Third Party Authorization in the Session Initiation Protocol
draft-lawrence-sip-3rd-party-authorization-00

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on November 4, 2009.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

This draft describes some circumstances that are common in SIP deployments which lack a rigorous authorization model, and points out some ways in which this has resulted in poor security
characteristics.

The purpose of this document is to stimulate discussion of the identified problem and proposed requirements for any solution.

Comments are solicited, and should be directed to the DISPATCH working group list at 'dispatch@ietf.org'.

Table of Contents

1. Introduction ....................................................... 3
2. Problem Description .............................................. 3
   2.1. Authentication vs Authorization ............................ 3
   2.2. The Locality of Authority and Enforcement ................. 4
      2.2.1. User Agent Based Authorization Scales Poorly .......... 5
      2.2.2. Request Delivery as an Authorization Indication ...... 6
   2.3. Examples of Distributed Authorization and Enforcement
      in SIP ....................................................... 7
      2.3.1. Dialog Event Subscriptions ............................ 8
      2.3.2. INVITE at a PSTN Gateway .............................. 8
      2.3.3. INVITE with Join .................................... 9
3. Candidate Solution Requirements ................................. 9
   3.1. Authorizing Third Party Requirements ....................... 10
      3.1.1. Third Party Topologies ............................... 11
   3.2. Authorization and Integrity Protection ..................... 13
4. IANA Considerations .............................................. 14
5. Security Considerations ......................................... 14
6. Acknowledgments ................................................. 14
7. Informative References ......................................... 15
   Author’s Address .............................................. 15
1. Introduction

This draft attempts to make the case that SIP lacks a mechanism needed to provide for the implementation and enforcement of security policy in typical deployments. It's purpose is to present the problem, outline some requirements for a solution, and to pose the question of whether or not is is an area the IETF community considers worth designing solutions for.

This draft does not propose any mechanisms to address the problem it describes; it also does not comprehensively evaluate the relevance of existing SIP security mechanisms (either standardize or proposed) in addressing this problem. Both are possible areas for future work.

2. Problem Description

The Session Initiation Protocol (SIP) [RFC3261], by design, distributes functionality broadly throughout the network - it is often characterised as "pushing intelligence out to the edge". This sometimes means that important security policy enforcement must also be distributed. SIP currently lacks some features that facilitate this distribution of security enforcement.

2.1. Authentication vs Authorization

It is important when considering the problems described in this memo to keep in mind the difference between Authentication and Authorization:

Authentication
- is the process of confirming some identity.

Authorization
- is the process of deciding whether or not something is permitted by a security policy.

Usually an Authorization decision depends upon an Authentication. For example, if Alice tries to call Bob and a proxy between them is deciding whether or not to allow the call, it must first Authenticate that the request is from Alice, and then consult the policy to see whether or not Alice is Authorized to make calls to Bob.

It is an unfortunate historical error that the header name used to carry Authentication information in a SIP (or HTTP) request is named "Authorization", which only increases the confusion between these two concepts.
In the remainder of this document, the modifier "Authorizing" is applied to various entities to designate the party that is making an Authorization decision - consulting a security policy and any relevant information - to determine whether or not a SIP request is permitted by the policy.

2.2. The Locality of Authority and Enforcement

In a distributed system, the ultimate enforcement of a security policy is best implemented as closely as possible to the object of that policy. In less abstract terms: if the goal of a policy is to control access to a resource, then the ultimate enforcement of whether or not to grant access to the resource is best made by an entity local to that resource. Any separation between the resource that is the object of the policy and the entity implementing the enforcement of that policy introduces a gap which an attacker may be able to exploit. Because it is not always possible to colocate the policy decision at the point of enforcement, it is sometimes necessary to have an Authorizing entity in one part of a distributed system that then conveys the appropriate decision to another at which the enforcement action takes place.

In SIP, the optimal enforcement point is often at either the ultimate UAS that is the target of a request, or at an Intermediate Entity whose role includes the enforcement of policy at the edge of an administrative domain. In this document, the term: Intermediate Entity is used to mean any SIP entity that receives SIP requests and sends requests somehow related to those received requests to other targets. The term includes Proxies, Application Servers, Back To Back User Agents, and any other entity that acts as forwarding or translating entity with SIP on both "sides".

However, these systems, and especially User Agents, are often not the appropriate place to actually make the Authorization decision. For various reasons it is sometimes appropriate to make the Authorization decision somewhere other than the point of enforcement, and in these cases at present SIP lacks a standard means to convey the results of that decision.

Most specifications of SIP mechanisms expect that a request will be subjected to an authorization decision by the receiving User Agent Server. If the Alice’s UA sends a request to Bob’s UA, it is Bob’s UA that decides (presumably after authenticating that the request is really from Alice) whether or not Alice should be allowed whatever access the request requires. This model is the correct one in many cases, especially those in which the actual authorization decision
can be made by the human user: if the request is for Bob’s presence information, the UA can let Bob decide whether Alice should be allowed to see his presence and respond appropriately.

Using the user to make the authorization decision is not always appropriate, however - it fits poorly or not at all when:

There is no way to consult a user

Many kinds of UAs either don’t have a "user" to consult (such as a PSTN gateway or a ceiling-mounted paging system speaker), or don’t have a user interface adequate to construct a query (an ATA).

Consulting the user is the wrong thing to do

The nature of the request may be such that consulting the user would be counter to the nature of the request. If the request is that Alice, a call center manager, be silently conferenced in order to monitor a call between a customer and Bob, a service representative under her supervision, then alerting Bob that Alice wishes to listen to the call would defeat the purpose.

Asking the user would be ambiguous

The request might be such that the user either could not understand the question or be such that the user cannot reasonably be expected to understand why permission should be granted. If Bob were asked whether or not Alice should be allowed to receive dialog event notices regarding the call he is on now, he would probably not understand the question and be annoyed at the interruption of his call; he could not tell whether this was in support of a future request to Join the call, a future request that Alice be able to call him when he becomes available, or just in support of lighting the Busy Lamp for his line on Alice’s phone.

2.2.1. User Agent Based Authorization Scales Poorly

Placing the authorization decision in the ultimate User Agent Server suffers from a scaling problem: the size and complexity of the authorization decision scales with the number of possible interactions. While Bob’s UA may be able to authenticate that a particular request came from Alise’s UA, if Alice and Bob are in a large administrative domain, the problem of deciding which users in that domain are authorized to make which requests grows too large to reasonably implement inside many UAs. Even if the problem is just one of storing the appropriate access control lists, it is not reasonable to replicate the lists into all UAs (and perhaps not
desirable from a security perspective, since this decreases the security of the lists themselves).

Even aside from the problem of database sizes that may be required for authorization decisions in the ultimate UAS, the non-uniformity of facilities in UAS makes central control of security policy impractical. If an organization wishes to create a security policy for a particular kind of request, the capabilities offered by UAs are too limited for any but the most primitive of possibilities (and often not even that). For example, one might want to make different rules for whether a particular request be allowed at one time of day but not at another. Since few UAs will have that capability (and since the means of expressing it probably vary even among those that did), an organization that needed such a policy would have to make the authorization decision elsewhere, but other than preventing delivery of the request, there is no standard way to communicate the result of that authorization decision to the UAS.

2.2.2. Request Delivery as an Authorization Indication

Today, authorization control that is not local to the target UAS usually takes the form of delivery control: an Intermediate entity makes an authorization decision and either delivers the request onwards toward the target UAS or rejects it. The fact that the target UAS receives the request from the authorizing entity is the indication that the request has been authorized. There are a number of problems with this as a security policy enforcement architecture:

Passive participation by the recipient

In many cases, the agent that is the recipient of the "authorized" request is not enforcing the requirement that the request be processed by the authorizing entity. It will act on any request it receives, and only the fact that the rest of the well behaved requestors will route through the Authorizing Intermediate allows the security policy to be enforced at all. An attacker who can find a way to route requests such that they do not pass through the Authorizing entity will find the target UAS happy to accept them.

Use of IP addresses as authenticators

The most common form that "delivery as authorization" takes is that the IP address(es) of one or more Authorizing Intermediates is configured into the UAS such that only requests from those Intermediates are to be considered to have been authorized. Most often, the UAS uses the peer address from the UDP or TCP level as the authentication indication. Since techniques for spoofing IP
addresses are well known, this is a very weak basis upon which to build any security. In some particularly weak implementations, the presence of an IP address value in the Via headers has been seen to be used as an authenticator; this does not even merit further discussion.

**TLS peer authentication**

A much stronger but unfortunately less common version of "delivery as authorization" is the use of a mutually authenticated TLS transport between an Intermediate and the point of enforcement. This has the advantage that it is based on the identities in the certificates, so given a usable certificate infrastructure and good implementations, at least the authentication component of the decision is a strong one. This solution is especially well suited to authorizing requests as they cross a boundary between one administrative domain and another (such as between an enterprise and a telephony service provider). It is, however, limited by the fact that it only allows one degree of separation between the Authorizing entity and the Enforcing entity — they may not be more than a single SIP hop apart for this to be a usable authorization indication. It is also worth noting that even in this limited deployment mode, there is no well-defined indication included in the SIP signalling that explicitly communicates that a particular message has been authorized by the sender; the indication is always implicit.

This last problem — the fact that for delivery to work as an authorization indication, the authorizing and enforcing entities must be adjacent — creates another more subtle architectural problem. The requirement for adjacency interferes with a property of SIP that is otherwise one of its great strengths: that a proxy (or other well-behaved Intermediate) need not understand every aspect of the requests and responses it forwards. It is this property that allows User Agents to evolve new capabilities, and, so long as they are able to find other UAs with complementary features, to use those capabilities without the SIP routing fabric between them needing to be upgraded. If there is a security property of some new capability, and authorization decisions appropriate to that new capability are not easily implementable in the originating and terminating UAs themselves, then there is only one place in the SIP routing fabric where the authorization decision can be placed: adjacent to the target UA.

### 2.3. Examples of Distributed Authorization and Enforcement in SIP

The following are examples of SIP operations that currently suffer from poor security properties in many deployments and/or are not well...
supported by User Agents in part because providing appropriate security is problematic. It is by no means an exhaustive list.

2.3.1. Dialog Event Subscriptions

The dialog event package \[RFC4235\]:

allows users to subscribe to another user and to receive notification of the changes in state of INVITE-initiated dialog usages in which the subscribed-to user is involved.

A SUBSCRIBE request \[RFC3265\] for the "dialog" package allows the requesting UA to ask for NOTIFY messages that include the dialog state and identifiers (call-id and tag values) for, potentially, all dialogs at the receiving UA. There are many legitimate purposes for such a request, since they are needed to implement a number of desirable features in a distributed communications system. However, the SUBSCRIBE request does not communicate what feature the request is being made to support, nor can the NOTIFY communicating the dialog information restrict the purposes for which it is used. The dialog package specification includes some recommendations on how to construct security policies in a UAS so as to restrict access to dialog identifiers and other sensitive information, but because leaving out parts of the information would often disable desired features in a distributed system, it is common for implementations to include all the dialog information, and to accept any requests they receive (in effect passively implementing a request delivery authorization policy as described in Section 2.2.2).

\[RFC4538\] defines the SIP Target-Dialog header field

This header field is used in requests that create SIP dialogs. It indicates to the recipient that the sender is aware of an existing dialog with the recipient, either because the sender is on the other side of that dialog, or because it has access to the dialog identifiers. The recipient can then authorize the request based on this awareness.

This authorization by awareness of dialog identifiers is used in many SIP services, so having weak authorization mechanisms in the UASes responsible for those identifiers creates vulnerabilities elsewhere.

2.3.2. INVITE at a PSTN Gateway

One of the most common operations in SIP telephony is the gateway function - receiving a SIP INVITE request and using the information in it to initiate a PSTN call setup request. Because a PSTN call is often a potentially expensive billable event, enforcement of access
control is important. In many (perhaps most) SIP telephony deployments, the actual authorization decision is not implemented in the gateway itself, but in some Intermediate system between the requesting UAC and the gateway; gateways typically implement at most a request delivery authorization policy as described in Section 2.2.2.

2.3.3. INVITE with Join

[RFC3911] defines the Join header field "to logically join an existing SIP dialog with a new SIP dialog"; the specification notes that:

This primitive can be used to enable a variety of features, for example: "Barge-In", answering-machine-style "Message Screening" and "Call Center Monitoring".

The Join mechanism is a good example of a primitive that, depending on the service it is being used to implement, can have very different authorization requirements — but the UAS may not be able to easily distinguish what the service is and therefore know which to requirements to apply. A simple example is an INVITE-with-Join request received by a phone, containing SDP that requests that the joining phone receive media but not provide any to be mixed into the call: this request may have resulted from

- The human User using the receiving UAS may have (through out-of-band means) asked a colleague to listen to the call so as to provide advice or to be available as a witness to what transpired on a sensitive call.

- The request may come from that User’s supervisor in order to determine (as allowed by the management policies that govern their relationship) whether or not the User is conducting the call appropriately.

At the protocol level, these two requests may be indistinguishable, but in the former case it would be appropriate for the UAS to query the user to authorize the Join request (and useful to that User to receive an indication that the monitoring has begun), while in the latter case it would defeat the purpose of the request to do so. In any case, it would clearly be inappropriate for a phone to unconditionally and without notice accept any such request.

3. Candidate Solution Requirements

This draft argues that SIP would be improved by a mechanism that
allowed for a small number of entities in a given deployment to make Authorization decisions with respect to SIP requests according to a locally defined policy, and have a secure means of communicating that a given has been authorized. This would allow other SIP entities (whether Intermediate Systems or User Agent Servers) to make very simple enforcement decisions based on whether or not a request had been authorized by an authority they recognize. The complexity of the security configuration for most SIP entities would be reduced to recognition and validation of a small number of valid authorizors.

This section proposes some requirements that any solution would need to meet in order to address the problem described in Section 2 and be usable and incrementally deployable in existing SIP usage.

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 [RFC2119].

These proposed requirements use the following terms (the definitions of which are admittedly somewhat vague):

Authorization Indication
The information incorporated into a SIP message that indicates that a request is authorized within some security policy. No specific forms are suggested for Authorization Indications in this draft.

Authorizing Third Party
Any implementation that can receive a SIP message generated by some originating UAC, evaluate that request with respect to some security policy, and generate an Authorization Indication to be incorporated into some other SIP message such that a later recipient of that SIP message or one derived from it can be assured that the request is allowed under the security policy.

3.1. Authorizing Third Party Requirements

SIP deployments often contain many different ‘types’ of implementations: some, like User Agents and Proxies, that are well defined by the existing formal specifications, and others that are less rigorously defined but meet practical needs and are essential to functionality required by end users. This reality must be recognized in order for any solution to be deployable in real (rather than just hypothetical architecturally "pure") SIP networks. This implies that the definition of an Authorizing Third Party is very flexible: The following are suggested requirements relating to which implementation roles can act as an Authorizing Third Party:
R1 It MUST be possible for a UAS to act as an Authorizing Third Party, incorporating an Authorization Indication in a response (see Figure 2 and Figure 3).

R2 It MUST be possible for a Proxy to act as an Authorizing Third Party, incorporating an Authorization Indication in requests it forwards (see Figure 1 and Figure 3).

R3 It MUST be possible for other non-Proxy intermediate systems, such as Back-To-Back User Agents to act as an Authorizing Third Party, incorporating an Authorization Indication in requests they forward.

The following are suggested requirements relating to the kinds of SIP messages a Third Party Authentication could be used with:

R4 It MUST be possible to include an Authorization Indication in a REFER request such that when the UA that is the target of the REFER makes a request as directed by the REFER, that new request can include an Authorization Indication (see Figure 4).

R5 It MUST be possible to include an Authorization Indication in both dialog-forming and in-dialog requests.

3.1.1. Third Party Topologies

In real SIP networks, Intermediate entities can appear in unexpected places - a solution that does not recognize that fact will not be useful.

R6 It MUST be possible for an Authorization Indication to be usefully forwarded by an Intermediate system that does not implement the Third Party Authorization mechanisms.

The requirement above does NOT mean that it must be possible to forward any Authorization Indication through any intermediate system at all and always be useful - such a requirement would not be possible to meet. The objective is that at least "well behaved" intermediates have some chance of forwarding; defining "well behaved" in this context will be an important part of evaluating any proposed mechanism.

Figure 1 illustrates a topology in which an Intermediate system adds the authorizing information, changing the original request (r1) into the authorized request (ar):
Figure 1 illustrates a topology in which a Redirecting system includes the authorizing information in a redirect (3xx) response (r1r) to the original request (r1), causing the UAC to generate the authorized request (ar):

```
+--------------+
| Authorizing  |
| Redirector   |
+--------------+

^     |
| r1   |
| r1r  |
| V    |
```

Figure 3 combines elements of the previous two topologies; the Originating UAC sends the original request (r1) to an Intermediate system, which then sends a request (r2) to a Redirecting system (r2r), which then adds the authorizing information to its redirect (3xx) response (r2r), causing the Intermediate system to generate the authorized request (ar). Either the r1 path or the ar path might contain other intermediates.
3.2. Authorization and Integrity Protection

It is tempting to suggest the requirement that it be impossible for an attacker to modify a SIP message containing an Authorization Indication, or to copy an Authorization Indication from one SIP message to another without detection. Unfortunately, it is an integral part of the SIP architecture that SIP messages are modified as they move from one system to another; these modifications are not only unavoidable but essential to the correct operation of the SIP
protocol as currently defined. There are also important use cases (see Figure 4) in which an Authorization Indication is explicitly meant to be copied from one request to another. The following requirements for message integrity protection are suggested:

R7  It MUST be possible for an Authorization Indication to include integrity protection of the parts of a request that it authorizes.

R8  It MUST be possible for an Intermediate system to detect what parts of a request are integrity protected so that the Intermediate system can either avoid changing the protected parts or return an error that indicates that the request cannot be satisfied because of the protection.

R9  It MUST be possible for an Enforcing system to determine what parts of a request are integrity protected by the Authorization Indication and what parts are not.

R10 It MUST be possible for an Enforcing system to determine the identity of the party that provided the Authorization Indication as distinct from the Originating UAC.

4. IANA Considerations

None until a solution is described.

5. Security Considerations

This memo attempts to make the case that the current definition and deployment of the Session Initiation Protocol is missing a mechanism important to appropriately secured operation in at least many of the environments in which it is used. Section 3 attempts to bound a solution space, and as such should be read carefully to determine whether or not appropriate goals have been described.

6. Acknowledgments

The author would like to thank Hadriel Kaplan, Michael Procter, Mary Barnes, and Marek Dutkiewicz for invaluable feedback on pre-publication drafts of this document.
7. Informative References


Author’s Address

Scott Lawrence
Nortel Networks, Inc.
600 Technology Park
Billerica, MA 01821
USA

Phone: +1 978 288 5508
Email: scott.lawrence@nortel.com