

SIPPING  
Internet-Draft  
Intended status: Standards Track  
Expires: September 8, 2009

B. Rosen  
NeuStar, Inc.  
H. Schulzrinne  
Columbia U.  
H. Tschofenig  
Nokia Siemens Networks  
March 7, 2009

Session Initiation Protocol (SIP) Event Package for the Common Alerting  
Protocol (CAP)  
[draft-rosen-sipping-cap-03.txt](#)

#### 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/lid-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 September 8, 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.

Internet-Draft

SIP CAP

March 2009

## Abstract

The Common Alerting Protocol (CAP) is an XML document format for exchanging emergency alerts and public warnings. This document allows CAP documents to be distributed via the event notification mechanism available with the Session Initiation Protocol (SIP).

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction . . . . .</a>	<a href="#">3</a>
<a href="#">2.</a>	<a href="#">Terminology . . . . .</a>	<a href="#">3</a>
<a href="#">3.</a>	<a href="#">The 'common-alerting-protocol' Event Package . . . . .</a>	<a href="#">3</a>
<a href="#">3.1.</a>	<a href="#">Package Name . . . . .</a>	<a href="#">3</a>
<a href="#">3.2.</a>	<a href="#">Event Package Parameters . . . . .</a>	<a href="#">4</a>
<a href="#">3.3.</a>	<a href="#">SUBSCRIBE Bodies . . . . .</a>	<a href="#">4</a>
<a href="#">3.4.</a>	<a href="#">Subscription Duration . . . . .</a>	<a href="#">4</a>
<a href="#">3.5.</a>	<a href="#">NOTIFY Bodies . . . . .</a>	<a href="#">4</a>
<a href="#">3.6.</a>	<a href="#">Notifier Processing of SUBSCRIBE Requests . . . . .</a>	<a href="#">5</a>
<a href="#">3.7.</a>	<a href="#">Notifier Generation of NOTIFY Requests . . . . .</a>	<a href="#">5</a>
<a href="#">3.8.</a>	<a href="#">Subscriber Processing of NOTIFY Requests . . . . .</a>	<a href="#">5</a>
<a href="#">3.9.</a>	<a href="#">Handling of Forked Requests . . . . .</a>	<a href="#">6</a>
<a href="#">3.10.</a>	<a href="#">Rate of Notifications . . . . .</a>	<a href="#">6</a>
<a href="#">3.11.</a>	<a href="#">State Agents . . . . .</a>	<a href="#">6</a>
<a href="#">3.12.</a>	<a href="#">Examples . . . . .</a>	<a href="#">6</a>
<a href="#">3.13.</a>	<a href="#">Use of URIs to Retrieve State . . . . .</a>	<a href="#">6</a>
<a href="#">3.14.</a>	<a href="#">PUBLISH Bodies . . . . .</a>	<a href="#">6</a>
<a href="#">3.15.</a>	<a href="#">PUBLISH Response Bodies . . . . .</a>	<a href="#">7</a>
<a href="#">3.16.</a>	<a href="#">Multiple Sources for Event State . . . . .</a>	<a href="#">7</a>
<a href="#">3.17.</a>	<a href="#">Event State Segmentation . . . . .</a>	<a href="#">7</a>
<a href="#">3.18.</a>	<a href="#">Rate of Publication . . . . .</a>	<a href="#">7</a>
<a href="#">4.</a>	<a href="#">Examples . . . . .</a>	<a href="#">7</a>
<a href="#">5.</a>	<a href="#">Security Considerations . . . . .</a>	<a href="#">8</a>
<a href="#">5.1.</a>	<a href="#">Man-in-the-Middle Attacks . . . . .</a>	<a href="#">9</a>
<a href="#">5.2.</a>	<a href="#">Forgery . . . . .</a>	<a href="#">9</a>
<a href="#">5.3.</a>	<a href="#">Replay Attack . . . . .</a>	<a href="#">9</a>
<a href="#">5.4.</a>	<a href="#">Unauthorized Distribution . . . . .</a>	<a href="#">10</a>
<a href="#">6.</a>	<a href="#">IANA Considerations . . . . .</a>	<a href="#">10</a>
6.1.	Registration of the 'common-alerting-protocol' Event Package . . . . .	<a href="#">10</a>
6.2.	Registration of the 'application/common-alerting-protocol+xml' MIME type . . .	<a href="#">11</a>
<a href="#">7.</a>	<a href="#">Acknowledgments . . . . .</a>	<a href="#">11</a>

8. Normative References . . . . .	<a href="#">12</a>
Authors' Addresses . . . . .	<a href="#">12</a>

## [1.](#) Introduction

The Common Alerting Protocol (CAP) [[cap](#)] is an XML document format for exchanging emergency alerts and public warnings. This document allows CAP documents to be distributed via the event notification mechanism available with the Session Initiation Protocol (SIP).

Additionally, a MIME object is registered to allow CAP documents to be exchanged in other SIP documents.

## [2.](#) Terminology

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](#)].

## [3.](#) The 'common-alerting-protocol' Event Package

[RFC 3265](#) [[RFC3265](#)] defines a SIP extension for subscribing to remote nodes and receiving notifications of changes (events) in their states. It leaves the definition of many aspects of these events to concrete extensions, known as event packages. This document defines such an event package. This section fills in the information required for all event packages by [RFC 3265](#).

Additionally, [RFC 3903](#) [[RFC3903](#)] defines an extension that allows SIP User Agents to publish event state. According to [RFC 3903](#), any event package intended to be used in conjunction with the SIP PUBLISH method has to include a considerations section. This section also fills the information for all event packages to be used with PUBLISH requests.

This document defines a new "common-alerting-protocol" event package. Event Publication Agents (EPA) use PUBLISH requests to inform an

Event State Compositor (ESC) of changes in the common-alerting-protocol event package. Acting as a notifier, the ESC notifies subscribers about emergency alerts and public warnings.

### [3.1.](#) Package Name

The name of this package is "common-alerting-protocol". As specified in [RFC 3265](#) [RFC3265], this value appears in the Event header field present in SUBSCRIBE and NOTIFY requests. As specified in [RFC 3903](#) [RFC3903], this value also appears in the Event header field present in PUBLISH requests.

Rosen, et al.

Expires September 8, 2009

[Page 3]

---

Internet-Draft

SIP CAP

March 2009

### [3.2.](#) Event Package Parameters

[RFC 3265](#) [RFC3265] allows event packages to define additional parameters carried in the Event header field. This event package, "common-alerting-protocol", does not define additional parameters.

### [3.3.](#) SUBSCRIBE Bodies

[RFC 3265](#) [RFC3265] allows a SUBSCRIBE request to contain a body. This document allows the body to contain civic and geodetic location information to be carried. The 2D location shapes listed in [\[I-D.ietf-geopriv-pdif-lo-profile\]](#), e.g., <Point> <Polygon>, <Circle>, <Ellipse>, <ArcBand>, and a <civicAddress> element, defined in [\[RFC5139\]](#), in the body of the message. The recipient of the SUBSCRIBE message SHOULD use this information to restrict the warning messages that are being delivered. [Editor's note: Information about the type of alerts that shall be received may need to be indicated as well.]

### [3.4.](#) Subscription Duration

The default expiration time for subscriptions within this package is 3600 seconds. As per [RFC 3265](#) [RFC3265], the subscriber MAY specify an alternate expiration in the Expires header field.

### [3.5.](#) NOTIFY Bodies

As described in [RFC 3265](#) [RFC3265], the NOTIFY message will contain bodies describing the state of the subscribed resource. This body is

in a format listed in the Accept header field of the SUBSCRIBE request, or a package-specific default format if the Accept header field was omitted from the SUBSCRIBE request.

In this event package, the body of the notification contains a Common Alerting Protocol (CAP) document, i.e., an XML document. The format of the XML documents used by CAP are described in [[cap](#)].

For an initial notify, unlike for other event packages, there is no current initial state, unless there's a pending alert. Hence, returning a NOTIFY with a non-empty body only makes sense if there are indeed active alerts.

All subscribers and notifiers of the "common-alerting-protocol" event package MUST support the "application/common-alerting-protocol+xml" data format. The SUBSCRIBE request MAY contain an Accept header field. If no such header field is present, it has a default value of "application/common-alerting-protocol+xml" (assuming that the Event header field contains a value of "common-alerting-protocol"). If the

Accept header field is present, it MUST include "application/common-alerting-protocol+xml".

### [3.6.](#) Notifier Processing of SUBSCRIBE Requests

The contents of a CAP document contains public information. Hence, providing CAP documents may not require authorization by subscribers.

### [3.7.](#) Notifier Generation of NOTIFY Requests

[RFC 3265](#) [[RFC3265](#)] details the formatting and structure of NOTIFY messages. However, packages are mandated to provide detailed information on when to send a NOTIFY, how to compute the state of the resource, how to generate neutral or fake state information, and whether state information is complete or partial. This section describes those details for the common-alerting-protocol event package.

A notifier MAY send a NOTIFY at any time. Typically, it will send one when an alert or early warning message is available. The NOTIFY request contains a body containing one or multiple CAP document(s). The times at which the NOTIFY is sent for a particular subscriber,

and the contents of the body within that notification, are subject to any rules specified by the authorization policy that governs the subscription.

In the case of a pending subscription, when final authorization is determined, a NOTIFY can be sent. If the result of the authorization decision was success, a NOTIFY SHOULD be sent and SHOULD contain a complete CAP document. If the subscription is rejected, a NOTIFY MAY be sent. As described in [RFC 3265](#) [[RFC3265](#)], the Subscription-State header field indicates the state of the subscription.

The body of the NOTIFY MUST be sent using one of the types listed in the Accept header field in the most recent SUBSCRIBE request, or using the type "application/common-alerting-protocol+xml" if no Accept header field was present.

Notifiers will typically act as Event State Compositors (ESC) and thus will learn the 'common-alerting-protocol' event state via PUBLISH requests sent from authorized Event Publication Agents (EPAs).

### [3.8.](#) Subscriber Processing of NOTIFY Requests

[RFC 3265](#) [[RFC3265](#)] leaves it to event packages to describe the process followed by the subscriber upon receipt of a NOTIFY request, including any logic required to form a coherent resource state.

### [3.9.](#) Handling of Forked Requests

[RFC 3265](#) [[RFC3265](#)] requires each package to describe handling of forked SUBSCRIBE requests.

This specification only allows a single dialog to be constructed as a result of emitting an initial SUBSCRIBE request.

### [3.10.](#) Rate of Notifications

[RFC 3265](#) [[RFC3265](#)] requires each package to specify the maximum rate at which notifications can be sent.

Notifiers SHOULD NOT generate notifications for a single user at a rate of more than once every five seconds.

### [3.11.](#) State Agents

[RFC 3265](#) [[RFC3265](#)] requires each package to consider the role of state agents in the package and, if they are used, to specify how authentication and authorization are done. This specification allows state agents to be located in the network.

### [3.12.](#) Examples

An example is provided in [Section 4](#).

### [3.13.](#) Use of URIs to Retrieve State

[RFC 3265](#) [[RFC3265](#)] allows packages to use URIs to retrieve large state documents.

CAP documents are fairly small. This event package does not provide a mechanism to use URIs to retrieve large state documents.

### [3.14.](#) PUBLISH Bodies

[RFC 3903](#) [[RFC3903](#)] requires event packages to define the content types expected in PUBLISH requests.

In this event package, the body of a PUBLISH request may contain a CAP document. A CAP document describes an emergency alert or an early warning event.

All EPAs and ESCs MUST support the "application/common-alerting-protocol+xml" data format and MAY support other formats.

Note that this document does not mandate how CAP documents are made available to the Public Warning System, for example by authorities or similar organizations. The PUBLISH mechanism is one way.

### [3.15.](#) PUBLISH Response Bodies

This specification assumes that a PUBLISH also conveys a CAP document that is later sent further on to watchers.

### [3.16.](#) Multiple Sources for Event State

[RFC 3903](#) [[RFC3903](#)] requires event packages to specify whether multiple sources can contribute to the event state view at the ESC.

This event package allows different EPAs to publish CAP documents for a particular user. The concept of composition is not applicable for this application usage.

### [3.17.](#) Event State Segmentation

[RFC 3903](#) [[RFC3903](#)] defines segments within a state document. Each segment is defined as one of potentially many identifiable sections in the published event state.

This event package defines does not differentiate between different segments.

### [3.18.](#) Rate of Publication

[RFC 3903](#) [[RFC3903](#)] allows event packages to define their own rate of publication.

There are no rate-limiting recommendations for common-alerting-protocol publication. Since emergency alerts and early warning events are typically rare there is no periodicity, nor a minimum or maximum rate of publication.

## [4.](#) Examples

Here is an example of a CAP document.



```

<alert xmlns="urn:oasis:names:tc:emergency:cap:1.1">
  <identifier>KST01055887203</identifier>
  <sender>KST0@NWS.NOAA.GOV</sender>
  <sent>2003-06-17T14:57:00-07:00</sent>
  <status>Actual</status>
  <msgType>Alert</msgType>
  <scope>Public</scope>
  <info>
    <category>Met</category>
    <event>SEVERE THUNDERSTORM</event>
    <urgency>Severe</urgency>
    <certainty>Likely</certainty>
    <senderName>NATIONAL WEATHER SERVICE SACRAMENTO</senderName>
    <headline>SEVERE THUNDERSTORM WARNING</headline>
    <description> AT 254 PM PDT...
      NATIONAL WEATHER SERVICE
      DOPPLER RADAR INDICATED A SEVERE
      THUNDERSTORM OVER SOUTH CENTRAL ALPINE COUNTY...
      OR ABOUT 18 MILES SOUTHEAST OF
      KIRKWOOD... MOVING SOUTHWEST AT 5 MPH. HAIL...
      INTENSE RAIN AND STRONG DAMAGING WINDS
      ARE LIKELY WITH THIS STORM </description>
    <instruction> TAKE COVER IN A SUBSTANTIAL SHELTER
      UNTIL THE STORM PASSES </instruction>
    <contact>BARUFFALDI/JUSKIE</contact>
    <area>
      <areaDesc> EXTREME NORTH CENTRAL TUOLUMNE COUNTY
        IN CALIFORNIA, EXTREME NORTHEASTERN
        CALAVERAS COUNTY IN CALIFORNIA, SOUTHWESTERN
        ALPINE COUNTY IN CALIFORNIA </areaDesc>
      <polygon> 38.47,-120.14 38.34,-119.95 38.52,-119.74
        38.62,-119.89 38.47,-120.14 </polygon>
    </area>
  </info>
</alert>

```

Example for a Severe Thunderstorm Warning

## 5. Security Considerations

This section discusses security considerations when using SIP to distribute warning messages using CAP.

### [5.1.](#) Man-in-the-Middle Attacks

#### Threat:

The attacker could then conceivably attempt to impersonate the subject (the putative caller) to some SIP-based target entity.

#### Countermeasures:

Such an attack is implausible for several reasons. The subject's assertion:

- \* should be signed, thus causing any alterations to break its integrity and make such alterations detectable.
- \* the intended recipients may be listed in the optionally present audience restriction, which is a cleartext field. As such, it would not allow automatic processing but could give the receiving user further hints.
- \* Issuer is represented in the CAP document (in the <sender> element).
- \* validity period for the CAP document may be restricted.

### [5.2.](#) Forgery

#### Threat:

A malicious user could forge or alter a CAP document in order to convey messages to SIP entities that get immediate attention of users.

#### Countermeasures:

To avoid this kind of attack, the entities must assure that proper mechanisms for protecting the CAP documents are employed, e.g., signing the CAP document itself. Section 3.3.2.1 of [[cap](#)] specifies the signing of CAP documents.

### [5.3.](#) Replay Attack

#### Threat:

Theft of CAP documents described in this document and replay of it at a later time.

#### Countermeasures:

A CAP document contains the mandatory <identifier>, <sender>,

<sent> elements and an optional <expire> element. These attributes make the CAP document unique for a specific sender and

provide time restrictions. An entity that has received a CAP message already within the indicated timeframe is able to detect a replayed message and, if the content of that message is unchanged, then no additional security vulnerability is created. Nodes that enter the area of a disaster after the initial distribution of warnings have not yet seen the CAP message and, as such, would not be able to distinguish a replay from the initial message being sent around. However, if the threat that lead to the distribution of warning messages is still imminent then there is no reason not to worry about that message. The source distributing the early warning messages is, however, advised to carefully select a value for the <expires> element and it is RECOMMENDED to set this element.

#### [5.4.](#) Unauthorized Distribution

##### Threat:

When an entity receives a CAP message it has to determine whether the entity distributing the CAP messages is genuine to avoid accepting messages that are injected by malicious users with the potential desire to at least get the users immediate attention.

##### Countermeasures:

When receiving a CAP document a couple of verification steps must be performed. First, it needs to be ensured that the message was delivered via a trusted entity (such as a trusted SIP proxy) and that the communication channel between the User Agent and it's SIP proxy is properly secured to exclude various attacks at the SIP level. Then, the message contains the <sender> that may contain an entity that falls within the white list of the entity receiving the message. Finally, the message is protected by a digital signature and the entity signing the CAP message may again be listed in a white list of the receiving entity and may therefore be trusted. If none of these verification checks lead to a positive indication of a known sender then the CAP document should be treated as suspicious and configuration at the receiving entity may dictate how to process and display CAP documents in such a

case.

## [6.](#) IANA Considerations

### [6.1.](#) Registration of the 'common-alerting-protocol' Event Package

This specification registers an event package, based on the registration procedures defined in [RFC 3265](#) [[RFC3265](#)]. The following

Rosen, et al.

Expires September 8, 2009

[Page 10]

---

Internet-Draft

SIP CAP

March 2009

is the information required for such a registration:

Package Name: common-alerting-protocol

Package or Template-Package: This is a package.

Published Document: RFC XXX [Replace by the RFC number of this specification].

Person to Contact: Hannes Tschofenig, Hannes.Tschofenig@nsn.com

### [6.2.](#) Registration of the 'application/common-alerting-protocol+xml' MIME type

To: ietf-types@iana.org

Subject: Registration of MIME media type application/ common-alerting-protocol+xml

MIME media type name: application

MIME subtype name: common-alerting-protocol+xml

Required parameters: (none)

Optional parameters: charset; Indicates the character encoding of enclosed XML. Default is UTF-8 [[RFC3629](#)].

Encoding considerations: Uses XML, which can employ 8-bit characters, depending on the character encoding used. See [RFC 3023](#) [[RFC3023](#)], [Section 3.2](#).

Security considerations: This content type is designed to carry payloads of the Common Alerting Protocol (CAP).

Interoperability considerations: This content type provides a way to convey CAP payloads.

Published specification: RFC XXX [Replace by the RFC number of this specification].

Applications which use this media type: Applications that convey alerts and early warnings according to the CAP standard.

Additional information: OASIS has published the Common Alerting Protocol at [[cap](#)].

Person & email address to contact for further information: Hannes

Tschofenig, Hannes.Tschofenig@nsn.com  
Intended usage: Limited use  
Author/Change controller: IETF SIPPING working group  
Other information: This media type is a specialization of  
application/xml [RFC 3023](#) [[RFC3023](#)], and many of the considerations  
described there also apply to application/  
common-alerting-protocol+xml.

## [7.](#) Acknowledgments

The authors would like to thank Cullen Jennings for supporting this work. We would also like to thank the participants of the Early Warning Adhoc meeting at IETF#69.

Rosen, et al. Expires September 8, 2009 [Page 11]

---

Internet-Draft SIP CAP March 2009

## [8.](#) Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", March 1997.
- [cap] Jones, E. and A. Botterell, "Common Alerting Protocol v. 1.1", October 2005.
- [RFC3265] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", [RFC 3265](#), June 2002.
- [RFC3903] Niemi, A., "Session Initiation Protocol (SIP) Extension for Event State Publication", [RFC 3903](#), October 2004.
- [RFC3023] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", [RFC 3023](#), January 2001.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, [RFC 3629](#), November 2003.
- [I-D.ietf-geopriv-pdif-lo-profile]  
Winterbottom, J., Thomson, M., and H. Tschofenig, "GEOPRIV PIDF-LO Usage Clarification, Considerations and Recommendations", [draft-ietf-geopriv-pdif-lo-profile-14](#) (work in progress), November 2008.

[RFC5139] Thomson, M. and J. Winterbottom, "Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO)", [RFC 5139](#), February 2008.

#### Authors' Addresses

Brian Rosen  
NeuStar, Inc.  
470 Conrad Dr  
Mars, PA 16046  
US

Phone:  
Email: [br@brianrosen.net](mailto:br@brianrosen.net)

Rosen, et al.

Expires September 8, 2009

[Page 12]

---

Internet-Draft

SIP CAP

March 2009

Henning Schulzrinne  
Columbia University  
Department of Computer Science  
450 Computer Science Building  
New York, NY 10027  
US

Phone: +1 212 939 7004  
Email: [hgs+ecrit@cs.columbia.edu](mailto:hgs+ecrit@cs.columbia.edu)  
URI: <http://www.cs.columbia.edu>

Hannes Tschofenig  
Nokia Siemens Networks  
Linnoitustie 6  
Espoo 02600  
Finland

Phone: +358 (50) 4871445  
Email: Hannes.Tschofenig@gmx.net  
URI: <http://www.tschofenig.priv.at>