Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)
draft-ietf-sip-identity-00

Status of this Memo

This Internet-Draft will expire on April 28, 2003.

Copyright Notice

Abstract

The existing mechanisms for expressing identity in the Session Initiation Protocol oftentimes do not permit an administrative domain to verify securely the identity of the originator of a request. This document recommends practices and conventions for authenticating end users, and proposes a way to distribute cryptographically secure authenticated identities within SIP messages.

Table of Contents

1. Introduction

This document provides enhancements to the existing mechanisms for authenticated identity management in the Session Initiation Protocol (SIP) [1]. An identity, for the purposes of this document, is defined as a canonical SIP URI employed to reach a user (such as 'sip:alice@atlanta.com').

RFC3261 enumerates a number of places within a SIP request that a user can express an identity for themselves, notably the From header field. However, the recipient of a SIP request has no way to verify that the From header field has been populated appropriately without some sort of cryptographic authentication mechanism.

Today, RFC3261 specifies a number of security mechanisms that can be used by SIP UAs, including Digest, TLS and S/MIME (and implementations may support other security schemes as well). However, few SIP user agents today can support the end-user certificates necessary to authenticate themselves via TLS or S/MIME, and Digest authentication is limited by the fact that the originator and destination must share a secret. It is desirable for SIP user agents to be able to send requests to destinations with they have no previous association - just as in the telephone network today, one can receive a call from someone with whom one has no previous association, and still have a reasonable assurance that their displayed Caller-ID is accurate.

Many SIP user agents today support a means of authenticating themselves to a SIP registrar - commonly with a shared secret (Digest, which MUST be supported by SIP user agents, is typically used for this purpose). Registration allows a user agent to express that it is the proper place to which requests should be sent for a particular address-of-record SIP URI. However, the credentials with which a user agent proves to a registrar that they are, for example, an authorized recipient of requests for 'sip:alice@atlanta.com' cannot be shared with a server in another domain - these credentials are currently only useful for local registration.

Coincidentally, the address-of-record URI of a SIP user is also the URI with which a SIP UA populates the From header of requests from that user - in other words, the address-of-record is an identity. So it turns out that users already have a means of providing their identity, if only to a registrar; in fact, since the contents of a From header field are essentially a 'return address' for SIP requests, being able to prove that you are eligible to receive requests for that 'return address' is identical to proving that you are authorized to assert this identity. In other words, the best way for a SIP user to prove that they can legitimately claim an identity

is to provide the same credentials they would need to provide in order to register to receive requests for that identity; the two have the same security properties. Since the operator of the registrar controls the namespace of local SIP users (assigning the user portion of SIP URIs in the domain), it is the ideal arbiter for identity in SIP.

In the absence of end-user certificates in user agents, it is possible to implement a mediated authentication architecture for SIP in which requests are sent to a server in the user's local domain which authenticates them (using the same practices by which the domain would authenticate REGISTER requests). Once a request has been authenticated, the local domain then needs some way to communicate to remote domains that it has sanctioned the request. This draft addresses how that identity can could be securely shared.

RFC3261 already describes an architecture very similar to this in Section 26.3.2.2, in which a user agent authenticates itself to a local proxy server which in turn authenticates itself to a remote proxy server via mutual TLS, creating a two-link chain of transitive authentication between the originator and the remote domain. While this works well in some architectures, there are a few respects in which this is impractical. For one, it is possible for SIP requests to cross multiple intermediaries in separate administrative domains, in which case transitive trust becomes far less compelling. It also requires intermediaries to act as proxies, rather than redirecting requests to their destinations (redirection lightens loads on SIP intermediaries). Both of these limitations result from the fact that authentication takes place outside the application, at the transport layer, rather than within SIP itself.

One solution to this problem is to use 'trusted' SIP intermediaries that assert an identity for users in the form of a privileged SIP header. A mechanism for doing so (with the P-Asserted-Identity header) is given in [6]. However, this solution allows only hop-by-hop trust between intermediaries, not end-to-end cryptographic authentication, and it assumes a managed network of nodes with strict mutual trust relationships, an assumption that is incompatible with widespread Internet deployment.

The desired mediated authentication architecture has quite a bit in common with the problem space of Kerberos [5]. Ideally, there should be a way for a user to authenticate themselves to the local domain, and receive some kind of token that they can share with recipients of requests that lets them know that the user has been authenticated by the local domain. However, Kerberos support in SIP user agents is not widespread, and moreover SIP uses other means (such as Digest) to perform key authentication functions already. An ideal solution

would adapt existing SIP security mechanisms to address this problem.

Therefore, this document defines a new logical role for SIP network intermediaries (typically proxy servers) called an 'authentication service'. Once an authentication service has verified the identity of the originator of a request, as describe above, it creates a cryptographic token that contains the authenticated identity of the user, and which has some reference integrity with the request itself. This token can then be added to a SIP request and inspected by recipients of the request who would like a cryptographic guarantee of the identity of the user.

One possible format for such tokens is the Authenticated Identity Body (AIB) described in [4]. Other plausible token formats are a matter for further investigation. Throughout this document, the use of the AIB format as a token is considered exclusively.

2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in RFC2119 [2] and indicate requirement levels for compliant SIP implementations.

3. Using an Authentication Service

A SIP user agents sends requests to an authentication service in order to receive an authentication token for the request. How exactly the association with an authentication service is configured is an implementation-specific matter for the user agent - it might be implemented with a pre-loaded Route header. The guidelines given in RFC3261 Sections 26.3.2.1 and 26.3.2.2 should be used when connecting to an authentication service; ideally, an authentication service should be one hop away from a user agent, it should use a lower-layer security protocol such as TLS or IPSec to authenticate the authentication service before providing credentials (especially shared secrets).

This document places no requirements on how an authentication services authenticates requests. Since Digest authentication MUST be supported by all SIP entities, the use of Digest for this purpose is likely.

4. How to Share Verified Identities

When an authentication service has authenticated the user, it must construct an identity for that user that will be contained in the

token. It is RECOMMENDED that these identities take the form of addresses-of-record, as they are defined in Section 10 of RFC3261; in other words, URIs of the form 'sip:alice@atlanta.com'.

This identity must be expressed in the authentication token that will be signed by the authentication service. For example, if the Authentication Identity Body (AIB) format described in [4] is used, this identity would be stored in the From header field within a 'message/sip' or 'message/sipfrag' [7] body that will be signed by the authentication service.

In some cases, an authentication service will hold a certificate corresponding to each user in its administrative domain (in other words, a certificate whose subjectAltName contains a URI equivalent to the address-of-record URI of the user). The appropriate certificate for the authenticated user will be used to sign the authentication token. However, an authentication service MAY instead use a common certificate for its administrative domain. The subjectAltName of this certificate MUST correspond with the host portion of the From header field of the identity in the authentication token (if the identity were 'sip:alice@atlanta.com', the subjectAltName of the certificate would be 'atlanta.com'); this should be the same certificate that the authentication service provides when proving its own identity (via TLS or some similar protocol). Maintaining individual certificates for each user is RECOMMENDED, since the name subordination rules involved with the use of a common certificate for the domain can become complicated.

After the authentication token has been signed, the authentication token MUST be added to any existing MIME bodies in the request, if necessary by transitioning the outermost MIME body to a 'multipart/mixed' format. Two options are considered for ways that an authentication token could be added to a SIP message: one in which the authentication service adds the token itself, and one in which it pushes the token back to the client, essentially asking the client to include the token in a retry of the request. Another possibility, using content indirection, is mentioned as a direction for future work. Authentication services MUST support the mechanism in Section 4.2 and MAY support the mechanism in Section 4.1.

4.1 Body Added by Authentication Service

The first possibility is that the authentication service could add the body to the request itself before forwarding the request. However, the authentication service role is usually played by entities that act as proxy servers for most requests, and proxy servers cannot modify message bodies (see RFC3261 Section 16.6). In order to add an authentication token, the authentication service

needs to act as a transparent back-to-back user agent, effectively terminating the request and re-originating it with a new body appended to any existing MIME bodies (again, transposing to various MIME multipart forms as necessary).

This mechanism has some potential advantages over push the authentication token back to the originating user agent. For one, it saves on additional round-trip times for signaling that result from passing the body back to the user agent. It also requires no new SIP mechanisms, whereas any method of asking a user agent to include a body in a resubmission to the current request would introduce new protocol requirements.

However, there are proposed SIP integrity mechanisms that place a signature over the entire message body in a SIP message header. Were a server to modify the body of a message that was protected by such signature, that would be perceived as an integrity violation by downstream recipients of the message. Presumably, a back-to-back user agent function would have to sacrifice this end-to-end integrity.

4.2 Body Added by Client

Alternatively, the authentication service could in some fashion return the authentication token to the originating user agent, prompting the user agent to retry the request with the authentication token attached. No existing SIP mechanism performs this function. Therefore, this document defines a 428 "Use Authentication Token" response code.

An authentication service sends a 428 with a MIME body in order to request that a user agent add the enclosed MIME body to their request and retry the request. A 428 MUST have at most a single MIME body. This MIME body MUST be signed by the authentication service.

The use of 428 without any MIME body is also defined in this document. It can be sent by any server to reject a request because the request does not contain an authentication token. A user agent receiving this rejection SHOULD retry their request through an authentication service.

In order to signal to the authentication services that the originating user agent supports the receipt of the 428 response code, a new option-tag has been defined, the 'auth-id' option-tag. User agents SHOULD supply the 'auth-id' option-tag in a Supported header whenever they provide credentials to a server (for example, in Digest authentication, whenever a Proxy-Authorization header is added to a request).

Using the 428 response code may introduce extra round-trip times for messages, delaying the setup of requests (one RTT for the 407, another for the 428). However, there are some circumstances under which extra RTTs may not impede performance. If the originating user agent possesses a non-stale nonce (assuming Digest authentication) from the authentication service, it can pre-emptively include a Proxy-Authorization header, eliminating one RTT. With regard to the second RTT, note that the request needn't necessarily go through the authentication service again once the authentication token has been added - it could go directly to its destination, which reduce the impact of the second RTT.

There are two reasons why the originating user agent should be the party responsible for adding the authentication token to the request. Firstly, because this gives the client the opportunity to inspect the body itself (perhaps only to see whether or not it is encrypted; see [4]) in order to verify that the authenticated identity corresponds with the provided credentials and the user's preferences. Secondly, the client can provide a signature over the entire body of the message (either with S/MIME or some header-based mechanism) so that the final recipient of messages can be assured that all information in the body is there at the originator's behest.

4.3 Using Content Indirection

Work [8] is currently underway in the SIP WG to define a content indirection mechanism for SIP, a mechanism by which a MIME body in a SIP request can refer, with a URL, to a document that it hosted somewhere in the network. This raises another interesting possibility for authentication token management.

A SIP user agent could create a content indirection MIME body (using the RFC2017 [9] URL MIME External-Body Access-Type) that contains a URL that identities a resource controlled by the authentication service, anticipating that the authentication service will make the authentication token available at that URL. Authentication services could define a standard way to anticipate URIs for a particular request (for example, an HTTP URL could be structured with a hostname corresponding to the authentication service, and a path corresponding to a unique identifier selected by the user agent for this request: http://auth-serv.biz/12345678901234567890). Once an authentication service has validated the request, it simply makes the authentication token available at the anticipated URL; recipients of the message would then dereference the URL in order to inspect the token.

This approach could allow user agents to have full control over the integrity of SIP requests, while still not requiring the extra RTT caused by the use of the 428 response code. It also has numerous

advantages over other ways of handling authentication tokens issued for SIP response messages (see Section 5).

5. Identity in Responses

Many of the practices described in the preceding sections can be applied to responses as well as requests, with some important differences. Primarily, the distinction is that a response cannot be challenged or resubmitted in the same manner as a request. However, when a user agent registers under a particular identity, and thereby becomes eligible to receive requests and send responses associated with that identity, it provides credentials that prove its identity, and thus the registrar is in a reasonable position to act as an authentication service for responses.

Note that the identity in an authentication token in a response almost certainly will not correspond with identity asserted in the From header field of the response (which is copied from the Request); the identity in the authentication token represents a different entity. In many network implementations, the identity in the authentication token of a response will correspond to the To header field of the request, but there are numerous legitimate architectures (which contain redirections) in which this will not be the case.

An authentication service that acts as a registrar can add to a response an authentication token that corresponds to the identity of the originator of that response in roughly the same manner described in Section 4.1 - the authentication service adds the authentication token to a response before it forwards the response towards the originator of the request. There is no way for an authentication service to perform a function for responses comparable to the mechanism described in Section 4.2; however, content indirection (see Section 4.3 could provide an alternative that would allow the client to retain end-to-end integrity properties on responses.

6. Receiving an Authentication Token

The manner in which an authentication token is handled is dependent on the nature of the token itself; for rules for handling the Authentication Identity Body (AIB) format, see [4].

6.1 Authentication Service Handling of Authentication Tokens

SIP intermediaries generally should not attempt to inspect MIME bodies; following the rules of RFC3261 Section 16.6, MIME bodies may be encrypted end-to-end or have other properties that make them unsuitable for consumption by intermediaries. However, intermediaries that implement the authentication service logical role

MAY inspect MIME bodies in order to find one with a Content-Disposition of 'auth-id'.

For the most part, the actual value of an authenticated identity is not likely to be of interest to a proxy server, though it MAY refuse to process a request that does not contain a valid authentication token (using the 428 request, as described in Section 4.2). However, proxy servers MAY additionally maintain lists of known problem users that are banned from making requests to their administrative domain, for example, and subsequently reject some requests after comparing their authenticated identities to such access control lists.

7. Selective Sharing of Identity

Most of the time, there is no need to restrict the propagation of verified identities in the network. User agents and intermediaries benefit from receiving verified identities. However, in some cases intermediaries might want to restrict the distribution of identity information, for example if

o the authenticated identity body contains an identity that is only meaningful as an internal identifier within a particular service provider's network, or,

o the originating user agent has requested privacy, and the unrestricted distribution of the authenticated identity body would violate that request.

If it is not appropriate to share an authenticated identity, an authenticated identity body SHOULD NOT be created and distributed. However, in some cases there may be other entities in the administrative domain of the authentication service that are consumers of the authenticated identity. If, for example, each of these servers needed to challenge the user individually for identity, it might significantly delay the processing of the request. For that reason, it may be appropriate to circulate authenticated identity bodies among a controlled set of entities. For that purpose, an encryption mechanism for authenticated identities is required.

7.1 Requesting Privacy

When users authenticate themselves to an authentication service, they MAY use SIP to explicitly notify the service that they do not wish their authenticated identity to be circulated. Usually, the user in question would also be taking other steps to preserve their privacy (perhaps by including an anonymous From header in the SIP request, and following other standard privacy practices).

Authentication services MUST support the privacy mechanism described in [3]. Users requesting privacy should also support the mechanisms described in that document.

In particular, this document uses an identity-specific priv-value that can appear in the Privacy header, a value of 'id', which was registered by [6]. This Privacy value requests that the results of authentication should not be shared by the authenticating server. An authentication service SHOULD NOT create an authentication token for a request when 'id' privacy has been requested. If such a token is created, it MUST be encrypted or rendered confidential in the manner most appropriate to the token. Guidelines for encrypting AIBs are given in [4], and these mechanisms MUST be supported by any authentication service that uses AIBs.

8. Security Considerations

Users SHOULD NOT provide credentials to an authentication service to which they cannot initiate a direct connection, preferably one secured by a network or transport layer security protocol such as TLS. If a user does not receive a certificate from the authentication service over this lower-layer protocol that corresponds to the expected domain (especially when they receive a challenge via a mechanism such as Digest), then it is possible that a rogue server is attempting to pose as a authentication service for a domain that it does not control, possibly in an attempt to collect shared secrets for that domain. If a user cannot connect directly to the desired authentication service, the user SHOULD at least use a SIPS URI to ensure that mutual TLS authentication will be used to reach the remote server.

Relying on an authentication token generated by a remote administrative domain assumes that the domain uses some trustworthy practice to authenticate its users. However, it is possible that some domains will implement policies that effectively make users unaccountable (such as accepting unauthenticated registrations from arbitrary users). Therefore, it is RECOMMENDED that authentication tokens contain some indication of the specific practice (for example, Digest) that was used to authenticate this request as a rough indicator of credential strength.

If a common certificate is used by an authentication service (rather than individual certificates for each identity), certain problems can arise with name subordination. For example, if an authentication service holds a common certificate for the hostname 'sip.atlanta.com', can it legitimately sign a token containing an identity of 'sip:alice@atlanta.com'? It is difficult for the recipient of a request to ascertain whether or not 'sip.atlanta.com'

is authoritative for the 'atlanta.com' domain unless the recipient has some foreknowledge of the administration of 'atlanta.com'. Therefore, it is RECOMMEND that user agent recipients of authentication tokens notify end users if there is ANY discrepancy between the subjectAltName of the signers certificate and the identity within the authentication token.

Authentication tokens MUST have some form of replay protection. The best protection is to copy the SIP request in its entirety (via the 'message/sip' MIME type) into the authentication token - in that way, it will be clear that this token has been issued for this request, since collectively the headers of a SIP request provide a unique identifier. However, SIP requests can be large, and it is reasonable to include only a subset of the SIP headers in a request (using the 'message/sipfrag' MIME type) as long as certain critical headers are provided. For further discussion of this issue, including guidelines for including particular headers in a sipfrag, see [4].

Because the common certificates that can be used by authentication services need to assert only the hostname of the authentication service, existing certificate authorities can provide adequate certificates for this mechanism. However, not all proxy servers and user agents will be able support the root certificates of all certificate authorities, and moreover there are some significant differences in the policies by which certificate authorities issue their certificates. This document makes no recommendations for the usage of particular certificate authorities, nor does it describe any particular policies that certificate authorities should follow, but it is anticipated that operational experience will create de facto standards for the purposes of authentication services. Some federations of service providers, for example, might only trust certificates that have been provided by a certificate authority operated by the federation.

9. IANA Considerations

This document defines a new SIP status code, '428 Use Authentication Token'. The use of this status code is further described in Section 4.2.

Normative References

[1] 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.

[2] Bradner, S., "Key words for use in RFCs to indicate requirement levels", RFC 2119, March 1997.

[3] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", draft-ietf-sip-privacy-general-02 (work in progress), June 2002.

[4] Peterson, J., "SIP Authenticated Identity Body (AIB) Format", draft-ietf-sip-authid-body-00 (work in progress), October 2002.

Informative References

[5] Kohl, J. and C. Neumann, "The Kerberos Network Authentication Service (V5)", RFC 1510, September 1993.

[6] Jennings, C., Peterson, J. and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", draft-ietf-sip-asserted-identity-02 (work in progress), June 2002.

[7] Sparks, R., "Internet Media Type message/sipfrag", draft-ietf-sip-sipfrag-00 (work in progress), September 2002.

[8] Olson, S., "A Mechanism for Content Indirection in SIP Messages", draft-ietf-sip-content-indirect-mech-01 (work in progress), August 2002.

[9] Freed, N., "Definition of the URL MIME External-Body Access-Type", RFC 2017, November 1996.

Author's Address

Jon Peterson
NeuStar, Inc.
1800 Sutter St
Suite 570
Concord, CA 94520
US

Phone: +1 925/363-8720
EMail: jon.peterson@neustar.biz
URI: http://www.neustar.biz/

Appendix A. Acknowledgments

The authors would like to thank Eric Rescorla, Rohan Mahy, Robert Sparks, Jonathan Rosenberg, Mark Watson and Patrik Faltstrom for their comments. Cullen Jennings assisted greatly in the development of the content indirection mechanism considered in Section 4.3.

Appendix B. Changelog

Changes from draft-peterson-sip-identity-01:

- Split off child draft-ietf-sip-authid-body-00 for defining of the AIB

- Clarified scope in introduction

- Removed a lot of text that was redundant with RFC3261 (especially about authentication practices)

- Added mention of content indirection mechanism for adding token to requests and responses

- Improved Security Considerations (added piece about credential strength)

Changes from draft-peterson-sip-identity-00:

- Added a section on authenticated identities in responses

- Removed hostname convention for authentication services

- Added text about using 'message/sip' or 'message/sipfrag' in authenticated identity bodies, also RECOMMENDED a few more headers in sipfrags to increase reference integrity

- Various other editorial corrections

Full Copyright Statement

Acknowledgement

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