Network Working Group
Internet-Draft
Expires: November 30, 2004

B. Sterman
Kayote Networks
D. Sadolevsky
SecureOL, Inc.
D. Schwartz
Kayote Networks
D. Williams
Cisco Systems
W. Beck
Deutsche Telekom AG

June 2004

RADIUS Extension for Digest Authentication
draft-sterman-aaa-sip-04.txt

Status of this Memo

Copyright Notice

Abstract

Several protocols borrow the authentication mechanisms from the Hypertext Transfer Protocol, HTTP. This document specifies an

extension to RADIUS that allows a RADIUS client in an HTTP-style server, upon reception of a request, retrieve and compute Digest authentication information from a RADIUS server. Additionally, a scenario describing the authentication of a user emitting an HTTP-style request is provided.

Table of Contents

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 limitations and weaknesses of Digest authentication (see [RFC2617], section 4 />), additional PKI-based authentication and encryption mechanisms have been introduced into SIP: TLS [RFC2246] and S/MIME [RFC2633]. The majority of today's SIP clients only supports HTTP digest.

Current RADIUS-based AAA infrastructures have been built and debugged over years. Some 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 the alternatives, 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 Overview

Figure 1 depicts the basic scenario that is relevant for this document. 'HTTP-style Client' and 'RADIUS Client' are entities using a protocol with support for HTTP Digest Authentication, like SIP or HTTP.

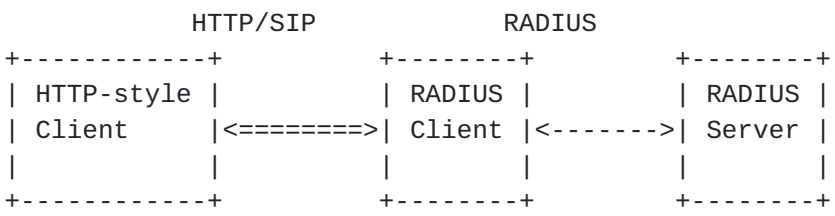


Figure 1: Overview of operation

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.

The nonces required by the digest algorithm are either generated by the RADIUS client or by the RADIUS server. If at least one HTTP-style client requires AKA authentication [RFC3310], the RADIUS server MUST support nonce generation and its RADIUS clients MUST NOT generate nonces locally.

1.3.1 Scenario 1, RADIUS client chooses nonces

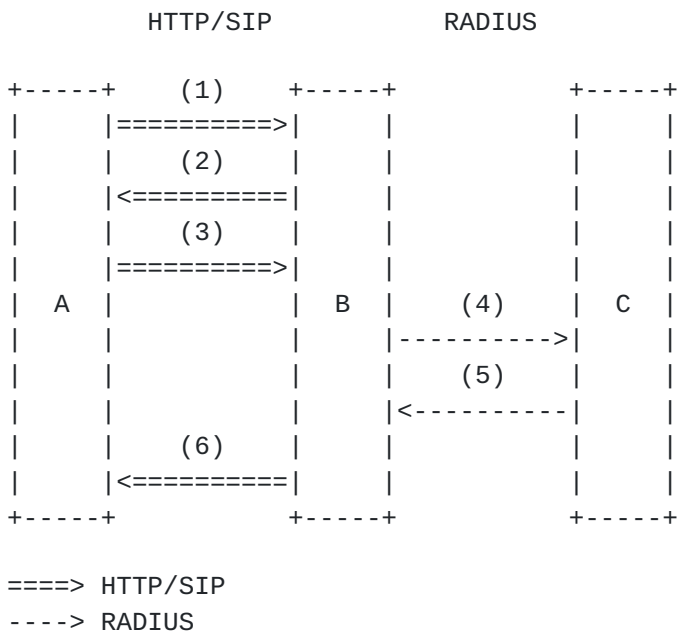


Figure 2: RADIUS client chooses nonces

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).

1.3.2 Scenario 2, RADIUS server chooses nonces

In most cases, the operation outlined in Section 1.3.1 is sufficient. It reduces the load on the RADIUS server to a minimum. However, when using AKA [RFC3310] the nonce is partially derived from a precomputed authentication vector. These authentication vectors are often stored centrally.

Figure 3 depicts a scenario, where the RADIUS server chooses 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.

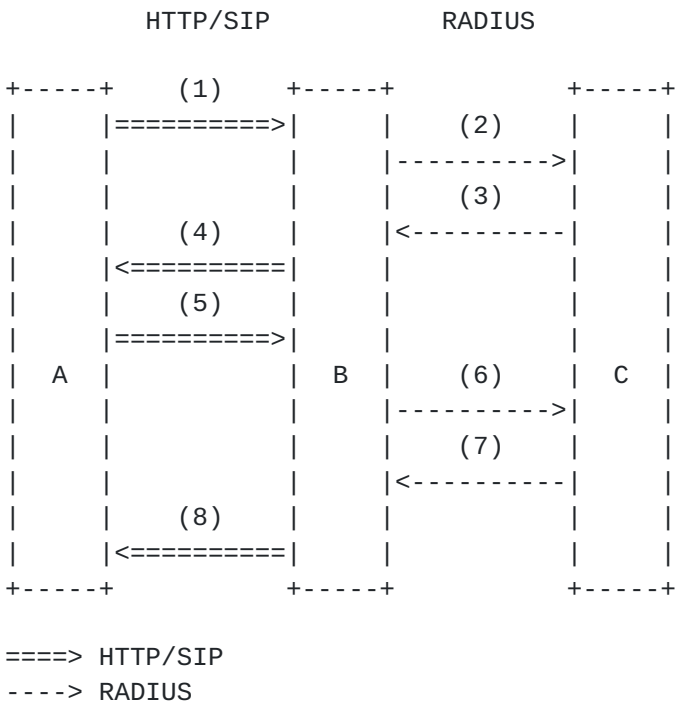


Figure 3: RADIUS server chooses nonces

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:

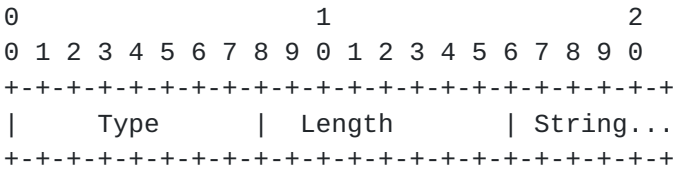
A sends B an HTTP/SIP request without authorization header (step 1). B sends an Access-Request message with the newly defined Digest-Method and Digest-URI attributes but without a Digest-Nonce attribute to the RADIUS server, C (step 2). C chooses a nonce and responds with an Access-Challenge (step 3). This Access-Challenge contains Digest attributes, from which B takes values to construct an HTTP/SIP "(Proxy) Authorization required" response. The remaining steps are identical with scenario 1 (Section 1.3.1): 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).

2. New RADIUS attributes

DIG-RES, DIG-REALM, DIG-NONCE, DIG-RSPAUTH, DIG-NEXTNONCE, DIG-METHOD, DIG-URI, DIG-QOP, DIG-ALG, DIG-BODY, DIG-CNONCE, DIG-NC, DIG-USER, DIG-OPAQUE, DIG-AUTHP, DIG-AUTS, DIG-DOMAIN, DIG-STALE and DIG-HA1 are placeholders for values that are taken from the RADIUS attribute type number space (see Section 5).

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.

If not stated otherwise, the attributes have the following format:



2.1 Digest-Response attribute

If this attribute is present, the RADIUS server SHOULD view the Access-Request as a Digest one. When a RADIUS client receives a (Proxy-)Authorization header, it puts the request-digest value into a Digest-Response attribute.

Type
: DIG-RES for Digest-Response.

Length
: 34

String
: This attribute is only used 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]).

2.2 Digest-Realm attribute

This string attribute describes a protection space of the RADIUS

server. See [RFC2617] 1.2 for details.

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-Challenge messages, the RADIUS server puts the expected realm value into this attribute.

2.3 Digest-Nonce attribute

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

Type
DIG-NONCE
Length
>=3
String
In Access-Requests, the RADIUS client takes the value of the nonce directive (nonce-value) from the HTTP-style request it wants to authenticate. If the Access-Request had a Digest-Method and a Digest-URI but no Digest-Nonce attribute and the RADIUS server is configured to choose nonces, it MUST put a Digest-Nonce attribute into its Access-Challenge message.

2.4 Digest-Response-Auth attribute

Type
DIG-RSPAUTH for Digest-Response-Auth.
Length
34
String
This attribute is only used in Access-Accept messages if the RADIUS server is configured to choose nonces. 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. The RADIUS client puts the string into the rspauth directive of the Authentication-Info header.

2.5 Digest-Nextnonce attribute

This attribute holds a random nonce to be used in the HTTP Digest

calculation.

Type
  DIG-NEXTNONCE
Length
  >=3
String
  If the RADIUS server is configured to choose nonces and to use Authentication-Info, it puts a Digest-Nextnonce attribute into its Access-Accept message. It contains the nonce value that SHOULD be used by the client in the next Access-Request message. The RADIUS client MUST put the contents of this attribute into the nextnonce directive of its HTTP-style response.

2.6 Digest-Method attribute

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

Type
  DIG-METHOD
Length
  >=3
String
  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.

2.7 Digest-URI attribute

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

Type
  DIG-URI
Length
  >=3
String
  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.

2.8 Digest-QoP attribute

This attribute holds the Quality of Protection parameter that

influences the HTTP Digest calculation.

Type
  DIG-QOP
Length
  >=3
String
  In Access-Requests, the RADIUS client takes the value of the qop directive (qop-value) from the HTTP-style request it wants to authenticate. In Access-Challenge messages, the RADIUS server SHOULD put the desired qop-value into this attribute.

2.9 Digest-Algorithm attribute

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

Type
  DIG-ALG
Length
  >=3
String
  In Access-Requests, the RADIUS client takes the value of the algorithm directive from the HTTP-style request it wants to authenticate. In Access-Accept messages, the RADIUS server MAY put the desired algorithm into this attribute.

2.10 Digest-Entity-Body-Hash 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 2. 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.

2.11 Digest-CNonce attribute

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

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-Accept, Access-Challenge or Access-Reject messages.

2.12 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.

2.13 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.

2.14 Digest-Opaque attribute

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

Type
: DIG-OPAQUE

Length
: >=3

String
: 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-Challenge messages, the RADIUS server MAY include this attribute.

2.15 Digest-Auth-Param attribute

This attribute is a placeholder for future extensions.

Type
: 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 a Digest-Auth-Param attribute MUST respond with an Access-Reject message. RADIUS clients that do not implement a parameter contained in a Digest-Auth-Param attribute MUST reject the original HTTP-style request. This attribute MAY be used in Access-Request and Access-Accept messages.

2.16 Digest-AKA-Auts attribute

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

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.

2.17 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-Challenge 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).

Type
      DIG-DOMAIN
Length
      3
String
      The string consists of a single URI, that defines a protection space. RADIUS servers MAY send attributes of this type in Access-Challenge messages. RADIUS clients MUST NOT put attributes of this type in Access-Request messages.

2.18 Digest-Stale attribute

If this attribute is present, the RADIUS server did not accept the nonce value.

Type
      DIG-STALE
Length
      3
String
      The string consists of a single character. If the nonce presented by the RADIUS client was stale, the character is '1' and is '0' otherwise. The attribute MUST be used in Access-Accept messages if the RADIUS server chooses nonces.

2.19 Digest-HA1 attribute

If this attribute is present, the RADIUS server did not accept the nonce value.

Type

DIG-HA1

Length

34

String

This string contains the hexadecimal representation of H(A1) as described in [RFC2617], section 3.2.1 and 3.2.2.2. This attribute is only used in Access-Accept messages. It is used by the RADIUS client to calculate the 'rspauth' directive in an Authentication-Info header when the quality of protection ('qop') is 'auth-int'. Digest-HA1 SHOULD only be sent if the 'algorithm' directive's value is 'MD5-sess' or 'AKAv1-MD5-sess'. This attribute MUST NOT be sent if the qop parameter was not specified or has value of 'auth'. If the 'algorithm' directive's value is 'MD5' or 'AKAv1-MD5', the Digest-HA1 attribute MUST NOT be sent by the RADIUS server or processed by the RADIUS client, unless the authenticity and integrity of the Access-Accept message was secured by cryptographic or equivalently secure means.

3. Detailed Description

3.1 RADIUS Client Behaviour

A RADIUS client without an encrypted or otherwise secured connection (see Section 6) 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-CNonce / 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 server processes the message and responds with an Access-Accept or an Access-Reject message.

The RADIUS clients constructs an Authentication-Info header:

- If the Access-Accept message contains a Digest-Response-Auth attribute, the RADIUS client checks the Digest-QoP attribute:
  - If the Digest-Qop attribute's value is 'auth' or not specified, the RADIUS client puts the Digest-Response-Auth attribute's content into the 'rspauth' directive of the HTTP-style response.
  - If the Digest-Qop attribute's value is 'auth-int', the RADIUS client ignores the Access-Accept message and behaves like it had received an Access-Reject message.
- If the Access-Accept message contains a Digest-HA1 attribute, the RADIUS client checks the Digest-QoP and Digest-Algorithm attributes:
  - If the Digest-Qop attribute is missing or its value is 'auth', the RADIUS client ignores the Digest-HA1 attribute. It does not include an Authentication-Info header into its HTTP-style response.

* If the Digest-Qop attribute's value is 'auth-int' and the Digest-Algorithm attribute's value is 'MD5-sess' or 'AKAv1-MD5-sess', the RADIUS client calculates the contents of the 'rspauth' directive. It creates the HTTP-style response message and calculates the hash of this message's body. It uses the result and the Digest-URI attribute's value of the corresponding Access-Request message to perform the H(A2) calculation. It takes the Digest-Nonce, Digest-Nonce-Count, Digest-CNonce and Digest-QoP values of the corresponding Access-Request and the Digest-HA1 attribute's value to finish the computation of the 'rspauth' value.
* If the Digest-Qop attribute's value is 'auth-int' and the Digest-Algorithm attribute's value is 'MD5' or 'AKAv1-MD5', the RADIUS client MUST NOT use the Digest-HA1 attribute, unless it knows for sure that the Access-Accept message was encrypted or otherwise protected against eavesdropping.

The RADIUS server MAY have added a Digest-Nextnonce attribute. If the RADIUS client discovers this, it puts the contents of this attribute into a 'nextnonce' directive. Now it can send an 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.

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 a Digest-URI attribute but without a Digest-Nonce attribute. The RADIUS server chooses a nonce and responds with an Access-Challenge 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 can send Digest-QoP, Digest-Algorithm, Digest-Realm, Digest-Domain and Digest-Opaque attributes in the Access-Challenge carrying the nonce. If these attributes are present, the client MUST use them.

If the Digest-Stale attribute is present in the Access-Accept message

following an Access-Challenge, the RADIUS client sends an error (401 or 407) response containing WWW-/Proxy-Authenticate header with the directives 'stale' and 'nextnonce'.

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-Challenge message to the RADIUS client. The RADIUS server MUST add Digest-Algorithm, Digest-Realm, SHOULD add Digest-QoP and MAY add Digest-Domain, Digest-Opaque attributes to the Access-Challenge message. 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-Entity-Body-Hash, Digest-CNonce and Digest-AKA-Auts, too. See [RFC2617] and [RFC3310] for details. If it has issued a Digest-Opaque attribute along with the 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 adds an attribute to the Access-Accept message which can be used by the RADIUS client to construct an Authentication-Info header:

- If the Digest-QoP attribute's value is 'auth' or unspecified, the RADIUS server puts a Digest-Response-Auth attribute into the Access-Accept message
- If the Digest-QoP attribute's value is 'auth-int' and the Digest-Algorithm attribute's value is 'MD5-sess' or 'AKAv1-MD5-sess', the RADIUS server puts a Digest-HA1 attribute into the Access-Accept message.
- If the Digest-QoP attribute's value is 'auth-int' and the Digest-Algorithm attribute's value is 'MD5' or 'AKAv1-MD5', the RADIUS server MUST NOT send a Digest-HA1 attribute unless the

connection between RADIUS server and client is encrypted or otherwise protected against eavesdropping.

RADIUS servers issuing nonces MAY construct a Digest-Nextnonce 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-Nextnonce 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]:

RADIUS	Diameter
Digest-Realm	Digest-Realm
Digest-Nonce	Digest-Nonce
Digest-URI	Digest-URI
Digest-Domain	Digest-Domain
Digest-QoP	Digest-Qop
Digest-Algorithm	Digest-Algorithm
Digest-CNonce	Digest-Cnonce
Digest-Nonce-Count	Digest-Nonce-Count
Digest-Method	SIP-Method AVP
Digest-Username	Digest-Username AVP
Digest-Entity-Body-Hash	SIP-Entity-Body-Hash AVP
Digest-Response	SIP-Authorization Digest-Response
Digest-Response-Auth	SIP-Authentication-Info Digest-Response
Digest-Opaque	Digest-Opaque AVP
Digest-Auth-Param	Digest-Auth-Param
Digest-AKA-Auts	Digest-AKA-Auts
Digest-Stale	Digest-Stale AVP

Table 1

4.1 Basic operation

If an Access-Request message 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 construct a MAR message and sends it to the Diameter server.

RADIUS	Diameter
Digest-URI	SIP-AOR
Digest-Method	SIP-Method

Table 2

The Diameter Server responds with a MAA message. This message contains a Result-Code AVP set to the value DIAMETER_MULTI_ROUND_AUTH and challenge parameters. The gateway translates the AVPs and puts the resulting RADIUS attributes into an Access-Challenge message. It sends the Access-Challenge message to the RADIUS client.

The gateway maps an Access-Request message containing a Digest-Response attribute to a MAR message with a Diameter SIP-Authorization AVP. All RADIUS attributes of the Access-Request message are mapped to the corresponding Diameter AVPs. The gateway sends the MAR message 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.

4.2 Limitations

This document covers not all functionality found in [I-D.ietf-aaa-diameter-sip-app].

- There is no equivalent to Diameter's UAR/UAA, SAR/SAA, LIR/LIA, RTR/RTA and PPR/PPA messages
- The operational mode where the Diameter server sends the expected digest response to the client is not possible.

The operational mode where the RADIUS client chooses nonces is not possible with [I-D.ietf-aaa-diameter-sip-app].

5. IANA Considerations

This document serves as IANA registration request for a number of values from the RADIUS attribute type number space:

placeholder	value assigned by IANA
DIG-RES	TBD
DIG-REALM	TBD
DIG-NONCE	TBD
DIG-NEXTNONCE	TBD
DIG-RSPAUTH	TBD
DIG-METHOD	TBD
DIG-URI	TBD
DIG-QOP	TBD
DIG-ALG	TBD
DIG-BODY	TBD
DIG-CNONCE	TBD
DIG-NC	TBD
DIG-USER	TBD
DIG-OPAQUE	TBD
DIG-AUTHP	TBD
DIG-AUTS	TBD
DIG-DOMAIN	TBD
DIG-STALE	TBD
DIG-HA1	TBD

Table 3

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 or RADIUS proxy, it can perform man-in-the-middle attacks even if the connections between A, B and B, C (Figure 2) 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.

RADIUS servers MAY include Digest-QoP and Digest-Algorithm attributes in Access-Accept messages. A man in the middle can modify or remove those attributes in a bidding down attack. In this case, the RADIUS client would use a weaker authentication scheme than intended. Informational RfC 3579 [RFC3579], section 3.2 describes a Message-Authenticator attribute which MAY be used to protect the integrity of RADIUS messages.

The Digest-HA1 attribute contains no random components if the algorithm is 'MD5' or 'AKAv1-MD5'. This makes offline dictionary attacks easier and can be used for replay attacks.

HTTP-style clients can use TLS with server side certificates together with HTTP-Digest authentication. Instead of TLS, IPSec can be used, too. TLS or IPSec secure the connection while Digest Authentication authenticates the user. If a RADIUS client accepts such connections, it MUST have an equally secure connection to the RADIUS server.

7. Example

This is an example sniffed from the traffic between a softphone (A), a Proxy Server (B) and examplecom RADIUS server (C). The communication between the Proxy Server and a SIP PSTN gateway is omitted for brevity. The SIP messages are not shown completely.

```
A->B

   INVITE sip:97226491335@10.0.69.38 SIP/2.0


B->A

   SIP/2.0 100 Trying


B->A

   SIP/2.0 407 Proxy Authentication Required
   Proxy-Authenticate: Digest realm="examplecom"
        ,nonce="3bada1a0", algorithm="md5"
   Content-Length: 0


A->B

   ACK sip:97226491335@10.0.69.38 SIP/2.0


A->B

   INVITE sip:97226491335@10.0.69.38 SIP/2.0
   Proxy-Authorization: Digest algorithm="md5",nonce="3bada1a0"
        ,opaque="",realm="examplecom"
        ,response="2ae133421cda65d67dc50d13ba0eb9bc"
        ,uri="sip:97226491335@10.0.69.38",username="12345678"


B->C

   Code = 1 (Access-Request)
   Attributes:
   NAS-IP-Address = a 0 45 26 (10.0.69.38)
   NAS-Port-Type = 5 (Virtual)
   User-Name = "12345678"
```

```
      Digest-Response (DIG-RES) = "2ae133421cda65d67dc50d13ba0eb9bc"
      Digest-Realm (DIG-REALM) = "examplecom"
      Digest-Nonce (DIG-NONCE) = "3bada1a0"
      Digest-Method (DIG-METHOD) = "INVITE"
      Digest-URI (DIG-URI) = "sip:97226491335@10.0.69.38"
      Digest-Algorithm (DIG-ALG) = "md5"
      Digest-Username (DIG-USER) =  "12345678"


   C->B

      Code = 2 (Access-Accept)
      Attributes:
      Digest-Response-Auth (DIG-RSPAUTH) =
                  "6303c41b0e2c3e524e413cafe8cce954"


   B->A

      SIP/2.0 180 Ringing


   B->A

      SIP/2.0 200 OK


   A->B

      ACK sip:97226491335@10.0.69.38:5060 SIP/2.0
```

A second example shows the traffic between a web browser (A), web server (B) and a RADIUS server (C).

```
   A->B

      GET /index.html HTTP/1.1


   B->A

      HTTP/1.1 407 Authentication Required
      WWW-Authenticate: Digest realm="examplecom", domain="/index.html",
           ,nonce="a3086ac8", algorithm="md5"
```

```
      Content-Length: 0


   A->B

      GET /index.html HTTP/1.1
      Authorization: Digest algorithm="md5",nonce="a3086ac8"
           ,opaque="",realm="examplecom"
           ,response="369b593b9a79e001256a2b40afe49f4c"
           ,uri="/index.html",username="12345678"


   B->C

      Code = 1 (Access-Request)
      Attributes:
      NAS-IP-Address = a 0 45 26 (10.0.69.38)
      NAS-Port-Type = 5 (Virtual)
      User-Name = "12345678"
      Digest-Response (DIG-RES) = "369b593b9a79e001256a2b40afe49f4c"
      Digest-Realm (DIG-REALM) = "examplecom"
      Digest-Nonce (DIG-NONCE) = "a3086ac8"
      Digest-Method (DIG-METHOD) = "GET"
      Digest-URI (DIG-URI) = "/index.html""
      Digest-Algorithm (DIG-ALG) = "md5"
      Digest-Username (DIG-USER) =  "12345678"


   C->B

      Code = 2 (Access-Accept)
      Attributes:
      Digest-Response-Auth (DIG-RSPAUTH) =
          "e644aa513effbfe1caff67103ff6433c"


   B->A

      HTTP/1.1 200 OK
      ...

      <html>
      ...
```

8. Changes from draft-sterman-aaa-sip-03

- addressed 'auth-int' issue
- New Digest-Nextnonce attribute
- revised abstract, motivational section and examples
- Access-Challenge instead of 'Access-Accept carrying a Digest-Nonce attribute'
- shortened SIP messages in example, removed real-world addresses and product names

9. Changes from draft-sterman-aaa-sip-02

- Relaxed restrictions for DIG-DOMAIN, DIG-REALM, DIG-OPAQUE, DIG-QOP and DIG-ALG
- Additional security considerations for DIG-DOMAIN, DIG-QOP and DIG-ALG usage in Access-Accept messages

10. Changes from draft-sterman-aaa-sip-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.

11. References

11.1 Normative References

[I-D.ietf-aaa-diameter-sip-app]
Garcia-Martin, M., Belinchon, M., Pallares-Lopez, M., Canales-Valenzuela, C. and K. Tammi, "Diameter Session Initiation Protocol (SIP) Application", draft-ietf-aaa-diameter-sip-app-03 (work in progress), July 2004.

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

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A. and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

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

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

11.2 Informative References

[RFC1750] Eastlake, D., Crocker, S. and J. Schiller, "Randomness Recommendations for Security", RFC 1750, December 1994.

[RFC2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[RFC2633] Ramsdell, B., "S/MIME Version 3 Message Specification", RFC 2633, June 1999.

[RFC3310] Niemi, A., Arkko, J. and V. Torvinen, "Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)", RFC 3310, September 2002.

[RFC3579] Aboba, B. and P. Calhoun, "RADIUS (Remote Authentication Dial In User Service) Support For Extensible

Authentication Protocol (EAP)", RFC 3579, September 2003.

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

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 Mcdermott (Cisco Systems) /or providing comments and experimental implementation.

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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