Network Working Group
Request for Comments: 4484
Category: Informational

J. Peterson
NeuStar
J. Polk
Cisco
D. Sicker
CU Boulder
H. Tschofenig
Siemens

August 2006

Trait-Based Authorization Requirements for the Session Initiation Protocol (SIP)

Status of This Memo

Copyright Notice

Abstract

This document lays out a set of requirements related to trait-based authorization for the Session Initiation Protocol (SIP). While some authentication mechanisms are described in the base SIP specification, trait-based authorization provides information used to make policy decisions based on the attributes of a participant in a session. This approach provides a richer framework for authorization, as well as allows greater privacy for users of an identity system.

Table of Contents

1. Introduction

This document explores requirements of the Session Initiation Protocol (SIP) [1] for enabling trait-based authorization. This effort stems from the recognition that when SIP requests are received by a User Agent Server (UAS), there are authorization requirements that are orthogonal to ascertaining of the identity of the User Agent Client (UAC). Supplemental authorization information might allow the UAS to implement non-identity-based policies that depend on further attributes of the principal that originated a SIP request.

For example, in traditional SIP authorization architectures, the mere fact that a UAC has been authenticated by a UAS doesn’t mean that the UAS will grant the UAC full access to its services or capabilities -- in most instances, a UAS will compare the authenticated identity of the UAC to some set of users that are permitted to make particular requests (as a way of making an authorization decision). However, in large communities of users with few preexisting relationships (such as federations of discrete service providers), it is unlikely that the authenticated identity of a UAC alone will give a UAS sufficient information to decide how to handle a given request.

Trait-based authorization entails an assertion by an authorization service of attributes associated with an identity. An assertion is a sort of document consisting of a set of these attributes that are wrapped within a digital signature provided by the party that generates the assertion (the operator of the authorization service). These attributes describe the ’trait’ or ’traits’ of the identity in question -- facts about the principal corresponding to that identity. For example, a given principal might be a faculty member at a

university. An assertion for that principal's identity might state that they have the 'trait' of 'is a faculty member', and the assertion would be issued (and signed) by a university. When a UAS receives a request with this trait assertion, if it trusts the signing university, it can make an authorization decision based on whether or not faculty members are permitted to make the request in question, rather than just looking at the identity of the UAC and trying to discern whether or not they are a faculty member through some external means. Thus, these assertions allow a UAS to authorize a SIP request without having to store or access attributes associated with the identity of the UAC itself. Even complex authorization decisions based the presence of multiple disjointed attributes are feasible; for example, a 'faculty' member could be part of the 'chemistry' department, and both of these traits could be used to make authorization decisions in a given federation.

It is easy to see how traits can be used in a single administrative domain, for example, a single university, where all users are managed under the same administration. In order for traits to have a broader usage for services like SIP, which commonly are not bounded by administrative domains, domains that participate in a common authorization scheme must federate with one another. The concept of federation is integral to any trait-based authorization scheme. Domains that federate with one another agree on the syntax and semantics of traits -- without this consensus, trait-based authorization schemes would only be useful in an intradomain context. A federation is defined as a set of administrative domains that implement common policies regarding the use and applicability of traits for authorization decisions. Federation necessarily implies a trust relationship, and usual implies some sort of pre-shared keys or other means of cryptographic assurance that a particular assertion was generated by an authorization service that participates in the federation.

In fact, when trait-based authorization is used, an assertion of attributes can be presented to a UAS instead of the identity of user of the UAC. In many cases, a UAS has no need to know who, exactly, has made a request -- knowing the identity is only a means to the end of matching that identity to policies that actually depend on traits independent of identity. This fact allows trait-based authorization to offer a very compelling privacy and anonymity solution. Identity becomes one more attribute of an assertion that may or may not be disclosed to various destinations.

Trait-based authorization for SIP depends on authorization services that are trusted by both the UAC and the UAS that wish to share a session. For that reason, the authorization services described in this document are most applicable to clients either in a single

domain or in federated domains that have agreed to trust one another's authorization services. This could be common in academic environments, or business partnerships that wish to share attributes of principals with one another. Some trait-based authorization architectures have been proposed to provide single sign-on services across multiple providers.

Although trait-based identity offers an alternative to traditional identity architectures, this effort should be considered complementary to the end-to-end cryptographic SIP identity effort [3]. An authentication service might also act as an authorization service, generating some sort of trait assertion token instead of an authenticated identity body.

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 RFC 2119 [2] and indicate requirement levels for compliant SIP implementations.

3. Trait-Based Authorization Framework

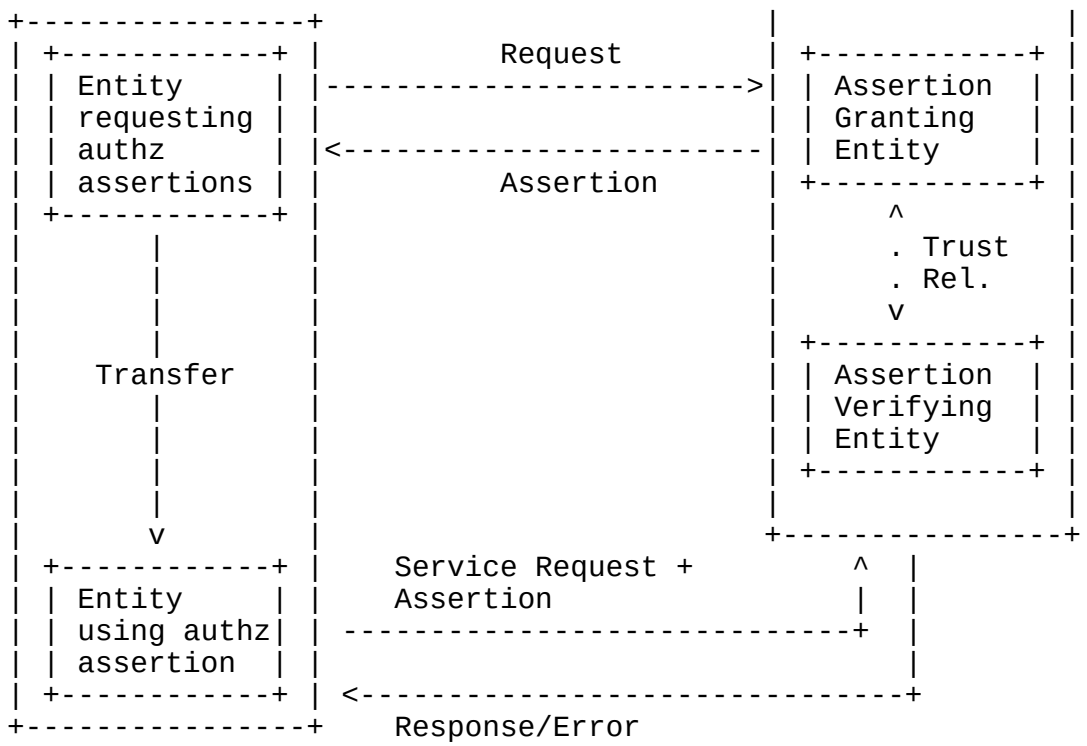
A trait-based authorization architecture entails the existence of an authorization service. Devices must send requests to an authorization service in order to receive an assertion that can be used in the context of a given network request. Different network request types will often necessitate different or additional attributes in assertions from the authorization service.

For the purposes of SIP, SIP requests might be supplied to an authorization service to provide the basis for an assertion. It could be the case that a user agent will take a particular SIP request, such as an INVITE, for which it wishes to acquire an assertion and forward this to the authorization service (in a manner similar to the way that an authenticated identity body is requested in [3]). User agents might also use a separate protocol to request an assertion. In either case, the client will need to authenticate itself to an authorization service before it receives an assertion. This authentication could use any of the standard mechanisms described in RFC 3261 [1], or use some other means of authentication.

Once a SIP UA has an assertion, it will need some way to carry an assertion within in a SIP request. It's possible that this assertion could be provided by reference or by value. For example, a SIP UA could include a MIME body within a SIP request that contains the assertion; this would be inclusion by value. Alternatively, content

indirection [4], or some new header, could be used to provide a URI (perhaps an HTTP URL) where interested parties could acquire the assertion; this is inclusion by reference.

The basic model is as follows:



The entity requesting authorization assertions (or the entity that gets some assertions granted) and the entity using these authorization assertions might be co-located in the same host or domain, or they might be entities in different domains that share a federate with one another. The same is true for the entity that grants these assertions to a particular entity and the entity that verifies these assertions.

From a protocol point of view, it is worth noting that the process of obtaining some assertions might happen some time before the usage of these assertions. Furthermore, different protocols might be used and the assertions may have a lifetime that might allow that these assertions are presented to the verifying entity multiple times (during the lifetime of the assertion).

Some important design decisions are associated with carrying assertions in a SIP request. If an assertion is carried by value, or uses a MIME-based content indirection system, then proxy servers will

be unable to inspect the assertion themselves. If the assertion were referenced in a header, however, it might be possible for the proxy to acquire and inspect the assertion itself. There are certainly architectures in which it would be meaningful for proxy servers to apply admissions controls based on assertions.

It is also the case that carrying assertions by reference allows versatile access controls to be applied to the assertion itself. For instance, an HTTP URL where an assertion could be acquired could indicate a web server that challenged requests, and only allowed certain authorized sources to inspect the assertion, or that provided different versions of the assertion depending on who is asking. When a SIP UA initiates a request with privacy controls [5], a web server might provide only trait information ('faculty', 'student', or 'staff') to most queries, but provide more detailed information, including the identity of the originator of the SIP request, to certain privileged askers. The end-users that make requests should have some way to inform authorization services of the attributes that should be shared with particular destinations.

Assertions themselves might be scoped to a particular SIP transaction or SIP dialog, or they might have a longer lifetime. The recipient of an assertion associated with a SIP request needs to have some way to verify that the authorization service intended that this assertion could be used for the request in question. However, the format of assertions is not specified by these requirements.

Trait assertions for responses to SIP requests are outside the scope of these requirements; it is not clear if there is any need for the recipient of a request to provide authorization data to the requestor.

Trait-based authorization has significant applicability to SIP. There are numerous instances in which it is valuable to assert particular facts about a principal other than the principal's identity to aid the recipient of a request in making an authorization policy decision. For example, a telephony service provider might assert that a particular user is a 'customer' as a trait. An emergency services network might indicate that a particular user has a privileged status as a caller.

4. Example Use Cases

The following use cases are by no means exhaustive, but provide a few high-level examples of the sorts of services that trait-based authorization might provide. All of the cases below consider interdomain usage of authorization assertions.

4.1. Settlement for Services

When endpoints in two domains share real-time communications services, sometimes there is a need for the domains to exchange accounting and settlement information in real-time. The operators of valuable resources (for example, Public Switched Telephone Network (PSTN) trunking, conference bridges, or the like) in the called domain may wish to settle with the calling domain (either with the operators of the domain or a particular user), and some accounting operations might need to complete before a call is terminated. For example, a caller in one domain might want to access a conference bridge in another domain, and the called domain might wish to settle for the usage of the bridge with the calling domain. Or in a wireless context, a roaming user might want to use services in a visited network, and the visited network might need to understand how to settle with the user's home network for these services.

Assuming that the calling domain constitutes some sort of commercial service capable of exchanging accounting information, the called domain may want to verify that the remote user has a billable account in good standing before allowing a remote user access to valuable resources. Moreover, the called domain may need to discover the network address of an accounting server and some basic information about how to settle with it.

An authorization assertion created by the calling domain could provide the called domain with an assurance that a user's account can settle for a particular service. In some cases, no further information may be required to process a transaction, but if more specific accounting data is needed, traits could also communicate the network address of an accounting server, the settlement protocol that should be used, and so on.

4.2. Associating Gateways with Providers

Imagine a case where a particular telephone service provider has deployed numerous PSTN-SIP gateways. When calls come in from the PSTN, they are eventually proxied to various SIP user agents. Each SIP user agent server is interested to know the identity of the PSTN caller, of course, which could be given within SIP messages in any number of ways (in SIP headers, bodies, or what have you). However,

in order for the recipient to be able to trust the identity (in this instance, the calling party's telephone number) stated in the call, they must first trust that the call originated from the gateway and that the gateway is operated by a known (and trusted) provider.

There are a number of ways that a service provider might try to address this problem. One possibility would be routing all calls from gateways through a recognizable 'edge' proxy server (say, 'sip.example.com'). Accordingly, any SIP entity that received a request via the edge proxy server (assuming the use of hop-by-hop mutual cryptographic authentication) would know the service provider from whom the call originated. However, it is possible that requests from the originating service provider's edge proxy might be proxied again before reaching the destination user agent server, and thus in many cases the originating service provider's identity would be known only transitively. Moreover, in many architectures requests that did not originate from PSTN gateways could be sent through the edge proxy server. In the end analysis, the recipient of the request is less interested in knowing which carrier the request came from than in knowing that the request came from a gateway.

Another possible solution is to issue certificates to every gateway corresponding to the hostname of the gateway ('gateway1.example.com'). Gateways could therefore sign SIP requests directly, and this property could be preserved end-to-end. But depending on the public key infrastructure, this could, however, become costly for large numbers of gateways, and moreover a user agent server that receives the request has no direct assurance from a typical certificate that the host is in fact a gateway just because it happens to be named 'gateway1'.

Trait-based authorization would enable the trait 'is a gateway' to be associated with an assertion that is generated by the service provider (i.e., signed by 'example.com'). Since these assertions would travel end-to-end from the originating service provider to the destination user agent server, SIP requests that carry them can pass through any number of intermediaries without discarding cryptographic authentication information. This mechanism also does not rely on hostname conventions to identify what constitutes a gateway and what does not -- it relies on an explicit and unambiguous attribute in an assertion.

4.3. Permissions on Constrained Resources

Consider a scenario wherein two universities are making use of a videoconferencing service over a constrained-bandwidth resource. Both universities would like to enforce policies that determine how this constrained bandwidth will be allocated to members of their

respective communities. For example, faculty members might have privileges to establish videoconferences during the day, while students might not. Faculty might also be able to add students to a particular videoconference dynamically, or otherwise moderate the content or attendance of the conference, whereas students might participate only more passively.

Trait-based authorization is ideal for managing authorization decisions that are predicated on membership in a group. Rather than basing access on individual users, levels (or roles) could be assigned that would be honored by both universities, since they both participate in the same federation.

If the federation honored the traits "faculty", "staff", and "student", they could be leveraged to ensure appropriate use of the network resource between universities participating in the federation. An assertion would then be attached to every request to establish a session that indicated the role of the requestor. Only if the requestor has the appropriate trait would the session request be granted. Ideally, these policies would be enforced by intermediaries (SIP proxy servers) that are capable of inspecting and verifying the assertions.

4.4. Managing Priority and Precedence

There is a significant amount of interest in the Internet telephony community in assigning certain calls a ’priority’ based on the identity of the user, with the presumption that prioritized calls will be granted preferential treatment when network resources are scarce. Different domains might have different criteria for assigning priority, and it is unlikely that a domain would correlate the identity of a non-local user with the need for priority, even in situations where domains would like to respect one another’s prioritization policies.

Existing proposals have focused largely on adding a new header field to SIP that might carry a priority indicator. This use case does not challenge this strategy, but merely shows by way of example how this requirement might be met with a trait-based authorization system. As such, the limitations of the header field approach will not be contrasted here with a hypothetical trait-based system.

An assertion created by a domain for a particular request might have an associated ’priority’ attribute. Recipients of the request could inspect and verify the signature associated with the assertion to determine which domain had authenticated the user and made the

priority assessment. If the assertion's creator is trusted by the evaluator, the given priority could be factored into any relevant request processing.

4.5. Linking Different Protocols

Cryptographic computations are expensive and computing authorization decisions might require a lot of time and multiple messages between the entity enforcing the decisions and the entity computing the authorization decision. Particularly in a mobile environment these entities are physically separated -- or not even in the same administrative domain. Accordingly, the notion of "single sign-on" is another potential application of authorization assertions and trait-based authorization -- a user is authenticated and authorized through one protocol, and can reuse the resulting authorization assertion in other, potential unrelated protocol exchanges.

For example, in some environments it is useful to make the authorization decision for a "high-level" service (such as a voice call). The authorization for the "voice call" itself might include authorization for SIP signaling and also for lower-level network functions, for example, a quality-of-service (QoS) reservation to improve the performance of real-time media sessions established by SIP. Since the SIP signaling protocol and the QoS reservation protocol are totally separate, it is necessary to link the authorization decisions of the two protocols. The authorization decision might be valid for a number of different protocol exchanges, for different protocols and for a certain duration or some other attributes.

To enable this mechanism as part of the initial authorization step, an authorization assertion is returned to the end host of the SIP UAC (cryptographically protected). If QoS is necessary, the end host might reuse the returned assertion in the QoS signaling protocol. Any domains in the federation that would honor the assertion generated to authorize the SIP signaling would similarly honor the use of the assertion in the context of QoS. Upon the initial generation of the assertion by an authorization server, traits could be added that specify the desired level of quality that should be granted to the media associated with a SIP session.

5. Trait-Based Authorization Requirements

The following are the constraints and requirements for trait-based authorization in SIP:

1. The mechanism MUST support a way for SIP user agents to embed an authorization assertion in SIP requests. Assertions can be carried either by reference or by value.

2. The mechanism MUST allow SIP UACs to deliver to an authorization service those SIP requests that need to carry an assertion. The mechanism SHOULD also provide a way for SIP intermediaries to recognize that an assertion will be needed, and either forward requests to an authorization service themselves or notify the UAC of the need to do so.

3. Authorization services MUST be capable of delivering an assertion to a SIP UAC, either by reference or by value. It MAY also be possible for an authorization service to add assertions to requests itself, if the user profile permits this (for example, through the use of content-indirection as described in [4]).

4. Authorization services MUST have a way to authenticate a SIP UAC.

5. The assertions generated by authorization services MUST be capable of providing a set of values for a particular trait that a principal is entitled to claim.

6. The mechanism MUST provide a way for authorized SIP intermediaries (e.g., authorized proxy servers) to inspect assertions.

7. The mechanism MUST have a single baseline mandatory-to-implement authorization assertion scheme. The mechanism MUST also allow support of other assertion schemes, which would be optional to implement. One example of an assertion scheme is Security Assertion Markup Language (SAML) [6] and another is RFC 3281 X.509 Attribute Certificates [7].

8. The mechanism MUST ensure reference integrity between a SIP request and assertion. Reference integrity refers to the relationship between a SIP message and the assertion authorizing the message. For example, a reference integrity check would compare the sender of the message (as expressed in the SIP request, for example, in the "From" header field value) with the identity provided by the assertion. Reference integrity is necessary to prevent various sorts of relay and impersonation

attacks. Note that reference integrity MAY apply on a per-message, per-transaction, or per-dialog basis.

9. Assertion schemes used for this mechanism MUST be capable of asserting attributes and/or traits associated with the identity of the principal originating a SIP request. No specific traits or attributes are required by this specification.

10. The mechanism MUST support a means for end-users to specify policies to an authorization service for the distribution of their traits and/or attributes to various destinations.

11. The mechanism MUST provide a way of preventing unauthorized parties (either intermediaries or endpoints) from viewing the contents of assertions.

12. Assertion schemes MUST provide a way of selectively sharing the traits and/or attributes of the principal in question. In other words, it must be possible to show only some of the attributes of a given principal to particular recipients, based on the cryptographically- assured identity of the recipient.

13. It MUST be possible to provide an assertion that contains no identity -- that is, to present only attributes or traits of the principal making a request, rather than the identity of the principal.

14. The manner in which an assertion is distributed MUST permit cryptographic authentication and integrity properties to be applied to the assertion by the authorization service.

15. It MUST be possible for a UAS or proxy server to reject a request that lacks a present and valid authorization assertion, and to inform the sending UAC that it must acquire such an assertion in order to complete the request.

16. The recipient of a request containing an assertion MUST be able to ascertain which authorization service generated the assertion.

17. It MUST be possible for a UAS or proxy server to reject a request containing an assertion that does not provide any attributes or traits that are known to the recipient or that are relevant to the request in question.

18. It SHOULD be possible for a UAC to attach multiple assertions to a single SIP request, in cases where multiple authorization services must provide assertions in order for a request to complete.

6. Security Considerations

The subject of this document is an authorization system for SIP that is not predicated on the distribution of end-users' identities, but rather shares traits of the users. As such, the bulk of this document discusses security.

The distribution of authorization assertions requires numerous security properties. An authorization service must be able to sign assertions, or provide some similar cryptographic assurance that can provide non-repudiation for assertions. These requirements are further detailed in Section 3.

7. Acknowledgements

The authors thank Christopher Eagan and Mary Barnes for their valuable input.

8. References

8.1. 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", BCP 14, RFC 2119, March 1997.

8.2. Informative References

[3] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, August 2006.

[4] Burger, E., Ed., "A Mechanism for Content Indirection in Session Initiation Protocol (SIP) Messages", RFC 4483, May 2006.

[5] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", RFC 3323, November 2002.

[6] Organization for the Advancement of Structured Industry Standards, "Security Assertion Markup Language v1.0", November 2002, <http://www.oasis-open.org>.

[7] Farrell, S. and R. Housley, "An Internet Attribute Certificate Profile for Authorization", RFC 3281, April 2002.

Authors' Addresses

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/

James M. Polk
Cisco Systems
2200 East President George Bush Turnpike
Suite 570
Richardson, TX 75802
US

EMail: jmpolk@cisco.com

Douglas C. Sicker
University of Colorado at Boulder
ECOT 531
Boulder, CO 80309
US

EMail: douglas.sicker@colorado.edu

Hannes Tschofenig
Siemens AG
Otto-Hahn-Ring 6
Munich 81739
Germany

EMail: Hannes.Tschofenig@siemens.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).