ECRIT
Internet-Draft
Intended status: Standards Track
Expires: September 2, 2016

B. Rosen
NeuStar, Inc.
H. Schulzrinne
Columbia U.
H. Tschofenig
ARM Limited
R. Gellens
March 1, 2016

Data-Only Emergency Calls
draft-ietf-ecrit-data-only-ea-11.txt

Abstract

RFC 6443 'Framework for Emergency Calling Using Internet Multimedia' describes how devices use the Internet to place emergency calls and how Public Safety Answering Points (PSAPs) handle Internet multimedia emergency calls natively. The exchange of multimedia traffic for emergency services involves a SIP session establishment starting with a SIP INVITE that negotiates various parameters for that session.

In some cases, however, the transmission of application data is all that is needed. Examples of such environments include alerts issued by a temperature sensor, burglar alarm, or chemical spill sensor. Often these alerts are conveyed as one-shot data transmissions. These type of interactions are called 'data-only emergency calls'. This document describes a container for the data based on the Common Alerting Protocol (CAP) and its transmission using the SIP MESSAGE transaction.

Status of This Memo

Table of Contents

1. Introduction

RFC 6443 [RFC6443] describes how devices use the Internet to place emergency calls and how Public Safety Answering Points (PSAPs) handle Internet multimedia emergency calls natively. The exchange of multimedia traffic for emergency services involves a SIP session establishment starting with a SIP INVITE that negotiates various parameters for that session.

In some cases, however, there is only application data to be conveyed from the end devices to a PSAP or an intermediary. Examples of such environments includes sensors issuing alerts, or vehicles sending crash data. These messages may be one-shot alerts to emergency authorities and do not require establishment of a session. These type of interactions are called ’data-only emergency calls’. In this document, we use the term "call" so that similarities between data-only (non-interactive) alerts and sessions with interactive media are more obvious.

Data-only emergency calls are similar to regular emergency calls in the sense that they require the emergency indications, emergency call routing functionality and may even have the same location requirements. However, the communication interaction will not lead to the exchange of interactive media, that is, Real-Time Protocol packets, such as voice, video data or real-time text.

The Common Alerting Protocol (CAP) [cap] is a document format for exchanging emergency alerts and public warnings. CAP is mainly used for conveying alerts and warnings between authorities and from authorities to citizen/individuals. This document is concerned with citizen to authority "alerts", where the alert is a call without any interactive media.

This document describes a method of including a CAP message in a SIP transaction, either by value (the CAP message is in the body of the message, using a CID) or by reference (a URI is included in the message, which when dereferenced returns the CAP message) by defining it as a block of "additional data" as defined in [I-D.ietf-ecrit-additional-data]. The additional data mechanism is also used to send alert specific data beyond that available in the CAP message. This document also describes how a SIP MESSAGE [RFC3428] transaction can be used to send a data-only call.

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. Architectural Overview

This section illustrates two envisioned usage modes: targeted and location-based emergency alert routing.

1. Emergency alerts containing only data are targeted to an intermediary recipient responsible for evaluating the next steps. These steps could include:

    1. Sending a non-interactive call containing only data toward a Public Safety Answering Point (PSAP);

    2. Establishing a third-party initiated emergency call towards a PSAP that could include audio, video, and data.

2. Emergency alerts may be targeted to a Service URN used for IP-based emergency calls where the recipient is not known to the originator. In this scenario, the alert may contain only data (e.g., a CAP, Geolocation header field and one or more Call-Info header fields containing Additional Data [I-D.ietf-ecrit-additional-data] in a SIP MESSAGE).

Figure 1 shows a deployment variant where a sensor is pre-configured (using techniques outside the scope of this document) to issue an alert to an aggregator that processes these messages and performs whatever steps are necessary to appropriately react to the alert. For example, a security firm may use different sensor inputs to dispatch their security staff to a building they protect or to initiate a third-party emergency call.

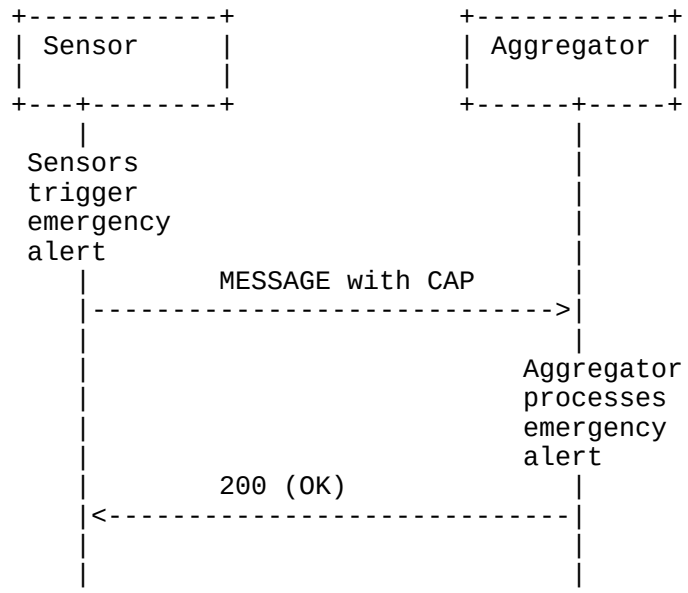

Figure 1: Targeted Emergency Alert Routing

In Figure 2 a scenario is shown whereby the alert is routed using location information and a Service URN. An emergency services routing proxy (ESRP) may use LoST to determine the next hop proxy to route the alert message to. A possible receiver is a PSAP and the recipient of the alert may be a call taker. In the generic case, there is very likely no prior relationship between the originator and the receiver, e.g., a PSAP. A PSAP, for example, is likely to receive and accept alerts from entities it cannot authorize. This scenario corresponds to the classic emergency services use case and the description in [RFC6881] is applicable. In this use case, the only difference between an emergency call and an emergency data-only call is that the former uses INVITE, creates a session, and negotiates one or more media streams, while the latter uses MESSAGE, does not create a session, and does not have media.

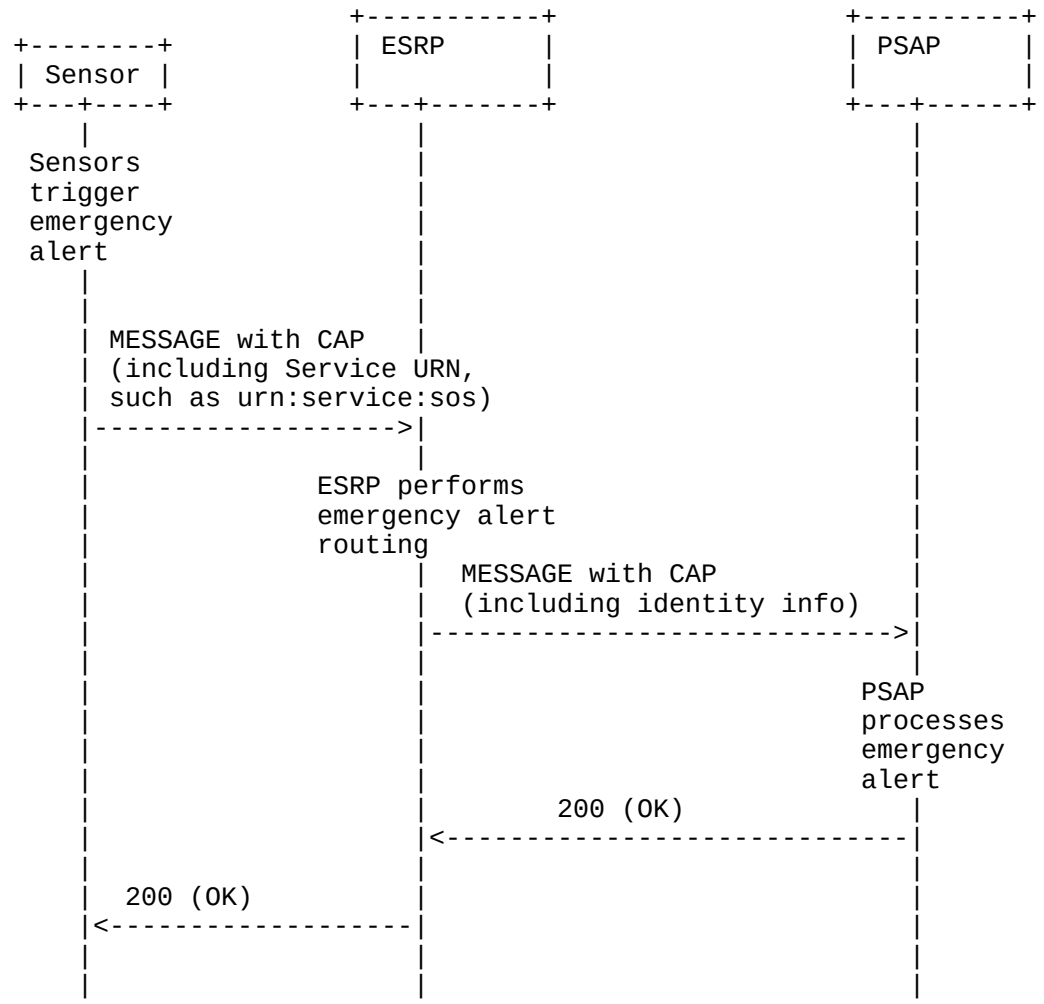

Figure 2: Location-Based Emergency Alert Routing

4. Protocol Specification

4.1. CAP Transport

A CAP message may be sent in the initial message of any SIP transaction. However, this document only addresses sending a CAP message in a SIP INVITE that initiates an emergency call, or in a SIP MESSAGE transaction for a one-shot, data-only emergency call. Behavior with other transactions is not defined.

The CAP message is included in a SIP message as an additional-data block [I-D.ietf-ecrit-additional-data]. Accordingly, it is introduced to the SIP message with a Call-Info header field with a purpose of "emergencyCall.cap". The header field may contain a URI that is used by the recipient (or in some cases, an intermediary) to obtain the CAP message. Alternative, the Call-Info header field may contain a Content Indirect url [RFC2392] and the CAP message included in the body of the message. In either case, the CAP message is located in a MIME block of the type 'application/ emergencyCall.cap+xml'.

If the SIP server does not support the functionality required to fulfill the request then a 501 Not Implemented MUST be returned as specified in RFC 3261 [RFC3261]. This is the appropriate response when a UAS does not recognize the request method and is not capable of supporting it for any user.

The 415 Unsupported Media Type error MUST be returned as specified in RFC 3261 [RFC3261] if the SIP server is refusing to service the request because the message body of the request is in a format not supported by the server for the requested method. The server MUST return a list of acceptable formats using the Accept, Accept-Encoding, or Accept-Language header fields, depending on the specific problem with the content.

4.2. Profiling of the CAP Document Content

The usage of CAP MUST conform to the specification provided with [cap]. For usage with SIP the following additional requirements are imposed:

sender: The following restrictions and conditions apply to setting the value of the <sender> element:

Originator is a SIP entity, Author indication irrelevant: When the alert was created by a SIP-based originator and it is not useful to be explicit about the author of the alert, then the <sender> element MUST be populated with the SIP URI of the user agent.

Originator is a non-SIP entity, Author indication irrelevant: When the alert was created by a non-SIP based entity and the identity of this original sender is to be preserved, then this identity MUST be placed into the <sender> element. In this situation it is not useful to be explicit about the author of the alert. The specific type of identity being used will depend on the technology used by the original originator.

Author indication relevant: When the author is different from the actual originator of the message and this distinction should be preserved, then the <sender> element MUST NOT contain the SIP URI of the user agent.

incidents: The <incidents> element MUST be present. This incident identifier MUST be chosen in such a way that it is unique for a given <sender, expires, incidents> combination. Note that the <expires> element is optional and may not be present.

scope: The value of the <scope> element MAY be set to "Private" if the alert is not meant for public consumption. The <addresses> element is, however, not used by this specification since the message routing is performed by SIP and the respective address information is already available in other SIP header fields. Populating information twice into different parts of the message may lead to inconsistency.

parameter: The <parameter> element MAY contain additional information specific to the sendor.

area: It is RECOMMENDED to omit this element when constructing a message. If the CAP message already contains an <area> element, then the specified location information SHOULD be copied into the PIDF-LO structure of the ’geolocation’ header field.

4.3. Sending a Data-Only Emergency Call

A data-only emergency call is sent using a SIP MESSAGE transaction with a CAP URI or body as described above in a manner similar to how an emergency call with interactive media is sent, as described in [RFC6881]. The MESSAGE transaction does not create a session nor send media, but otherwise, the header content of the transaction, routing, and processing of data-only calls are the same as those of other emergency calls.

5. Error Handling

This section defines a new error response code and a header field for additional information.

5.1. 425 (Bad Alert Message) Response Code

This SIP extension creates a new location-specific response code, defined as follows:

425 (Bad Alert Message)

The 425 response code is a rejection of the request due to its included alert content, indicating that it was malformed or not satisfactory for the recipient’s purpose.

A SIP intermediary can also reject an alert it receives from a UA when it understands that the provided alert is malformed.

Section 5.2 describes an AlertMsg-Error header field with more details about what was wrong with the alert message in the request. This header field MUST be included in the 425 response.

It is only appropriate to generate a 425 response when the responding entity has no other information in the request that is usable by the responder.

A 425 response code MUST NOT be sent in response to a request that lacks an alert message, as the user agent in that case may not support this extension.

A 425 response is a final response within a transaction, and MUST NOT terminate an existing dialog.

5.2. The AlertMsg-Error Header Field

The AlertMsg-Error header field provides additional information about what was wrong with the original request. In some cases the provided information will be used for debugging purposes.

The AlertMsg-Error header field has the following ABNF [RFC5234]:

```
message-header   /= AlertMsg-Error
                        ; (message-header from 3261)
AlertMsg-Error   = "AlertMsg-Error" HCOLON
                        ErrorValue
ErrorValue       =  error-code
                         *(SEMI error-params)
error-code       = 1*3DIGIT
error-params     = error-code-text
                         / generic-param ; from RFC3261
error-code-text  = "code" EQUAL quoted-string ; from RFC3261
```

HCOLON, SEMI, and EQUAL are defined in RFC3261 [RFC3261]. DIGIT is defined in RFC5234 [RFC5234].

The AlertMsg-Error header field MUST contain only one ErrorValue to indicate what was wrong with the alert payload the recipient determined was bad.

The ErrorValue contains a 3-digit error code indicating what was wrong with the alert in the request. This error code has a corresponding quoted error text string that is human understandable. The text string are OPTIONAL, but RECOMMENDED for human readability, similar to the string phrase used for SIP response codes. That said, the strings are complete enough for rendering to the user, if so desired. The strings in this document are recommendations, and are not standardized -- meaning an operator can change the strings -- but MUST NOT change the meaning of the error code. Similar to how RFC 3261 specifies, there MUST NOT be more than one string per error code.

The AlertMsg-Error header field MAY be included in any response if an alert message was in the request part of the same transaction. For example, a UA includes an alert in an MESSAGE to a PSAP. The PSAP can accept this MESSAGE, thus creating a dialog, even though its UA determined that the alert message contained in the MESSAGE was bad. The PSAP merely includes an AlertMsg-Error header field value in the 200 OK to the MESSAGE, thus informing the UA that the MESSAGE was accepted but the alert provided was bad.

If, on the other hand, the PSAP cannot accept the transaction without a suitable alert message, a 425 response is sent.

A SIP intermediary that requires the UA’s alert message in order to properly process the transaction may also sends a 425 with an AlertMsg-Error code.

This document defines an initial list of AlertMsg-Error values for any SIP response, including provisional responses (other than 100 Trying) and the new 425 response. There MUST be no more than one AlertMsg-Error code in a SIP response.

AlertMsg-Error: 100 ; code="Cannot Process the Alert Payload"

AlertMsg-Error: 101 ; code="Alert Payload was not present or could not be found"

AlertMsg-Error: 102 ; code="Not enough information to determine the purpose of the alert"

AlertMsg-Error: 103 ; code="Alert Payload was corrupted"

Additionally, if an entity cannot or chooses not to process the alert message from a SIP request, a 500 (Server Internal Error) SHOULD be used with or without a configurable Retry-After header field.

6. Updates to the CAP Message

If the sender anticipates that the content of the CAP message may need to be updated during the lifecycle of the event referred to in the message, it may include an update block as defined in [I-D.rosen-ecrit-addldata-subnot].If the sender includes an update block and does not have a globally reachable URI, then the UA must register it’s contact with a Registrar, and include a GRUU in in the update block

7. Call Backs

This document does not describe any method for the recipient to call back the sender of a data-only call. Usually, these alerts are sent by automata, which do not have a mechanism to receive calls of any kind. The identifier in the ’From’ header field may be useful to obtain more information, but any such mechanism is not defined in this document. The CAP message may contain related contact information for the sender.

8. Handling Large Amounts of Data

It is not atypical for sensors to have large quantities of data that they may wish to send. Including large amounts of data in a MESSAGE is not advisable, because SIP entities are usually not equipped to handle very large messages. In such cases, the sender SHOULD make use of the by-reference mechanisms defined in [I-D.ietf-ecrit-additional-data], which involves making the data available via HTTPS (either at the originator or at another entity), placing a URI to the data in the ’Call-Info’ header field, and the recipient using HTTPS to retrieve the data. The CAP message itself can be sent by-reference using this mechanism, as well as any or all of the Additional Data blocks that may contain sensor-specific data.

9. Example

Figure 3 shows a CAP document indicating a BURGLARY alert issued by a sensor called ’sensor1@domain.com’. The location of the sensor can be obtained from the attached location information provided via the ’geolocation’ header field contained in the SIP MESSAGE structure. Additionally, the sensor provided some data along with the alert message, using proprietary information elements intended only to be

processed by the receiver, a SIP entity acting as an aggregator. This example reflects the description in Figure 1.

```
MESSAGE sip:aggregator@domain.com SIP/2.0
Via: SIP/2.0/TCP sensor1.domain.com;branch=z9hG4bK776sgdkse
Max-Forwards: 70
From: sip:sensor1@domain.com;tag=49583
To: sip:aggregator@domain.com
Call-ID: asd88asd77a@1.2.3.4
Geolocation: <cid:abcdef@domain.com>
  ;routing-allowed=yes
Supported: geolocation
Accept: application/pidf+xml, application/emergencyCall.cap+xml
CSeq: 1 MESSAGE
Call-Info: cid:abcdef2@domain.com;purpose=emergencyCall.cap
Content-Type: multipart/mixed; boundary=boundary1
Content-Length: ...

--boundary1

Content-Type: application/emergencyCall.cap+xml
Content-ID: <abcdef2@domain.com>
Content-Disposition: by-reference;handling=optional
<?xml version="1.0" encoding="UTF-8"?>

<alert xmlns="urn:oasis:names:tc:emergency:cap:1.1">
  <identifier>S-1</identifier>
  <sender>sip:sensor1@domain.com</sender>
  <sent>2008-11-19T14:57:00-07:00</sent>
  <status>Actual</status>
  <msgType>Alert</msgType>
  <scope>Private</scope>
  <incidents>abc1234</incidents>
  <info>
      <category>Security</category>
      <event>BURGLARY</event>
      <urgency>Expected</urgency>
      <certainty>Likely</certainty>
      <severity>Moderate</severity>
      <senderName>SENSOR 1</senderName>
      <parameter>
        <valueName>SENSOR-DATA-NAMESPACE1</valueName>
        <value>123</value>
      </parameter>
      <parameter>
        <valueName>SENSOR-DATA-NAMESPACE2</valueName>
        <value>TRUE</value>
```

```
          </parameter>
       </info>
      </alert>

      --boundary1

      Content-Type: application/pidf+xml
      Content-ID: <abcdef2@domain.com>
      Content-Disposition: by-reference;handling=optional
      <?xml version="1.0" encoding="UTF-8"?>
          <presence
             xmlns="urn:ietf:params:xml:ns:pidf"
             xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
             xmlns:gbp=
                    "urn:ietf:params:xml:ns:pidf:geopriv10:basicPolicy"
             xmlns:cl="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr"
             xmlns:gml="http://www.opengis.net/gml"
             xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
             entity="pres:alice@atlanta.example.com">
           <dm:device id="sensor">
             <gp:geopriv>
               <gp:location-info>
                 <gml:location>
                   <gml:Point srsName="urn:ogc:def:crs:EPSG::4326">
                     <gml:pos>32.86726 -97.16054</gml:pos>
                   </gml:Point>
                </gml:location>
               </gp:location-info>
               <gp:usage-rules>
                 <gbp:retransmission-allowed>false
                 </gbp:retransmission-allowed>
                 <gbp:retention-expiry>2010-11-14T20:00:00Z
                 </gbp:retention-expiry>
               </gp:usage-rules>
               <gp:method>802.11</gp:method>
             </gp:geopriv>
             <dm:timestamp>2010-11-04T20:57:29Z</dm:timestamp>
           </dm:device>
         </presence>
      --boundary1--
```

Figure 3: Example Message conveying an Alert to an Aggregator

Figure 4 shows the same CAP document sent as a data-only emergency call towards a PSAP.

```
MESSAGE urn:service:sos SIP/2.0
```

```
Via: SIP/2.0/TCP sip:aggreg.1.example.com;branch=z9hG4bK776absa
Max-Forwards: 70
From: sip:aggregator@example.com;tag=32336
To: 112
Call-ID: asdf33443a@example.com
Route: sip:psap1.example.gov
Geolocation: <cid:abcdef@example.com>
  ;routing-allowed=yes
Supported: geolocation
Accept: application/pidf+xml, application/emergencyCall.cap+xml
Call-info: cid:abcdef2@domain.com;purpose=emergencyCall.cap
CSeq: 1 MESSAGE
Content-Type: multipart/mixed; boundary=boundary1
Content-Length: ...

--boundary1

Content-Type: application/emergencyCall.cap+xml
Content-ID: <abcdef2@example.com>
<?xml version="1.0" encoding="UTF-8"?>

<alert xmlns="urn:oasis:names:tc:emergency:cap:1.1">
  <identifier>S-1</identifier>
  <sender>sip:sensor1@domain.com</sender>
  <sent>2008-11-19T14:57:00-07:00</sent>
  <status>Actual</status>
  <msgType>Alert</msgType>
  <scope>Private</scope>
  <incidents>abc1234</incidents>
  <info>
      <category>Security</category>
      <event>BURGLARY</event>
      <urgency>Expected</urgency>
      <certainty>Likely</certainty>
      <severity>Moderate</severity>
      <senderName>SENSOR 1</senderName>
      <parameter>
        <valueName>SENSOR-DATA-NAMESPACE1</valueName>
        <value>123</value>
      </parameter>
      <parameter>
        <valueName>SENSOR-DATA-NAMESPACE2</valueName>
        <value>TRUE</value>
      </parameter>
  </info>
</alert>

--boundary1
```

```
Content-Type: application/pidf+xml
Content-ID: <abcdef2@domain.com>
<?xml version="1.0" encoding="UTF-8"?>
    <presence
       xmlns="urn:ietf:params:xml:ns:pidf"
       xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
       xmlns:gbp=
              "urn:ietf:params:xml:ns:pidf:geopriv10:basicPolicy"
       xmlns:cl="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr"
       xmlns:gml="http://www.opengis.net/gml"
       xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
       entity="pres:alice@atlanta.example.com">
     <dm:device id="sensor">
       <gp:geopriv>
         <gp:location-info>
           <gml:location>
             <gml:Point srsName="urn:ogc:def:crs:EPSG::4326">
               <gml:pos>32.86726 -97.16054</gml:pos>
             </gml:Point>
           </gml:location>
         </gp:location-info>
         <gp:usage-rules>
           <gbp:retransmission-allowed>false
           </gbp:retransmission-allowed>
           <gbp:retention-expiry>2010-11-14T20:00:00Z
           </gbp:retention-expiry>
         </gp:usage-rules>
         <gp:method>802.11</gp:method>
       </gp:geopriv>
       <dm:timestamp>2010-11-04T20:57:29Z</dm:timestamp>
     </dm:device>
   </presence>
--boundary1--
```

Figure 4: Example Message conveying an Alert to a PSAP

10. Security Considerations

This section discusses security considerations when SIP user agents issue emergency alerts utilizing MESSAGE and CAP. Location specific threats are not unique to this document and are discussed in [RFC7378] and [RFC6442].

The ECRIT emergency services architecture [RFC6443] considers classic individual-to-authority emergency calling where the identity of the emergency caller does not play a role at the time of the call establishment itself, i.e., a response to the emergency call does not depend on the identity of the caller. In the case of emergency

alerts generated by devices such as sensors, the processing may be different in order to reduce the number of falsely generated emergency alerts. Alerts may get triggered based on certain sensor input that may have been caused by factors other than the actual occurrence of an alert relevant event. For example, a sensor may simply be malfunctioning. For this reason, not all alert messages are directly sent to a PSAP, but rather may be pre-processed by a separate entity, potentially under supervision by a human, to filter alerts and potentially correlate received alerts with others to obtain a larger picture of the ongoing situation.

In any case, for alerts initiated by sensors, the identity may play an important role in deciding whether to accept or ignore an incoming alert message. With the scenario shown in Figure 1 it is very likely that only authorized sensor input will be processed. For this reason, it needs to be possible to refuse to accept alert messages from an unknown origin. Two types of information elements can be used for this purpose:

1. SIP itself provides security mechanisms that allow the verification of the originator's identity. These mechanisms can be re-used, such as P-Asserted-Identity [RFC3325] or SIP Identity [RFC4474]. The latter provides a cryptographic assurance while the former relies on a chain of trust model.

2. CAP provides additional security mechanisms and the ability to carry further information about the sender's identity. Section 3.3.2.1 of [cap] specifies the signing algorithms of CAP documents.

In addition to the desire to perform identity-based access control, the classic communication security threats need to be considered, including integrity protection to prevent forgery or replay of alert messages in transit. To deal with replay of alerts, a CAP document contains the mandatory <identifier>, <sender>, <sent> elements and an optional <expire> element. Together, these elements make the CAP document unique for a specific sender and provide time restrictions. An entity that has already received a CAP message 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. Additionally, it is RECOMMENDED to make use of SIP security mechanisms, such as SIP Identity [RFC4474], to tie the CAP message to the SIP message. To provide protection of the entire SIP message exchange between neighboring SIP entities, the usage of TLS is MANDATORY.

Note that none of the security mechanism in this document protect against a compromised sensor sending crafted alerts.

11. IANA Considerations

11.1. Registration of the 'application/emergencyCall.cap+xml' MIME type

To: ietf-types@iana.org

Subject: Registration of MIME media type application/emergencyCall.cap+xml

MIME media type name: application

MIME subtype name: cap+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). RFC XXX [Replace by the RFC number of this specification] discusses security considerations for this.

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 warnings according to the CAP standard.

Additional information: OASIS has published the Common Alerting Protocol at http://www.oasis-open.org/committees/documents.php&wg_abbrev=emergency

Person and email address to contact for further information: Hannes Tschofenig, hannes.tschofenig@gmx.net

Intended usage: Limited use

Author/Change controller: IETF ECRIT 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/cap+xml.

11.2. IANA Registration of 'cap' Additional Data Block

This document registers a new block type in the sub-registry called 'Additional Data Blocks' defined in [I-D.ietf-ecrit-additional-data]. The token is "cap" and the reference is this document.

11.3. IANA Registration for 425 Response Code

In the SIP Response Codes registry, the following is added

Reference: RFC-XXXX (i.e., this document)

Response code: 425 (recommended number to assign)

Default reason phrase: Bad Alert Message

Registry:

Response Code	Reference
Request Failure 4xx	
425 Bad Alert Message	[this doc]

This SIP Response code is defined in Section 5.

11.4. IANA Registration of New AlertMsg-Error Header Field

The SIP AlertMsg-error header field is created by this document, with its definition and rules in Section 5, to be added to the IANA Session Initiation Protocol (SIP) Parameters registry with two actions:

1. Update the Header Fields registry with

Registry: Header Name	compact	Reference
AlertMsg-Error		[this doc]

2. In the portion titled "Header Field Parameters and Parameter Values", add

Header Field	Parameter Name	Predefined Values	Reference
AlertMsg-Error	code	yes	[this doc]

11.5. IANA Registration for the SIP AlertMsg-Error Codes

This document creates a new registry for SIP, called "AlertMsg-Error Codes". AlertMsg-Error codes provide reasons for an error discovered by a recipient, categorized by the action to be taken by the error recipient. The initial values for this registry are shown below.

Registry Name: AlertMsg-Error Codes

Reference: [this doc]

Registration Procedures: Specification Required

Code	Default Reason Phrase	Reference
100	"Cannot Process the Alert Payload"	[this doc]
101	"Alert Payload was not present or could not be found"	[this doc]
102	"Not enough information to determine the purpose of the alert"	[this doc]
103	"Alert Payload was corrupted"	[this doc]

Details of these error codes are in Section 5.

12. Acknowledgments

The authors would like to thank the participants of the Early Warning adhoc meeting at IETF#69 for their feedback. Additionally, we would like to thank the members of the NENA Long Term Direction Working Group for their feedback.

Additionally, we would like to thank Martin Thomson, James Winterbottom, Shida Schubert, Bernard Aboba, and Marc Linsner for their review comments.

13. References

13.1. 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.

[RFC2392] Levinson, E., "Content-ID and Message-ID Uniform Resource Locators", RFC 2392, DOI 10.17487/RFC2392, August 1998, <http://www.rfc-editor.org/info/rfc2392>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <http://www.rfc-editor.org/info/rfc3261>.

[RFC3428] Campbell, B., Ed., Rosenberg, J., Schulzrinne, H., Huitema, C., and D. Gurle, "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC 3428, DOI 10.17487/RFC3428, December 2002, <http://www.rfc-editor.org/info/rfc3428>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <http://www.rfc-editor.org/info/rfc5234>.

[RFC3023] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", RFC 3023, DOI 10.17487/RFC3023, January 2001, <http://www.rfc-editor.org/info/rfc3023>.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <http://www.rfc-editor.org/info/rfc3629>.

[RFC6442] Polk, J., Rosen, B., and J. Peterson, "Location Conveyance for the Session Initiation Protocol", RFC 6442, DOI 10.17487/RFC6442, December 2011, <http://www.rfc-editor.org/info/rfc6442>.

[RFC6881] Rosen, B. and J. Polk, "Best Current Practice for Communications Services in Support of Emergency Calling", BCP 181, RFC 6881, DOI 10.17487/RFC6881, March 2013, <http://www.rfc-editor.org/info/rfc6881>.

[I-D.ietf-ecrit-additional-data] Gellens, R., Rosen, B., Tschofenig, H., Marshall, R., and J. Winterbottom, "Additional Data Related to an Emergency Call", draft-ietf-ecrit-additional-data-37 (work in progress), October 2015.

[I-D.rosen-ecrit-addldata-subnot] Rosen, B., "Updating Additional Data related to an Emergency Call using Subscribe/ Notify", draft-rosen-ecrit-addldata-subnot-01 (work in progress), November 2013.

13.2. Informative References

[RFC7378] Tschofenig, H., Schulzrinne, H., and B. Aboba, Ed., "Trustworthy Location", RFC 7378, DOI 10.17487/RFC7378, December 2014, <http://www.rfc-editor.org/info/rfc7378>.

[RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, DOI 10.17487/RFC4474, August 2006, <http://www.rfc-editor.org/info/rfc4474>.

[RFC3325] Jennings, C., Peterson, J., and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, DOI 10.17487/RFC3325, November 2002, <http://www.rfc-editor.org/info/rfc3325>.

[RFC6443] Rosen, B., Schulzrinne, H., Polk, J., and A. Newton, "Framework for Emergency Calling Using Internet Multimedia", RFC 6443, DOI 10.17487/RFC6443, December 2011, <http://www.rfc-editor.org/info/rfc6443>.

Authors' Addresses

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

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
ARM Limited
Austria

Email: Hannes.Tschofenig@gmx.net
URI: http://www.tschofenig.priv.at

Randall Gellens

Email: rg+ietf@randy.pensive.org