-Info: nextnonce="ef0189c5",
                   rspauth="630341b0e2c3e524e413cafe8c9e954"
Server: Cisco-SIPGateway/IOS-12.x
Record-Route: <sip:972264913350213.137.69.38:5060
            ;maddr=213.137.69.38>
CSeq: 2544266 INVITE
Contact: <sip:97226491335@213.137.69.36:5060;user=phone>
Content-Type: application/sdp
Content-Length: 158

v=0
o=CiscoSystemsSIP-GW-UserAgent 1901 5895 IN IP4 213.137.69.36
s=SIP Call
c=IN IP4 213.137.69.36
t=0 0
m=audio 17724 RTP/AVP 0
a=rtpmap:0 PCMU/8000

A->B

ACK sip:97226491335@213.137.69.36:5060 SIP/2.0
Via: SIP/2.0/UDP 213.137.67.67:5061
From: <sip:12345678@213.137.67.67>;tag=29e97f
To: sip:97226491335@213.137.69.38;tag=7BF5248C-177E
Call-ID: b0f487c9-04a0-4108-a5a3-580ecbaf0e24@213.137.67.67
CSeq: 2544266 ACK
Content-Length: 0
Route: <sip:97226491335@213.137.69.36:5060;user=phone>
8. Changes from -01

- Replaced Sub-attributes with flat attributes
- Aligned naming with [I-D.ietf-aaa-diameter-sip-app]
- Added how a server must treat unknown attributes.
- Added a section 'Migration path to DIAMETER'
- Added an optional attribute for support of the digest scheme described in informational [RFC3310].
- Added a mode of operation where the RADIUS server chooses the nonce. This was required for AKA [RFC3310], but can be useful for ordinary Digest authentication when the qop directive is not used. This required the addition of several attributes.
9. References

9.1 Normative References

[I-D.ietf-aaa-diameter-sip-app]


9.2 Informative References


Authors’ Addresses

Baruch Sterman
Kayote Networks
P.O. Box 1373
Efrat  90435
Israel
EMail: baruch@kayote.com

Daniel Sadolevsky
SecureOL, Inc.
Jerusalem Technology Park
P.O. Box 16120
Jerusalem  91160
Israel
EMail: dscreat@dscreat.com

David Schwartz
Kayote Networks
P.O. Box 1373
Efrat  90435
Israel
EMail: david@kayote.com

David Williams
Cisco Systems
7025 Kit Creek Road
P.O. Box 14987
Research Triangle Park NC 27709
USA
EMail: dwilli@cisco.com

Wolfgang Beck
Deutsche Telekom AG
Am Kavalleriesand 3
Darmstadt  64295
Germany
EMail: beckw@t-systems.com
Appendix A. Acknowledgements

We would like to acknowledge Kevin Mc Dermott (Cisco Systems) for providing comments and experimental implementation.
Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2004). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.