SIP Working Group
Internet Draft
Category: Standards Track

Expires on Dec 2002

James Undery
Ubiquity

Sanjoy Sen
Nortel Networks

Vesa Torvinen
Ericsson

June 2002

Enhanced Usage of HTTP Digest Authentication for SIP

<draft-undery-sip-auth-01.txt>

Status of this Memo

Abstract

HTTP Digest has some shortcomings if applied for SIP. Firstly, SIP UA has serious difficulties to distinguish the source of Authentication-Info and Proxy-Authentication-Info headers in SIP forking situations. This is due to the absence of the ærealmÆ parameter in these headers. Secondly, HTTP authentication is particularly vulnerable against MITM bid-down attacks on the list of algorithms (e.g., MD-5, SHA-1) or the desired security level (auth, auth-int). Thirdly, HTTP authentication provides limited integrity protection of only the message body. In SIP, important information can be carried in many of the headers that may need integrity protection. This draft proposes to add the realm parameter in the *-Authentication-Info headers, recommends a format for computing the

nonce for detection of bid-down attack and proposes a mechanism for integrity protection of SIP headers using MIME body.

1 Introduction

HTTP Digest [3] has some shortcomings if applied for SIP. RFC 3261 [2] allows the use of Authentication-Info header in responses for mutual authentication between the client and the server. As these headers currently do not contain the ærealmÆ parameter, the client has serious difficulties to distinguish the source of Authentication-Info headers if the SIP request is forked. Thus there is no way for the client to match the credential carried in the Authentication-Info header with the corresponding challenge.

HTTP authentication is also vulnerable against MITM bid-down attacks, possibly on the algorithms (e.g., MD-5, SHA-1) or on the protection levels (auth, auth-int). A MITM attacker can easily remove the stronger mechanism among the existing mechanisms in the challenge (e.g., remove æauth-intÆ from a challenge containing both æauthÆ and æauth-intÆ). Also, HTTP authentication provides limited integrity protection of only the message body. In SIP, important information can be carried in many of the headers that may need integrity protection. This draft proposes to add the realm parameter in the *-Authentication-Info headers, recommends a format for computing the nonce for detection of bid-down attack and proposes a mechanism for integrity protection of SIP headers using MIME body.

2 Conventions used in this document

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 RFC-2119.

3 Syntax for Authentication-Info header

The Authentication-Info header is used by the server to communicate some information regarding the successful authentication in the response. The modified syntax is shown. The two new parameters added are ærealmÆ and æauth-paramÆ.

```
AuthenticationInfo = "Authentication-Info" ":" auth-info
auth-info          = 1#(nextnonce | realm | [message-qop]
                     | [response-auth ] | [cnonce]
                     | [nonce-count] | [auth-param])
nextnonce          = "nextnonce" "=" nonce-value
response-auth      = "rspauth" "=" response-digest
response-digest    = <"> *LHEX <">
```

realm
A string contained in the server challenge that can be rendered for the end user to provide the context with which to authenticate itself. The ærealmÆ parameter in Authentication-Info header SHOULD contain the same value as the corresponding element carried in the most recent challenge from the server.

auth-param
This is included for future extensions; unknown values should be ignored.

The same syntax changes apply to the Proxy-Authentication-Info header that can be added by the Proxies in responses to UAC.

4 Recommendation for nonce construction

Traditionally nonces have contained no meaning for the client, however, in order to detect bid-down attacks this draft will recommend a format. This format is designed to allow a server to encode the list that needs to be protected against bid-down attack.

The following definition will replace nonce-value:

```
nonce-value      = LDQUOT "(" 1#auth-prefix ")"
                   trad-nonce-value RDQUOT
auth-prefix      = auth-algorithms | digest-auth-type | token
auth-algorithms  = "MD5" | "AKA" | "SHA1"
auth-type        = "auth" | "auth-int"
trad-nonce-value = *(qdtext | quoted-pair)
```

auth-algorithm
These are the algorithms used by the Digest scheme to produce the digest.

auth-type
These are the protection levels for Digest authentication. Each value corresponds to a qop-value.

The client compares the list of algorithms or protection level values with that encoded in the nonce to detect a bid-down attack. If the attacker modifies the auth-prefix in the nonce, then the digest credential computed by the server with the original nonce will not match that in the Digest response and the attack will be detected.

A possible implementation of trad-nonce-value is:

```
trad-nonce-value = time-stamp "-" H(time-stamp ":" request-uri
":" private-key)
```

where, the time-stamp is a non repeating value, the request-uri is the Request URI from the request and the private-key is to ensure that the nonce was generated by an entity that knows the private-key. The auth-prefix MAY be included in the hash function above. The inclusion of the auth-prefix in the hash is only really useful if the server generating the challenge varies the challenge on a request-by-request basis and does not want to reevaluate its policy rules.

5 Integrity protection of headers by the client

If the client wants to integrity protect certain headers and if the server supports the level of protection æauth-intÆ, the client can do so by using the process of including headers in the message body by using MIME type æmessage/sipfragÆ, as described in [6] or by using the mechanism of tunneling entire SIP messages by using MIME type æmessage/sipÆ as discussed in [2].

A valid æmessage/sipfragÆ message body is formed by taking the original SIP message and deleting either (1) entire start-line, or (2) one or more complete headers, or (3) entire body. The qop parameter in the Authorization or Proxy-Authorization header MUST be set to æauth-intÆ. This can be used to provide hop-by-hop, end-to-end, end-to-middle or middle-to-end integrity protection of selected headers using Digest. Some of the headers that can be integrity protected by this mechanism are: From, To, Call-ID, Contact, CSeq, Expires etc. An example of using this mechanism to protect a SIP header (Security-Verify) is discussed in [7]. The MIME type æmessage/sipÆ can be used for integrity protection of entire SIP message.

The same mechanism can also be used by the server (or Proxy) to integrity protect headers, the response Status Code or the entire response message using the Authentication-Info (or Proxy-Authentication-Info) header.

6 Client and Server Behavior

The server MUST include the ærealmÆ parameter in the Authentication-Info header. The server MUST ignore any æauth-paramÆ value that it does not understand. Otherwise, the semantic for the Authentication-Info header is the same as in [3].

For bid-down attack detection, the client MUST compare the list of algorithms or protection levels in the challenge with those encoded in the nonce. The server MUST store the original nonce that it had sent in the challenge and use it to compute the credential to match against the credential sent by the client.

If the client wants to integrity protect one or more headers of the SIP message, then it MUST use the process of including headers in the message body by using MIME type æmessage/sipfragÆ as described in [6]. If the client wants to integrity protect the entire SIP message, then it MUST use the process of including entire SIP message in the message body (tunneling) using MIME type æmessage/sipÆ as discussed in [2]. In both cases, the qop parameter in the Authorization or Proxy-Authorization header MUST be set to æauth-intÆ. The client MUST use the mechanism for computing Digest credential for qop = auth-int, as described in [3].

7 Security Considerations

The purpose of this draft is security. Items ruled out of scope of this document are privacy and resistance to denial of service attacks. Since this draft either proposes fix to RFC 2617 headers or discusses application of RFC 2617 for SIP, most of the security considerations discussed in [3] are applicable here.

Note that, two of the mechanisms proposed in this draft û bid-down detection and integrity protection of headers / entire message û have to be initiated by the client. There is no way for the server or the Proxy to request that the client uses these mechanisms. [7] discusses a way that can be used by the server (or Proxy) to negotiate the use of these mechanisms with the client.

The bid-down protection mechanism does not include algorithm revocation mechanisms. The result of this is if a one-way hashing algorithm is broken such that a message s + m can be recovered from KD(s + m) if m is known; or given a message s + m, it is possible to find n such that KD(s + m) = KD(s + n) for constant unknown s. Then this mechanism will fail, although in the second case m and n will have to be syntactically equivalent.

7.1 Security Considerations Missing From RFC 2617

RFC 2617 [3] has a remarkably thorough security considerations section, however, in our opinion an important consideration is missed. In the WWW-Authenticate header the qop directive can contain a list of schemes supported. It is possible for an attacker to downgrade the security on offer by removing auth-int if present so the body of the message is not included in the protection, or

simply remove the qop parameter entirely.

8 Future Work

Future work on this topic should include the following:

- A mechanism for the server to recommend the list of headers that needs to be integrity protected
- A way for the server to specify the protection mechanism to be used in the challenge
- Authentication of Proxies by the UAS

9 References

1 Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

2 Handley, M., Schulzrinne, H, Schooler, E. and Rosenberg, J., "SIP: Session Initiation Protocol", RFC 3261.

3 Franks, J. et al, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

4 Bellovin, S., "Report of the IAB Security Architecture Workshop", RFC 2316, April 1998.

5 Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

6 Sparks, R., "Internet Media Types message/sip and message/sipfrag", draft-sparks-sip-mimetypes-01 (work in progress), March 2002.

7. Arkko, et. Al., "Security Mechanism Agreement for SIP Sessions", draft-ietf-sip-sec-agree-02.txt, May 2002.

10 Acknowledgments

The authors acknowledge that the idea of using æmessage/sip-fragÆ to provide integrity protection of SIP headers using Digest was proposed by Henning.

11 Authors' Addresses

James Undery
Ubiquity Software Corporation Ltd.
Ubiquity House

Langstone Park
Newport, UK
Email: jundery@ubiquity.net

Sanjoy Sen
Nortel Networks
sanjoy@nortelnetworks.com
Richardson, Texas
Email: sanjoy@nortelnetworks.com

Vesa Torvinen
Oy LM Ericsson Ab
Email: vesa.torvinen@ericsson.fi

12 Full Copyright Statement