SIPPING Working Group
Internet-Draft
Expires: March 1, 2004

G. Camarillo
Ericsson

September 2003

A Session Initiation Protocol (SIP) Event Package for Media Policy
draft-camarillo-sipping-policy-package-00.txt

Status of this Memo

Copyright Notice

Abstract

This document defines a SIP event package for session policies. User agents can subscribe to this event package to obtain information about the session policy of a domain (e.g., allowed and disallowed codecs or maximum bandwidth).

Table of Contents

1. Introduction

Some domains have certain policies regarding the types of sessions users can establish. These policies are typically enforced somehow. For example, if the policy of a domain disallows the use of a particular codec, access routers will discard packets that transport media encoded with that codec. Unfortunatelly, enforcement mechanisms do not usually inform the user about what is happening. They silently keep the user from doing anything against the policy.

Therefore, users need a means to obtain the policy of their domain in order not to try anything against it. Users also need to be informed about changes in this policy, since the session policy of a domain is a dynamic piece of information (e.g., high-bandwidth codecs are disallowed only in presence of a high number of users).

Other domains have 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.

We define a SIP event package that allows a user agent to subscribe to the session policy information of a domain.

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, RFC 2119 [1] and indicate requirement levels for compliant implementations.

3. Package Definition

This section fills in the details needed to specify an event package as defined in Section 4.4 of RFC 3265 [2].

3.1 Event Package Name

The name of this package is "session-policy". As specified in RFC 3265 [2], this value appears in the Event header field present in SUBSCRIBE and NOTIFY requests.

```
Event: session-policy
```

3.2 Event Package Parameters

No package specific Event header field parameters are defined for this event package.

3.3 SUBSCRIBE Bodies

A SUBSCRIBE for session policy events MAY contain a body. This body would serve the purpose of filtering the subscription. The definition of such a body is outside the scope of this specification.

A SUBSCRIBE for the session policy package MAY be sent without a body. This implies that the default session policy filtering policy has been requested. The default policy is that notifications are generated every time there is any change in the media policy for the user.

3.4 Subscription Duration

The default expiration of subscriptions to session policy state is one hour (3600 seconds).

3.5 NOTIFY Bodies

In this event package, the body of the notification contains a session policy document. This document describes the session policy of a domain for a user. All subscribers and notifiers MUST support the "application/session-policy+xml" data format described in Section 4. The subscribe request MAY contain an Accept header field. If no such header field is present, it has a default value of "application/session-policy+xml". If the header field is present, it MUST include "application/session-policy+xml", and MAY include any other types capable of representing session policy state.

3.6 Notifier Processing of SUBSCRIBE Requests

Session policy state can be sensitive information. Therefore, all subscriptions to it SHOULD be authenticated and authorized before approval. Authentication MAY be performed using any of the techniques available through SIP, including digest, S/MIME, TLS or other transport specific mechanisms. It is RECOMMENDED that a user be allowed to subscribe to their own session policy.

3.7 Notifier Generation of NOTIFY Requests

Notifications SHOULD be generated for the session policy package whenever there is a change in the session policy for the user.

3.8 Subscriber Processing of NOTIFY Requests

NOTIFY requests contain the full session policy state. The subscriber does not need to perform any type of information aggregation.

3.9 Handling of Forked Request

Session policy state is normally stored in some repository. Therefore, there is usually a single place where the session policy for a user is resident. This implies that a subscription for this information is readily handled by a single element with access to this repository. There is, therefore, no compelling need for a subscription to session policy information to fork. As a result, a subscriber MUST NOT create multiple dialogs as a result of a single subscription request. The required processing to guarantee that only a single dialog is established is described in Section 4.4.9 of RFC 3265 [2].

3.10 Rate of Notifications

For reasons of congestion control, it is important that the rate of notifications not become excessive. As a result, it is RECOMMENDED that the server not generate notifications for a single subscriber at a rate faster than once every 5 seconds.

3.11 State Agents

State agents have no role in the handling of this package.

4. Session Policy Information Format

Session policy information information is an XML document that MUST be well-formed and SHOULD be valid. Session 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 [3], using the namespace identifier 'ietf' defined by RFC 2648 [4] and extended by [6]. This URN is:

```
urn:ietf:params:xml:ns:sessionpolicy
```

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

5. Structure of the Session Policy Information

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 media 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.

6. 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.

6.1 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).

6.2 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).

6.3 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).

6.4 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).

6.5 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.

6.6 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>
```

7. 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.

7.1 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.

7.1.1 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, and the policy attribute contains the policy for that codec (allowed, or disallowed).

7.1.2 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 tranport element can have two attributes: name and policy. The name attribute identifies a transport, and the policy attribute contains the policy

for that tranport (allowed, or disallowed).

7.1.3 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).

7.1.4 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.

7.2 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>
```

8. Schema

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

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

9. 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>
```

10. Security Considerations

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

11. IANA Considerations

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

11.1 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 [5].

Encoding considerations: Same as encoding considerations of

application/xml as specified in RFC 3023 [5].

Security considerations: See Section 10 of RFC 3023 [5] and Section 10 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.

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

This section registers a new XML namespace, as per the guidelines in [6]

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
```

Normative References

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

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

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

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

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

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

Informational References

Author's Address

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

EMail: Gonzalo.Camarillo@ericsson.com

Intellectual Property Statement

Full Copyright Statement

Acknowledgment

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