ATOCA
Internet-Draft
Intended status: Standards Track
Expires: April 21, 2011

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

October 18, 2010

Session Initiation Protocol (SIP) Event Package for the Common Alerting Protocol (CAP)

draft-ietf-atoca-cap-00.txt

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

1. Introduction

The Common Alerting Protocol (CAP) [cap] is an XML document format for exchanging emergency alerts and public warnings. The abstract architectural description for the distribution of alerts can be found in [I-D.ietf-atoca-requirements].

This document specifies how CAP documents are 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 messages.

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].

Note that early warning specific definitions can be found in [I-D.ietf-atoca-requirements].

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, i.e. event packages. This document defines such a new "common-alerting-protocol" event package. RFC 3903 [RFC3903] defines an extension that allows SIP User Agents to publish event state. 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.

3.2. Event Package Parameters

RFC 3265 [RFC3265] allows event packages to define additional parameters carried in the Event header field. This event package 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 event filters, see [RFC4660] and [RFC4661] with the information elements listed in the subsections below.

3.3.1. Location Filter

The 2D location shapes listed in [RFC5491] (e.g., <Point> <Polygon>, <Circle>, <Ellipse>, <ArcBand>) and the <civicAddress> element, defined in [RFC5139]. Repeating these elements is allowed and the semantic is equivalent to a union. The <alertArea> element indicates the area of interest; whenever an event happens in this area an alert message is delivered.

An example can be found below:

```
<?xml version="1.0" encoding="UTF-8"?>
<filter-set
    xmlns="urn:ietf:params:xml:ns:simple-filter"
    xmlns:af="urn:ietf:params:xml:ns:alert-filter"
    xmlns:gml="http://www.opengis.net/gml"
    xmlns:gs="http://www.opengis.net/pidflo/1.0">

    <filter id="123" uri="sip:presentity@example.com">
        <trigger>
            <af:alertArea>
                <gs:Circle
                    srsName="urn:ogc:def:crs:EPSG::4326">
                    <gml:pos>42.5463 -73.2512</gml:pos>
                    <gs:radius
                        uom="urn:ogc:def:uom:EPSG::9001">
                        5000
                    </gs:radius>
                </gs:Circle>
            </af:alertArea>
        </trigger>
    </filter>
</filter-set>
```

Example of a SIP SUBSCRIBE Body with a Location Filter

3.3.2. Service Filter

To filter different types of alerts the <serviceFilter> element MUST be included as a child element of the <what> element and it MUST list one or more Service URNs [RFC5031], which indicate the type of alerts the recipient is interested in. This document registers a number of alerts relevant for exigent communications, which can be found in Section 6.

An example can be found below:

```
<?xml version="1.0" encoding="UTF-8"?>
<filter-set
    xmlns="urn:ietf:params:xml:ns:simple-filter"
    xmlns:af="urn:ietf:params:xml:ns:alert-filter">
    <filter id="123" uri="sip:presentity@example.com">
        <what>
            <af:serviceFilter>
              urn:service:warning.met
            </af:serviceFilter>
        </what>
```

```
        </filter>
    </filter-set>
```

Example of a SIP SUBSCRIBE Body with a Service Filter

3.3.3. Rate Control

[I-D.ietf-sipcore-event-rate-control] extends the SIP events framework by defining three "Event" header field parameters that allow a subscriber to set a minimum, a maximum and an average rate of event notifications generated by the notifier. This allows a subscriber to have overall control over the stream of notifications, for example to avoid being flooded.

A notifier is required to send a NOTIFY request immediately after creation of a subscription. If state is not available at that time, then the NOTIFY request may be sent with no content. A separate NOTIFY containing an alert message may be generated some time later when it becomes available. Figure 1 shows a SUBSCRIBE/NOTIFY exchange.

```
Subscriber          Notifier
   |---SUBSCRIBE(1)--->|     Create subscription
   |<--------200--------|
   |<-----NOTIFY(2)----|     Return initial notify with no state
   |--------200------->|
   |<-----NOTIFY(3)----|     Alert message comes available
   |--------200------->|
```

Figure 1: SUBSCRIBE/NOTIFY with Rate Control

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 may contain public information, depending on the alert message type and the intended recipient of the alert message. It is, however, expected that in many cases providing CAP documents does 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.

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 act as Event State Compositors (ESC). Thus, they learn the 'common-alerting-protocol' event state via PUBLISH requests sent from authorized Event Publication Agents (EPAs). A Notifier may also be an EPA, or might accept PUBLISH requests from authorized 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.

Given that a single SUBSCRIBE might include multiple services and locations, it is reasonable and useful for a SUBSCRIBE request to fork and to reach multiple UAs. This is equivalent to multiple sources providing alerts for the same geographical, with a dedicated relay serving an aggregation function.

As a result, a forked SUBSCRIBE request can install multiple subscriptions. Subscribers to this package MUST be prepared to install subscription state for each NOTIFY generated as a result of a single SUBSCRIBE.

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.

3.15. PUBLISH Response Bodies

This specification assumes that the response to a PUBLISH does not contain a body.

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. This event package allows rate control to be utilized, as described in Section 3.3.3.

4. Examples

Here is an example of a CAP document.


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

<alert xmlns="urn:oasis:names:tc:emergency:cap:1.1">
    <identifier>KSTO1055887203</identifier>
    <sender>KSTO@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.

Based on the framework outlined in [I-D.ietf-atoca-requirements] the following security concerns arise:

Amplification Attacks: An adversary could inject alerts into the message handling system and therefore a single PUBLISH request could potentially results in millions of NOTIFY messages delivered to receivers. Injecting messages may happen at a number of ways, such as by adversaries who manage to impersonate a legitimate originator, a relay or gateway. Ensuring that only authorized entities are permitted to inject alerts is a pre-condition. This does, however, not help if the host of a trusted participant in the message handling system got compromised.

Forgery of Alerts: Alerts may get modified or replayed. The former is possible if the adversary manages to get access to a relay or gateway. Two mechanisms are proposed for countering forgery: using digital signatures or channel security (TLS). The first provides end-to-end security; the second utilizes a hop by hop security model based on a transitive chain of trust.

The sub-sections below discuss these threats and their countermeasures in more detail.

5.1. Amplification

Threat:

The attacker could then conceivably attempt to impersonate an originator, or a relay. A side effect of being able to inject an alert for distribution is the amplification effect.

Countermeasures:

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 entities.

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 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.

5.1.1. Forgery of Alerts

Threat:

A malicious user could forge a CAP document. Alternatively, a CAP document distributed earlier could be replied.

Countermeasures:

To avoid forgery, the entities must assure that proper mechanisms for protecting the CAP documents are employed, for example signing the CAP document itself or securing the communication between participating entities using TLS. Section 3.3.2.1 of [cap] specifies the signing of CAP documents.

Regarding replay attacks the following observations can be made. 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 originator/ author 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. Recipients who enter the area of a disaster after the initial distribution of warnings may not yet have seen the original 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 originator/author distributing the alert is, however, adviced to carefully select a value for the <expires> element and it is RECOMMENDED to set a value for this element.

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 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.

6.3. Early Warning Service URNs

In according with RFC 5031 this document defines a new top-level service called 'warning'. This section defines the first service registration within the IANA registry using the top-level service label 'warning'.

The 'warning' service type describes emergency services requiring an immediate action or remedy by the recipient of the alert message as instructed by the author of the message. Additional sub-services can be added after expert review and must be of general public interest and have a similar emergency nature. The expert is designated by the ECRIT working group, its successor, or, in their absence, the IESG. The expert review should only approve emergency services that are offered widely and in different countries, with approximately the same caller expectation in terms of services rendered.

The following list contains the initial IANA registration for the 'warning' service.

urn:service:warning.geo: Geophysical (inc. landslide)

urn:service:warning.met: Meteorological (inc. flood)

urn:service:warning.safety: General emergency and public safety

urn:service:warning.security: Law enforcement, military, homeland and local/private security urn:service:warning.rescue: Rescue and recovery

urn:service:warning.fire: Fire suppression and rescue

urn:service:warning.health: Medical and public health

urn:service:warning.env: Pollution and other environmental

urn:service:warning.transport: Public and private transportation

urn:service:warning.infra: Utility, telecommunication, other non-transport infrastructure

urn:service:warning.cbrne: Chemical, Biological, Radiological, Nuclear or High-Yield Explosive threat or attack

urn:service:warning.other: Other events

7. Open Issues

The authors would like to point to a number of issues that require discussion:

Rate Control: The -00 version of the document introduced rate control for notifications Section 3.3.3. Is this functionality is needed?

Early Warning Service URNs: Specifying services is always difficult since there is no universally agreed service semantic. This document contains a proposal that re-use the classification in the CAP specification [cap]. Is the proposal acceptable?

Event Filter: By using [RFC4660] and [RFC4661] filters in the body of a SUBSCRIBE the number of notifications can be reduced to those of interest to the subscriber. There is a certain overhead associated with the generic usage of those event filters. Should alternatives be considered?

Forked SUBSCRIBE Requests: This document allows forked subscribe request. This is useful when a single service is offered by more than one entity and therefore related to the cases discussed in [I-D.forte-lost-extensions] and in [I-D.forte-ecrit-service-classification]. For example, imagine a warning service like 'urn:service:warning.geo' that is advertised by a number of different service providers.

Security: The security consideration section was re-written and focuses now mostly on two types of attacks, namely amplificiation and forgery. Does this reflect the understanding of the group?

8. Acknowledgments

The authors would like to thank Cullen Jennings for his early support and the participants of the Early Warning Adhoc meeting at IETF#69 for their feedback.

We would furthermore like to thank Martin Thomson for his detailed draft review in July 2010.

9. References

9.1. 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.

[RFC5491] Winterbottom, J., Thomson, M., and H. Tschofenig, "GEOPRIV Presence Information Data Format Location Object (PIDF-LO) Usage Clarification, Considerations, and Recommendations", RFC 5491, March 2009.

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

[RFC5031] Schulzrinne, H., "A Uniform Resource Name (URN) for Emergency and Other Well-Known Services", RFC 5031, January 2008.

[I-D.ietf-sipcore-event-rate-control]
Niemi, A., Kiss, K., and S. Loreto, "Session Initiation Protocol (SIP) Event Notification Extension for Notification Rate Control", draft-ietf-sipcore-event-rate-control-04 (work in progress), July 2010.

[RFC4660] Khartabil, H., Leppanen, E., Lonnfors, M., and J. Costa-Requena, "Functional Description of Event Notification Filtering", RFC 4660, September 2006.

[RFC4661] Khartabil, H., Leppanen, E., Lonnfors, M., and J. Costa-Requena, "An Extensible Markup Language (XML)-Based Format for Event Notification Filtering", RFC 4661, September 2006.

9.2. Informative References

[I-D.ietf-atoca-requirements]
Schulzrinne, H., Norreys, S., Rosen, B., and H. Tschofenig, "Requirements, Terminology and Framework for Exigent Communications", draft-ietf-atoca-requirements-00 (work in progress), September 2010.

[I-D.forte-lost-extensions]
Forte, A. and H. Schulzrinne, "Location-to-Service Translation Protocol (LoST) Extensions", draft-forte-lost-extensions-02 (work in progress), September 2010.

[I-D.forte-ecrit-service-classification]
Forte, A. and H. Schulzrinne, "Labels for Common Location-Based Services", draft-forte-ecrit-service-classification-03 (work in progress), January 2010.

Authors' Addresses

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

Phone:
Email: br@brianrosen.net

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
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

ATOCA
Internet-Draft
Intended status: Informational
Expires: September 13, 2012

H. Schulzrinne
Columbia University
S. Norreys
BT Group
B. Rosen
NeuStar, Inc.
H. Tschofenig
Nokia Siemens Networks

March 12, 2012

Requirements, Terminology and Framework for Exigent Communications
draft-ietf-atoca-requirements-03.txt

Abstract

Before, during and after emergency situations various agencies need to provide information to a group of persons or to the public within a geographical area. While many aspects of such systems are specific to national or local jurisdictions, emergencies span such boundaries and notifications need to reach visitors from other jurisdictions.

This document provides terminology, requirements and an architectural description for protocols exchanging alerts between IP-based end points.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

1.1. Classical Early Warning Situations

During large-scale emergencies, public safety authorities need to reliably communicate with citizens in the affected areas, to provide warnings, indicate whether citizens should evacuate and how, and to dispel misinformation. Accurate information can reduce the impact of such emergencies.

Traditionally, emergency alerting has used church bells, sirens, loudspeakers, radio and television to warn citizens and to provide information. However, techniques, such as sirens and bells, provide limited information content; loud speakers cover only very small areas and are often hard to understand, even for those not hearing impaired or fluent in the local language. Radio and television offer larger information volume, but are hard to target geographically and do not work well to address the "walking wounded" or other pedestrians. Both are not suitable for warnings, as many of those needing the information will not be listening or watching at any given time, particularly during work/school and sleep hours.

This problem has been illustrated by the London underground bombing on July 7, 2006, as described in a government report [July2005]. The UK authorities could only use broadcast media and could not, for example, easily announce to the "walking wounded" where to assemble.

1.2. Exigent Communications

With the usage of the term 'Exigent Communications' this document aims to generalize the concept of conveying alerts to IP-based systems and at the same time to describe the actors that participate in the messaging communication. More precisely, exigent communications is defined as:

> Communication that requirs immediate action or remedy. Information about the reason for action and details about the steps that have to be taken are provided in the alert message.
>
> An alert message (or warning message) is a cautionary advice about something imminent (especially imminent danger or other unpleasantness). In the context of exigent communication such an alert message refers to a future, ongoing or past event as the signaling exchange itself may relate to different stages of the lifecycle of the event. The alert message itself, and not the signaling protocol that convey it, provides sufficient context about the specific state of the lifecycle the alert message refers to.

On a high level the communication occurs in two phases with the subscription phase sometimes being implicit:

Subscription:

In this step Recipients express their interest in receiving certain types of alerts. This step happens prior to the actual delivery of the alert. This expression of interest may be in form of an explicit communication step by having the Receiver send a subscribe message (potentially with an indication of the type of alerts they are interested in, the duration of the subscription and a number of other indicators). For example, parents may want to be alerted of emergencies affecting the school attended by their children and adult children may need to know about emergencies affecting elderly parents. The subscription step may, however, also happen outside the Internet communication infrastructure and instead by the Recipient signing a contract and thereby agreeing to receive certain alerts. The Receiver, a software application, still needs to be configured in such a way that incoming alerts are accepted, processed and passed up to the user interface alerting a Recipient. Additionally, certain subscriptions may happen without the Recipient's explicit consent and without the Receiver sending a subscription. For example, a Tsunami flood alert may be delivered to all Receivers in case they are located in a specific geographical area.

It is important to note that a protocol interaction initiated by the Receiver may need to take place to subscribe to certain types of alerts. In some other cases the subscription does not require such interaction from the Receiver. Orthogonal to the need to have a protocol interaction is the question of opt-in vs. opt-out. Whether the Recipient, as a human actor, needs to consent to receive certain types of alerts is a policy decision that is largely outside the scope of a technical specification.

Alert Delivery:

In this step the alert message is distributed to one or multiple Receivers. The Receiver as a software module that presents the alert message to the Recipient. The alert encoding is accomplished via the Common Alerting Protocol (CAP) and such an alert message contains useful information needed for dealing with the imminent danger.

Note that an alert receiver software modules may not necessarily only be executed on end devices humans typically carry around, such as mobile phones, Internet tablets, or laptops. Instead, alerts may well be directly sent to displays in subway stations, or electronic bill boards. Furthermore, a software module that obtains an alert may not necessarily need to interact with a human (as the Recipient) but may instead use it as input to another process to trigger automated behaviors, such as closing vents during a chemical spill or activating sirens or other warning systems in commercial buildings.

This document provides terminology, requirements and an architectural description. Note that the requirements focus on the communication protocols for subscription and alert delivery rather than on the content of the alert message itself. With the usage of CAP these alert message content requirements are delegated to the Authors and Originators of alerts.

2. Terminology

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119], with the important qualification that, unless otherwise stated, these terms apply to the design of a protocol conveying warning messages, not its implementation or application.

Alert messages are typically produced by humans and consumed by users, Authors and Recipients in our system, are the sources and sinks of alert messages.

The Author is a human responsible for creating the content of the alert message, and to make a decision about the intended recipients, even though the exact list of recipients may be unknown to the Author at the time of writing the alert message. Instead, the recipients may, for example, be described in terms of a geographical region, or recipients with interest in a specific alert type.

The Recipient is a consumer of the delivered alert message. It is a human reading the alert message.

From the user’s perspective, all alert message transfer activities are performed by a monolithic Message Handling Service (MHS), even though the actual service can be provided by many independent organizations. The Message Handling Service (MHS) performs a single end-to-end transfer of warning messages on behalf of the Author to reach the Recipients.

Figure 1 shows the relationships among transfer participants. Transfers typically entail one or more Relays. However, direct delivery from the Originator to Receiver is possible.

```
       ++==========++                        ++===========++
       ||  Author  ||                        || Recipient ||
       ++====++====++                        ++===========++
             ||                                     /\
             ||                                     ||
             \/                                     ||
        +----------+                            +---++----+
        |          |                            |         |
      /-+----------+----------------------------+---------+---\
      | |          |     Message Handling       |         |   |
      | |Originator|       System (MHS)         |Receiver |   |
      | |          |                            |         |   |
      | +---++-----+                            +---------+   |
      |     ||                                      /\        |
      |     ||                                      ||        |
      |     \/                                      ||        |
      | +---------+         +---------+        +-+--++---+    |
      | |  Relay  +======-=>|  Relay  +=======>|  Relay  |    |
      | +---------+         +----++---+        +---------+    |
      |                          ||                           |
      |                          ||                           |
      |                          \/                           |
      |                     +---------+                       |
      |                     | Gateway +-->                    |
      |                     +---------+                       |
      \-------------------------------------------------------/

     Legend: === and || lines indicate primary (possibly
                 indirect) transfers or roles
```

Figure 1: Relationships Among MHS Actors

2.1. Originator

The Originator ensures that a warning message is valid for transfer and then submits it to a Relay. A message is valid if it conforms to both communication and warning message encapsulation standards and local operational policies. The Originator can simply review the message for conformance and reject it if it finds errors, or it can create some or all of the necessary information.

The Originator serves the Author and can be the same entity in absence of a human crafting alert messages.

The Originator also performs any post-submission, Author-related administrative tasks associated with message transfer and delivery. Notably, these tasks pertain to sending error and delivery notices, and enforcing local policies. The Author creates the message, but the Originator handles any transmission issues with it.

2.2. Relay

The Relay performs MHS-level transfer-service routing and store-and-forward, by transmitting or retransmitting the message to its Recipients. The Relay may add history information (e.g., the SIP History Info [RFC4244] serves as a good example of the type of information that may be conveyed) or security related protection (e.g., as available with SIP Identity [RFC4474]) but does not modify the envelope information or the message content semantics.

A Message Handling System (MHS) network consists of a set of Relays. This MHS network is typically above any underlying IP network but may involve technologies like IP multicast.

2.3. Gateway

A Gateway connects heterogeneous communication infrastructures and its purpose is to emulate a Relay and the closer it comes to this, the better. A Gateway needs the ability to modify message content.

Differences between the different communication systems can be as small as minor syntax variations, but they usually encompass significant, semantic distinctions. Hence, the Relay function in a Gateway presents a significant design challenge, if the resulting performance is to be seen as nearly seamless. The challenge is to ensure end-to-end communication between the communication services, despite differences in their syntax and semantics.

2.4. Receiver

The Receiver performs final delivery and is typically responsible for ensuring that the appropriate user interface rendering is executed to interact with the Recipient.

3. Framework

Section 1 describes the basic two steps that are involved with the alert message handling, namely subscription and alert delivery. From an architectural point of view there are, however, a few variations possible depending on the characteristics of the subscription process and the style of message delivery. This section offers more details on the communication architecture. Note that this document does not mandate a specific deployment architecture. Instead it aims to illustrate how various different protocol components fit together to present the reader with the ’big picture’.

3.1. Small Group Alert Delivery

We start our description with the so-called "school closed" example where school authorities send alerts to all parents to notify them about the fact that their children cannot attend school. Parents subscribe to these events when their children start attending the school and unsubscribe when they are finished with a particular school. The subscription procedures establishes some form of group communication by requiring an initial registration procedure. Typically, alert messages stay within the closed group and are not shared with others and alert message delivery is point-to-point with whatever communication protocol is most suitable. This also means that the alerts reach those who have subscribed rather than those who are in the vicinity of the school. The number of Recipients is typically rather small, in the order of hundreds to several thousands.

A variation of the "school closed" example is an explicit subscription model where no closed group pattern exists. The main difference to the former case is in the authorization model. Consider a traveler who would like to receive weather alerts about a specific geographical region. He may have to manually search for how to subscribe to alerts for the desired region, potentially looking a different subscription points for different types of alerts. As an automated version of this procedure some form of discovery may help to find these subscription servers. The approach described in [I-D.rosen-ecrit-lost-early-warning] is one possible way to discovery such alert subscription servers. The number of alert message Recipients is much larger than in the previous school example but will typically stay below the millions.

These alert delivery examples are supported by a number of standardized communication protocols, such as SIP, XMPP, eMail, or RSS feeds.

Note that there are optimizations for application layer alert delivery that mimic a multicast delivery with the help of Relays. However, a subscription is still necessary by the Receiver and the last-hop delivery of the alert is still done using unicast transmission.

While these two examples are important for many deployments they are not in focus of the ATOCA working group.

3.2. Mass Alert Delivery

With the next category we move to a scenario where large number of Recipients shall be notified but the subscription itself is implicit, as it is the case when persons are within a specific region that can easily be reached by making use of broadcast link layer technologies. The placement of the actors from Figure 1 is thereby important. An Originator distributes the alert message to Relays within the geographically affected area. Those Relays are located within Internet Service Providers so that multicast and broadcast communication protocols can be utilized for efficient distribution to a large number of Recipients within the affected area. When the alert message delivery has to be accomplished at the network layer then various requirements, such as the ability to traverse NATs and firewalls, have to be met by such a protocol. In this scenario the number of alert message Recipients is very large, potentially in the millions.

As a variation of the previously described model consider an alert distribution with subscriptions to the alerts. Figure 2 shows the architecture.

A discovery server ensures that Receivers are able to learn the local alert distribution servers. Once a Receiver had discovered a local alert distribution server it sends a subscribe message to it. As a response, it will receive information about the security credentials the alert distribution server is going to use for subsequent alert delivery.

When an Author creates an alert for distribution the affected region will be indicated and so the alert will be sent to a Relay within the realm of the local alert distribution server and a notification will be sent to all the subscribed Receivers.

```
                           ,-----------.
                           | Discovery |
                           | Server    |
                           '...........'
                                :
                                :
    ,''''''''''''''''''''''''\  :
    | Local          ,------. | :
    | Alert          | Local| | :                ...................
    | Distribution   | Relay|.|..:   Alert        | +------+ Author |
    | Server         '......'-+<-----------------+-|Sender|    O   |
    |                  |      |    Notification  | |      |   /|\  |
    ''''''''''''''''''''+''''''                  | +------+   / \  |
       ^        Alert  |                         '-----------------'
    Subscr.  +---------+
       |     |  Notification
       |     |
       |     V
    .....................
    | +------+ Recipient|
    | |Recvr |    O     |
    | |      |   /|\    |
    | +------+   / \    |
    '-------------------'
```

Figure 2: Multicast/Broadcast Alert Delivery Mechanism with Explicit Subscription

4. Requirements

The requirements listed below focus on the goal of mass alert distribution, which has to utilize multicast/broadcast communication for scalability reasons. The requirements for point-to-point alert delivery are shown in Appendix A for completeness reasons only since the focus of the IETF ATOCA working group is on the multicast/ broadcast alert delivery.

4.1. Requirements for the Discovery of an Alert Distribution Server

Req-D1:

The protocol solution MUST allow a receiver to discover a local alert distribution server, as discussed in Section 3 and shown in Figure 2, and to discover the necessary security credentials for subsequent alert message distribution.

4.2. Requirements for Multicast/Broadcast Alert Message Delivery

Req-B1:

The protocol solution MUST leverage multicast/broadcast technologies. This implies non-TCP transport and congestion control being considered.

Req-B2:

The protocol solution MUST allow delivery of messages simultaneously to a large audience.

Req-B3:

The protocol solution MUST be able to traverse firewalls and NATs as they are common in today's deployments.

5. IANA Considerations

This document does not require actions by IANA.

6. Security Considerations

Figure 1 shows the actors for delivering an alert message assuming that a prior subscription has taken place already. The desired security properties of an MHS for conveying alerts will depend on the number of administrative domains involved. Each administrative domain can have vastly different operating policies and trust-based decision-making. One obvious example is the distinction between alert messages that are exchanged within an closed group (such as alert messages received by parents affecting the school attended by their children) and alert messages that are exchanged between independent organizations (e.g., in case of large scale disasters). The rules for handling both types of communication architectures tend to be quite different. That difference requires defining the boundaries of each.

Operation of communication systems that are used to convey alert messages are typically carried out by different providers (or operators). Since each be in operated in an independent administrative domain it is useful to consider administrative domain boundaries in the description to facilitate discussion about designs, policies and operations that need to distinguish between internal issues and external entities. Most significant is that the entities communicating across administrative boundaries typically have the added burden of enforcing organizational policies concerning external communications. For example, routing alerts between administrative domains can create requirements, such as needing to route alert messages between organizational partners over specially trusted paths.

The communication interactions are subject to the policies of that domain, which cover concerns such as these:

- Reliability
- Access control
- Accountability
- Content evaluation, adaptation, and modification

Many communication systems make the distinction of administrative domains since they impact the requirements on security solutions. However, with the distribution of alert messages a number of additional security threats need to be addressed. Due to the nature of alerts it is quite likely that end device implementations will offer user interface enhancements to get the Recipients attention whenever an alert arrives, which is an attractive property for adversaries to exploit. Below we list the most important threats any solution will have to deal with.

Originator Impersonation:

An attacker could then conceivably attempt to impersonate the Originator of an alert message. This threat is particularly applicable to those deployment environments where authorization decisions are based on the identity of the Originator.

Alert Message Forgery:

An attacker could forge or alter an alert message in order to convey custom messages to Recipients to get their immediate attention.

Replay:

An attacker could obtain previously distributed alert messages and to replay them at a later time in the hope that Recipients could be tricked into believing they are fresh.

Unauthorized Distribution:

When a Receiver receives an alert message it has to determine whether the Author distributing the alert messages is genuine to avoid accepting messages that are injected by malicious entities with the potential desire to at least get the immediate attention of the Recipient.

Amplification Attack:

An attacker may use the Message Handling System to inject a single alert message for distribution that may then be instantly turned into potentially millions of alert messages for distribution.

One important security challenge is related to authorization. When an alert message arrives at the Receiver then certain security checks may need to be performed to ensure that the alert message meets certain criteria. The final consumer of the alert message is, however, the Recipient - a human. From a security point of view the work split between the Recipient and the Receiver for making the authorization decision is important, particularly when an alert message is rejected due to a failed security verification by the Receiver. False positives may be fatal but accepting every alert message lowers the trustworthiness in the overall system.

7. Acknowledgments

This document re-uses text from [RFC5598]. The authors would like to thank Dave Crocker for his work.

The authors would like to thank Martin Thomson, Carl Reed, Leopold Murhammer, and Tony Rutkowski for their comments.

At IETF#79 the following persons provided feedback leading to changes in this document: Keith Drage, Scott Bradner, Ken Carberg, Keeping Li, Martin Thomson, Igor Faynberg, Mark Wood, Peter Saint-Andre.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5598] Crocker, D., "Internet Mail Architecture", RFC 5598, July 2009.

8.2. Informative References

[I-D.rosen-ecrit-lost-early-warning]
Rosen, B., Schulzrinne, H., and H. Tschofenig, "A Uniform

Resource Name (URN) for Early Warning Emergency Services and Location-to-Service Translation (LoST) Protocol Usage", draft-rosen-ecrit-lost-early-warning-01 (work in progress), July 2009.

[July2005] , ., "Report of the 7 July Review Committee, ISBN 1 85261 878 7", (PDF document), http://www.london.gov.uk/assembly/reports/7july/report.pdf, June 2006.

[RFC4244] Barnes, M., "An Extension to the Session Initiation Protocol (SIP) for Request History Information", RFC 4244, November 2005.

[RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, August 2006.

[RFC5582] Schulzrinne, H., "Location-to-URL Mapping Architecture and Framework", RFC 5582, September 2009.

Appendix A. Supplementary Requirements

A.1. Requirements for Alert Subscription

The requirements listed below refer to the alert subscription phase as it is used to tailor alert message delivery in a point-to-point alert delivery scenario. As noted in the main part of the document these requirements are not the main focus of the ATOCA work.

Req-S1:

The protocol solution MUST allow a potential Recipient to indicate the language used by alert messages.

Req-S2:

The protocol solution MUST allow a potential Recipient to express the geographical area it wants to receive alerts about.

Req-S3:

The protocol solution MUST allow a potential Recipient to indicate preferences about the type of alerts it wants to receive.

Req-S4:

The protocol solution MUST allow a potential Recipient to express preference for certain media types. The support for different media types depends on the content of the warning message but also impacts the communication protocol. This functionality is, for example, useful for hearing and vision impaired persons.

A.2. Point-to-Point Alert Delivery

Req-P1:

The protocol solution MUST build on existing communication protocols and support the delivery of alert messages. Examples of such protocols are SIP, XMPP, Atom, eMail.

Req-P2:

The protocol solution MUST allow targeting notifications to specific subscribers.

Authors' Addresses

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
URI: http://www.cs.columbia.edu

Steve Norreys
BT Group
1 London Road
Brentwood, Essex CM14 4QP
UK

Phone: +44 1277 32 32 20
Email: steve.norreys@bt.com

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

Phone:
Email: br@brianrosen.net

Hannes Tschhofenig
Nokia Siemens Networks
Linnoitustie 6
Espoo 02600
Finland

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