Session Initiation Proposal
Investigation Working Group
Internet-Draft
Expires: November 15, 2004

V. Hilt
Bell Labs/Lucent Technologies
G. Camarillo
Ericsson
J. Rosenberg
dynamicsoft
May 17, 2004

Session-Independent Policies for the Session Initiation Protocol (SIP)
draft-hilt-sipping-session-indep-policy-01

Status of this Memo

Copyright Notice

Abstract

Session policies are often independent of a specific session and generally apply to sessions during a certain period of time. This draft defines a document format for session-independent session policies. It also discusses the use of policy documents with the user agent profile delivery framework.

Table of Contents

1. Introduction

Some domains have policies in place, which impact the sessions established using the Session Initiation Protocol (SIP). These policies are typically needed to support the operation of the network infrastructure or certain services. For example, a SIP user agent might be located in a domain that is behind a Network Address Translator (NAT). This domain might have a policy in place that requires the user agent to contact a TURN [10] relay before setting up a session. Information about this policy is essential for a user agent to successfully set up a session.

In another example, SIP is used in a wireless network. The network provider has limited resources for media traffic. During periods of high activity, the provider would like to restrict codec usage on the network to lower rate codecs. In existing approaches, this is frequently accomplished by having the proxies examine the SDP [2] in the body and remove the higher rate codecs or reject the call and require the UA to start over with a different set of codecs. Having information about the current policy would enable user agents to initiate a session with an acceptable codec.

In a third example, a domain has established policies regarding the type of user agents that can use their network. For example, a domain could require that user agents using its network use a particular protocol (e.g., SIP) with a set of extensions (e.g., preconditions must be used). A user agent needs to know the exact policy of a domain in order to be able to use the right configuration to send and receive traffic in that domain.

Some domains have policies in place that are enforced by network elements. For example, a domain might have a configuration in which all packets containing a certain voice encoding are dropped. Unfortunately, enforcement mechanisms usually do not inform the user about the policies they are enforcing and silently keep the user from doing anything against them. This may lead to the malfunctioning of devices that is in-apprehensible to the user. With session policies, the user could decide to switch to a different codec or connect to a domain with less stringent policies.

Session policies may be specific to a certain session and may change from session to session. Such policies can be set up using the framework for session-specific policies [3]. Other session policies

remain in place for a longer period of time, typically in the range of hours or days. In principle, these policies could also be set up on a session-to-session basis. However, establishing the same policies over and over again is expensive, causing the continuous transmission of the same information during session setup, and

possibly adding to session setup latencies. It is therefore desirable to enable user agents to obtain the policies relevant for them and to inform the user agents about changes in these policies.

Our solution for supporting session-independent session policies is to enable user agents to retrieve policies, for example, as part of their device configuration. We define a document format for SIP session policies. SIP session policy documents can be transmitted using RFC3265 [9] and the Framework for SIP User Agent Profile Delivery [8]. We discuss the use of this framework for SIP session policies. However, session policy documents can also conveyed to user agents using other mechanisms.

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 BCP 14, [1] and indicate requirement levels for compliant implementations.

3. User Agent Profile Delivery Framework

One way of conveying session policy documents as defined in Section 5 to a user agent is by using the Framework for SIP User Agent Profile Delivery [8]. The following sections describe the use of this framework.

3.1 Use of URIs for Policy Subscriptions

Session-independent policies are frequently provided by the home domains of a registered user (i.e. the domain of an address-of-record) and the local domain (i.e. the domain the user agent is currently connected to). Policies of the local domain may be specific to a certain user (i.e. address-of-record) or apply generally to all user agents in the network.

The home domain is responsible for providing SIP service to a user. This domain will frequently maintain user preferences and subscriptions to services and may provide session-independent

policies, that are needed to implement them. The local domain is responsible for providing IP service to a user agent. It may be the same domain as the home domain or a different domain in case the user is roaming in a foreign network or obtains SIP services and IP connectivity from different providers. The local domain often provides policies, which impact the network traffic created by a certain user or all devices in general.

The following three types of policy URIs are typically of interest for a user agent:

- "device" policies from the local network that apply to all user agents,
- "user" policies from the local network for a certain address-of-record,
- "user" policies from the home domain of an address-of-record.

The only way to find out if a domain provides policies to a user agent is to subscribe to the respective policy URI. It is therefore RECOMMENDED that a user agent subscribes to all of the above policy URI types. The subscription will be rejected if the respective domain does not have policies in place. The creation of these URIs is defined in [8].

3.2 Support of Policy Formats

This specification defines a document format for session policies with the MIME type "application/basic-session-policy+xml". A user agent which is conform with this specification MUST indicate its support for this document format in the Accept header of a SUBSCRIBE.

Some session policies are required by the network and sessions can't be established without using them. An example is a policy enabling NAT traversal. Other policies are optional. They are often used by services or needed to improve the quality of service in the network. Session can be established without using them but may lack the respective service or be of a lower quality.

If the subscriber does not indicate its support for the MIME type used by a policy that is mandatory in the Accept header of a SUBSCRIBE request, the notifier MUST reject this request with a 406 "Not Acceptable" response. This way, the subscriber knows that there are mandatory policies it does not support, which will cause a session setup attempt to fail.

4. Policy Profile Considerations

This specification defines an initial profile for session policies using the framework for user agent profile delivery. Other profiles for different session policies might be defined in additional

specifications. The following considerations may serve as guidelines when developing additional session policy profiles.

A policy document encodes one or more session policies. Each policy package must specify or cite detailed specifications for the syntax and semantics associated with the format of such documents.

The policy package should support versioning so that the recipients of policy document can properly order them. This may be achieved using a version attribute.

Policy documents often have an expiration time. After this time, the policies encoded in the document will not be used any more. A policy document may contain an expiration attribute.

A policy document may contain multiple policies. Each policy in the document may have a different scope. For example, a policy for firewall traversal would only apply to external calls whereas a policy limiting the bandwidth available could be in effect during peak hours. A policy document may define a scope attribute that specifies to which sessions a certain policy applies. Possible scopes are:

- Time and day: limits the use of a policy to certain times or days.
- Local entity: limits the use of a policy to a specific to a certain local user. This is in particular useful for devices that supports multiple identities.
- Remote entity: limits the use of a policy to sessions involving certain remote addresses, for example all non-local addresses.
- Media streams: limits the use of a policy to certain media streams.

The use of policies may be mandatory or optional. A policy document may specify whether a policy is mandatory or optional.

5. Session Policy Profile Format

A session policy document is an XML document that MUST be well-formed and SHOULD be valid. Policy documents MUST be based on XML 1.0 and MUST be encoded using UTF-8. This specification makes use of XML namespaces for identifying session policy documents. The namespace URI for elements defined by this specification is a URN [5], using the namespace identifier 'ietf' defined by RFC 2648 [6] and extended by [4]. This URN is:

urn:ietf:params:xml:ns:sessionpolicy

A session policy document begins with the root element tag "sessionpolicy".

5.1 Policy Document Format

A session policy document starts with a sessionpolicy element. This element has three mandatory attributes:

version: This attribute allows the recipient of session policy information documents to properly order them. Versions start at 0, and increment by one for each new document sent to a subscriber. Versions are scoped within a subscription. Versions MUST be representable using a 32 bit integer.
domain: This attribute contains the domain the policy belongs to.
entity: This attribute contains a URI that identifies the user whose policy information is reported in the remainder of the document.

The sessionpolicy element has a series of sessionpolicy sub-elements: zero or one protocols element and zero or one media element.

5.1.1 Protocols Element

The protocols element contains a series of protocol sub-elements. Each protocol sub-element contains the policy related to the usage of a particular protocol.

The protocol element has a single mandatory attribute, name. The name attribute identifies a protocol the policy of each protocol element is referring to. The protocol element has a series of sub-elements: methods, option-tags, feature-tags, and bodies.

5.1.1.1 Methods Element

The methods element contains a default-policy attribute and method elements. The default-policy attribute contains the policy for methods that are not listed as method elements. A method element has two attributes: name and policy. The name attribute identifies a method, and the policy attribute contains the policy for that method (allowed or disallowed).

5.1.1.2 Option-tags Element

The option-tags element contains a default-policy attribute and option-tag elements. The default-policy attribute contains the policy for option-tags that are not listed as option-tag elements. An option-tag element has two attributes: name and policy. The name attribute identifies a method, and the policy attribute contains the

policy for that method (mandatory, allowed, or disallowed).

5.1.1.3 Feature-tags Element

The feature-tags element contains a default-policy attribute and feature-tag elements. The default-policy attribute contains the policy for feature-tags that are not listed as feature-tag elements. An feature-tag element has two attributes: name and policy. The name

attribute identifies a method, and the policy attribute contains the policy for that method (allowed, or disallowed).

5.1.1.4 Bodies Element

The bodies element contains a default-policy attribute, a default-encryption attribute and body-disposition elements. The default-policy attribute contains the policy for body dispositions that are not listed as body-disposition elements. The default-encryption attribute contains the encryption policy for body dispositions that are not listed as body-disposition elements.

A body-disposition element can have a number of attributes: name, policy, default-policy, and encryption. The name attribute identifies a body-disposition, and the policy attribute contains the policy for that body-disposition (allowed, or disallowed). The default-policy attribute contains the policy for body formats that are not listed as body-format elements. The encryption attribute indicates whether or not encryption is allowed for a particular body disposition.

A body-disposition element contains body-format elements. A body-format element can have a two attributes: name and policy. The name attribute identifies a body-format, and the policy attribute contains the policy for that body-format (allowed or disallowed).

5.1.1.5 Extensibility

Other elements from different namespaces MAY be present within a protocol element for the purposes of extensibility; elements or attributes from unknown namespaces MUST be ignored.

5.1.1.6 Example of a Protocol Element

```
   <protocols>
     <protocol name="SIP">
       <methods default-policy="allowed">
          <method name="MESSAGE" policy="disallowed"/>
       </methods>
       <option-tags default-policy="disallowed">
          <option-tag name="100rel" policy="mandatory"/>
          <option-tag name="preconditions" policy="allowed"/>
       </option-tags>
       <feature-tags default-policy="disallowed">
          <feature-tag name="video" policy="allowed"/>
       </feature-tags>
       <bodies default-policy="allowed" default-encryption="allowed">
          <body-disposition name="session" policy="allowed"
                            encryption="disallowed" default-
policy="disallowed">
             <body-format name="application/sdp" policy="allowed"/>
          </body-disposition>
       </bodies>
     </protocol>
   </protocols>
```

5.1.2 Media Element

The media element contains the policy related to the characteristics of media streams of different types. It has three attributes: maxbandwidth, maxnostreams, and default-policy. They contain the maximum bandwidth the user can count on, the maximum number of media streams that the user is allowed to established at the same time, and the default policy (allowed or disallowed) for stream types that are not listed as stream elements.

The media element contains a series of stream elements.

5.1.2.1 Stream Element

A stream element can have a number of attributes: type, policy, maxbandwidth, and maxnostreams. The type attribute identifies a media type, and the policy attribute contains the policy for that media type (allowed or disallowed).

The stream element has a number of optional sub-element: the codecs element, the transports element and the directions element.

5.1.2.1.1 Codecs Element

The codecs element contains a default-policy attribute and codec elements. The default-policy attribute contains the policy for codecs

that are not listed as codec elements. A codec element can have two attributes: name and policy. The name attribute identifies a codec name, and the policy attribute contains the policy for that codec (allowed, or disallowed). The codec name is the encoding name as defined by the respective RTP profile.

5.1.2.1.2 Transports Element

The transports element contains a default-policy attribute and transport elements. The default-policy attribute contains the policy for transports that are not listed as transport elements. A transport element can have two attributes: name and policy. The name attribute identifies a transport, and the policy attribute contains the policy for that transport (allowed, or disallowed).

5.1.2.1.3 Directions Element

The directions element contains a default-policy attribute and direction elements. The default-policy attribute contains the policy for directions that are not listed as direction elements. A direction element can have two attributes: name and policy. The name attribute identifies a direction (sendrecv, sendonly, recvonly), and the policy attribute contains the policy for that direction (allowed, or disallowed).

5.1.2.1.4 Extensibility

Other elements from different namespaces MAY be present within a stream element for the purposes of extensibility; elements or attributes from unknown namespaces MUST be ignored.

5.1.2.2 Example of a Media Element

```
<media maxnostreams="4" default-policy="disallowed">
   <stream type="audio" policy="allowed">
        <codecs default-policy="allowed">
            <codec name="PCMU" policy="disallowed"/>
            <codec name="PCMA" policy="disallowed"/>
        </codecs>
```

```
          <transports default-policy="disallowed">
              <transport name="RTP/AVP" policy="allowed"/>
          </transports>
          <directions default-policy="disallowed">
              <direction name="sendonly" policy="allowed"/>
          </directions>
      </stream>
   </media>
```

5.2 Schema

The following is the schema for the application/session-policy+xml type:

```
<?xml version="1.0" encoding="UTF-8"?>
TBD
```

5.3 Example

The following is is an example of an application/session-policy+xml document:

```
<?xml version="1.0" encoding="UTF-8"?>
<sessionpolicy xmlns="urn:ietf:params:xml:ns:sessionpolicy"
               version="0"
               domain="example.com"
               entity="sip:alice@example.com">
 <protocols>
  <protocol name="SIP">
    <methods default-policy="allowed"/>
    <option-tags default-policy="allowed"/>
    <feature-tags default-policy="allowed"/>
    <bodies default-policy="allowed" default-encryption="allowed"/>
  </protocol>
 </protocols>
 <media default-policy="allowed"/>
</sessionpolicy>
```

6. Security Considerations

Session policy information can be sensitive information. The protocol used to distribute it SHOULD ensure privacy, message integrity and authentication. Furthermore, the protocol SHOULD provide access controls which restrict who can see who else's session policy information.

7. IANA Considerations

This document registers a new MIME type, application/session-policy+xml, and registers a new XML namespace.

7.1 MIME Registration for application/session-policy+xml

MIME media type name: application

MIME subtype name: session-policy+xml

Mandatory parameters: none

Optional parameters: Same as charset parameter application/xml as specified in RFC 3023 [7].

Encoding considerations: Same as encoding considerations of application/xml as specified in RFC 3023 [7].

Security considerations: See Section 10 of RFC 3023 [7] and Section 6 of this specification.

Interoperability considerations: none.

Published specification: This document.

Applications which use this media type: This document type has been used to download the session policy of a domain to SIP user agents.

Additional Information:

Magic Number: None

File Extension: .wif or .xml

Macintosh file type code: "TEXT"

Personal and email address for further information: Gonzalo Camarillo, <Gonzalo.Camarillo@ericsson.com>

Intended usage: COMMON

Author/Change controller: The IETF.

7.2 URN Sub-Namespace Registration for urn:ietf:params:xml:ns:sessionpolicy

This section registers a new XML namespace, as per the guidelines in [4]

URI: The URI for this namespace is urn:ietf:params:xml:ns:sessionpolicy.

Registrant Contact: IETF, SIPPING working group,<sipping@ietf.org>,
Gonzalo Camarillo, <Gonzalo.Camarillo@ericsson.com>

XML:

```
BEGIN
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
          "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type"
     content="text/html;charset=iso-8859-1"/>
  <title>Session Policy Namespace</title>
</head>
<body>
  <h1>Namespace for Session Policy Information</h1>
  <h2>application/session-policy+xml</h2>
  <p>See <a href="[[[URL of published RFC]]]">RFCXXXX</a>.</p>
</body>
</html>
END
```

8 References

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

[2] Handley, M. and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

[3] Hilt, V. and J. Rosenberg, "A Framework for Session-Specific Intermediary Session Policies in SIP", September 2003.

[4] Mealling, M., "The IETF XML Registry", draft-mealling-iana-xmlns-registry-05 (work in progress), June 2003.

[5] Moats, R., "URN Syntax", RFC 2141, May 1997.

[6] Moats, R., "A URN Namespace for IETF Documents", RFC 2648, August 1999.

[7] Murata, M., St. Laurent, S. and D. Kohn, "XML Media Types", RFC 3023, January 2001.

[8] Petrie, D., "A Framework for SIP User Agent Profile Delivery", draft-ietf-sipping-config-framework-02 (work in progress), February 2004.

[9] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.

[10] Rosenberg, J., "Traversal Using Relay NAT (TURN)", draft-rosenberg-midcom-turn-03 (work in progress), October 2003.

[11] 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.

Authors' Addresses

Volker Hilt
Bell Labs/Lucent Technologies
101 Crawfords Corner Rd
Holmdel, NJ 07733
USA

EMail: volkerh@bell-labs.com

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

EMail: Gonzalo.Camarillo@ericsson.com

Jonathan Rosenberg
dynamicsoft
72 Eagle Rock Avenue
East Hanover, NJ 07936

USA

EMail: jdrosen@dynamicsoft.com

Appendix A. Acknowledgements

Many thanks to Allison Mankin and Markus Hofmann for their contributions to this draft.

Intellectual Property Statement

Full Copyright Statement

Acknowledgment

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