ï»¿

Internet Engineering Task Force (IETF)
Request for Comments: 8876
Category: Standards Track
ISSN: 2070-1721

B. Rosen

H. Schulzrinne
Columbia U.
H. Tschofenig

R. Gellens
Core Technology Consulting

September 2020

Non-interactive Emergency Calls

Abstract

Use of the Internet for emergency calling is described in RFC 6443, 'Framework for Emergency Calling Using Internet Multimedia'. In some cases of emergency calls, the transmission of application data is all that is needed, and no interactive media channel is established: a situation referred to as 'non-interactive emergency calls', where, unlike most emergency calls, there is no two-way interactive media such as voice or video or text. This document describes use of a SIP MESSAGE transaction that includes a container for the data based on the Common Alerting Protocol (CAP). That type of emergency request does not establish a session, distinguishing it from SIP INVITE, which does. Any device that needs to initiate a request for emergency services without an interactive media channel would use the mechanisms in this document.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8876.

Table of Contents

1. Introduction

[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 include sensors issuing alerts, and certain types of medical monitors. These messages may be alerts to emergency authorities and do not require establishment of a session. These types of interactions are called 'non-interactive emergency calls'. In this document, we use the term "call" so that similarities between non-interactive alerts and sessions with interactive media are more obvious.

Non-interactive emergency calls are similar to regular emergency calls in the sense that they require the emergency indications, emergency call routing functionality, and location. However, the communication interaction will not lead to the exchange of interactive media, that is, Real-Time Transport Protocol [RFC3550] packets, such as voice, video, or real-time text.

The Common Alerting Protocol (CAP) [CAP] is a format for exchanging emergency alerts and public warnings. CAP is mainly used for conveying alerts and warnings between authorities and from authorities to the public. The scope of this document is conveying CAP alerts from private devices to emergency service authorities, as a call without any interactive media.

This document describes a method of including a CAP alert in a SIP transaction by defining it as a block of "additional data" as defined in [RFC7852]. The CAP alert is included either by value (the CAP alert is in the body of the message, using a CID) or by reference (the message includes a URI that, when dereferenced, returns the CAP alert). The additional data mechanism is also used to send alert-specific data beyond that available in the CAP alert. This document also describes how a SIP MESSAGE [RFC3428] transaction can be used to send a non-interactive call.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Non-interactive emergency call: An emergency call where there is no

two-way interactive media

SIP: Session Initiation Protocol [RFC3261]

PIDF-LO: Presence Information Data Format Location Object, a data structure for carrying location [RFC4119]

LoST: Location To Service Translation protocol [RFC5222]

CID: Content-ID [RFC2392]

CAP: Common Alerting Protocol [CAP]

PSAP: Public Safety Answering Point, the call center for emergency calls

ESRP: Emergency Services Routing Proxy, a type of SIP Proxy Server used in some emergency services networks

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:

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

   b. 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 [RFC5031] 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 SIP MESSAGE with CAP content, a Geolocation header field, and one or more Call-Info header fields containing additional data [RFC7852]).

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.

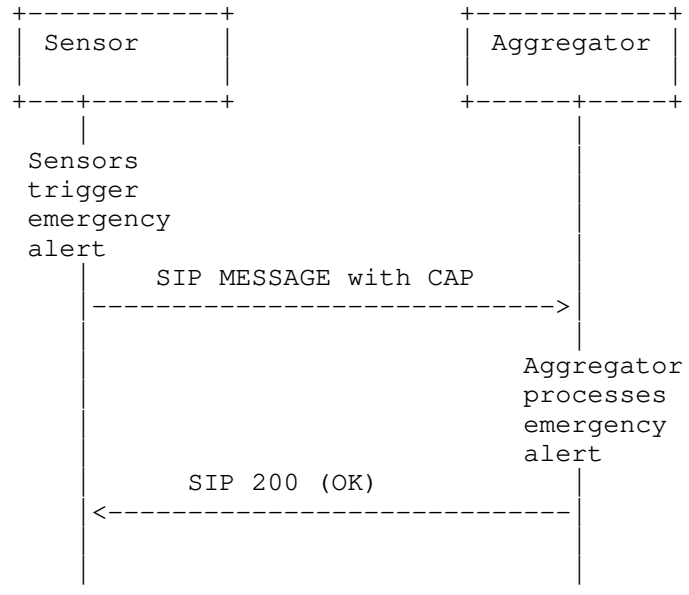

Figure 1: Targeted Emergency Alert Routing

In Figure 2, a scenario is shown where the alert is routed using location information and a service URN. An emergency services routing proxy (ESRP) may use LoST (a protocol defined by [RFC5222], which translates a location to a URI used to route an emergency call) 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. For example, a PSAP is likely to receive and accept alerts from entities it has no previous relationship with. This scenario is similar to a classic voice emergency services call, and the description in [RFC6881] is applicable. In this use case, the only difference between an emergency call and an emergency non-interactive 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 interactive media.

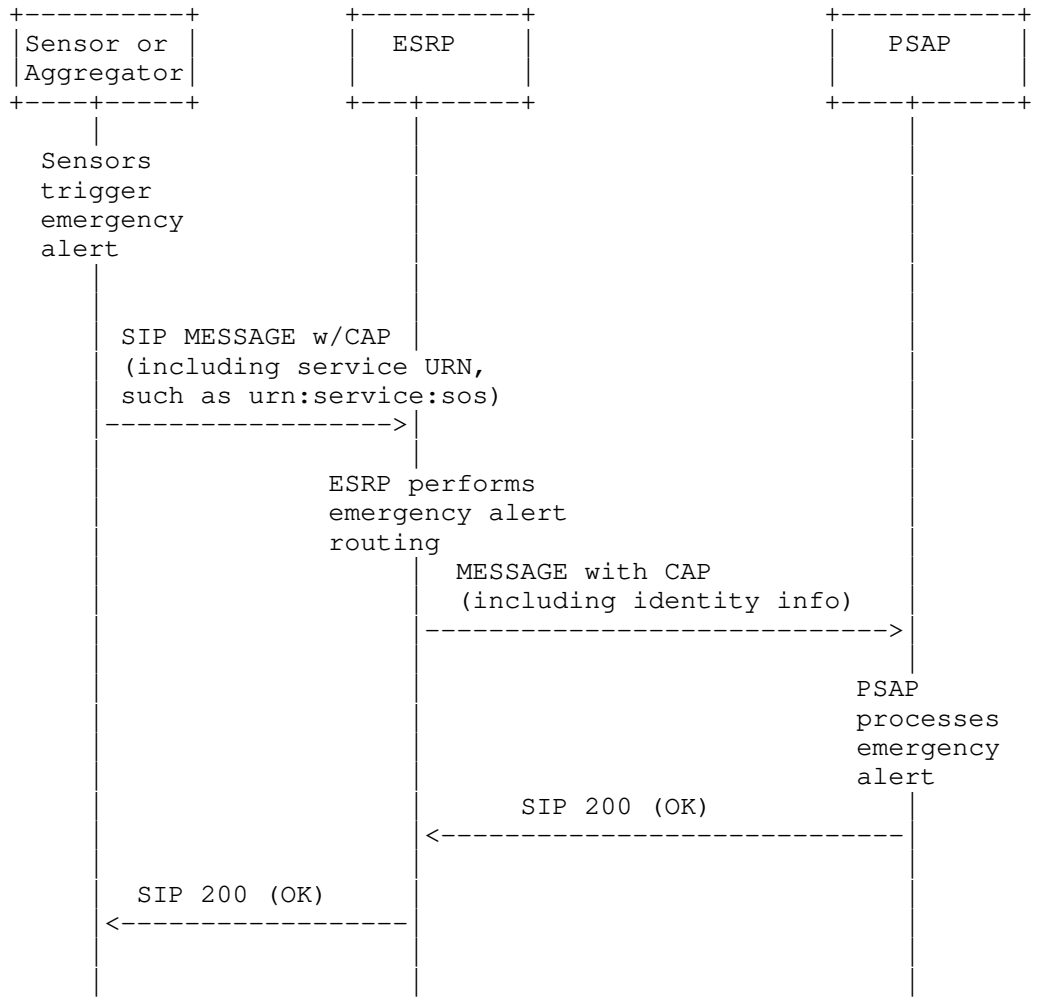

Figure 2: Location-Based Emergency Alert Routing

4. Protocol Specification

4.1. CAP Transport

This document addresses sending a CAP alert in a SIP MESSAGE transaction for a non-interactive emergency call. Behavior with other transactions is not defined.

The CAP alert is included in a SIP message as an additional data block [RFC7852]. Accordingly, it is conveyed in the SIP message with a Call-Info header field with a purpose of "EmergencyCallData.cap". The header field may contain a URI that is used by the recipient (or in some cases, an intermediary) to obtain the CAP alert. Alternatively, the Call-Info header field may contain a Content-ID

URL [RFC2392] and the CAP alert included in the body of the message. In the latter case, the CAP alert is located in a MIME block of the type 'application/emergencyCallData.cap+xml'.

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

The 415 Unsupported Media Type error will be returned as specified in [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 (where "sender" and "author" are as defined in CAP and "originator" is the entity sending the CAP alert, which may be different from the entity sending the SIP MESSAGE):

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 originator.

* Author indication relevant: When the author is different from the 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 might 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 sender, conforming to the CAP alert syntax.

area: It is RECOMMENDED to omit this element when constructing a message. If the CAP alert is given to the SIP entity to transport and it already contains an <area> element, then the specified location information SHOULD be copied into a PIDF-LO structure (the data format for location used by emergency calls on the Internet) referenced by the SIP 'Geolocation' header field. If

the CAP alert is being created by the SIP entity using a PIDF-LO structure referenced by 'geolocation' to construct <area>, implementers must be aware that <area> is limited to a circle or polygon, and conversion of other shapes will be required. Points SHOULD be converted to a circle with a radius equal to the uncertainty of the point. Arc-bands and ellipses SHOULD be converted to polygons with similar coverage, and 3D locations SHOULD be converted to 2D forms with similar coverage.

4.3. Sending a Non-interactive Emergency Call

A non-interactive emergency call is sent using a SIP MESSAGE transaction with a CAP URI or body part 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 establish interactive media streams, but otherwise, the header content of the transaction, routing, and processing of non-interactive 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 response code defined as follows:

425 (Bad Alert Message)

The 425 response code is a rejection of the request, indicating that it was malformed enough that no reasonable emergency response to the alert can be determined.

A SIP intermediary can also use this code to reject an alert it receives from a User Agent (UA) when it detects 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 usually the case that emergency calls are not rejected if there is any useful information that can be acted upon. 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 (i.e., CAP data), 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 RFC 3261)
AlertMsg-Error   = "AlertMsg-Error" HCOLON
                         ErrorValue
ErrorValue       =  error-code
                          *(SEMI error-params)
error-code       = 3DIGIT
error-params     = error-code-text
```

```
                              / generic-param ; from RFC 3261
   error-code-text  = "message" EQUAL quoted-string ; from RFC 3261
```

HCOLON, SEMI, and EQUAL are defined in [RFC3261]. DIGIT is defined in [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 readable. The text string is OPTIONAL, but RECOMMENDED for human readability, similar to the string phrase used for SIP response codes. 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. The code space for ErrorValue is separate from SIP Status Codes.

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, suppose a UA includes an alert in a MESSAGE to a PSAP. The PSAP can accept this MESSAGE, 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 send a 425 response 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 NOT be more than one AlertMsg-Error code in a SIP response. AlertMsg-Error values sent in provisional responses MUST be sent using the mechanism defined in [RFC3262]; or, if that mechanism is not negotiated, they MUST be repeated in the final response to the transaction.

AlertMsg-Error: 100 ; message="Cannot process the alert payload"

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

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

AlertMsg-Error: 103 ; message="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. Call Backs

This document does not describe any method for the recipient to call back the sender of a non-interactive 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 alert may contain related contact information for the sender.

7. Handling Large Amounts of Data

Sensors may have large quantities of data that they may wish to send.

Including large amounts of data (tens of kilobytes) 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 [RFC7852], which involves making the data available via HTTPS [RFC2818] (either at the originator or at another entity), placing a URI to the data in the 'Call-Info' header field, and the recipient uses HTTPS to retrieve the data. The CAP alert itself can be sent by reference using this mechanism, as can any or all of the additional data blocks that may contain sensor-specific data.

There are no rate-limiting mechanisms for any SIP transactions that are standardized, although implementations often include such functions. Non-interactive emergency calls are typically handled the same as any emergency call, which means a human call-taker is involved. Implementations should take note of this limitation, especially when calls are placed automatically without human initiation.

8. Example

The following example shows a CAP document indicating a BURGLARY alert issued by a sensor called 'sensor1@example.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.

```
MESSAGE sip:aggregator@example.com SIP/2.0
Via: SIP/2.0/TCP sensor1.example.com;branch=z9hG4bK776sgdkse
Max-Forwards: 70
From: sip:sensor1@example.com;tag=49583
To: sip:aggregator@example.com
Call-ID: asd88asd77a@2001:db8::ff
Geolocation: <cid:abcdef@example.com>
  ;routing-allowed=yes
Supported: geolocation
CSeq: 1 MESSAGE
Call-Info: cid:abcdef2@example.com;purpose=EmergencyCallData.cap
Content-Type: multipart/mixed; boundary=boundary1
Content-Length: ...

--boundary1
Content-Type: application/EmergencyCallData.cap+xml
Content-ID: <abcdef2@example.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@example.com</sender>
 <sent>2020-01-04T20:57:35Z</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@example.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>44.85249659 -93.238665712</gml:pos>
                </gml:Point>
              </gml:location>
            </gp:location-info>
            <gp:usage-rules>
              <gbp:retransmission-allowed>false
              </gbp:retransmission-allowed>
              <gbp:retention-expiry>2020-02-04T20:57:29Z
              </gbp:retention-expiry>
            </gp:usage-rules>
            <gp:method>802.11</gp:method>
          </gp:geopriv>
          <dm:timestamp>2020-01-04T20:57:29Z</dm:timestamp>
        </dm:device>
       </presence>
   --boundary1--
```

Figure 3: Example Message Conveying an Alert to an Aggregator

The following shows the same CAP document sent as a non-interactive emergency call towards a PSAP.

```
   MESSAGE urn:service:sos SIP/2.0
   Via: SIP/2.0/TCP sip:aggreg.1.example.com;branch=z9hG4bK776abssa
   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
   Call-info: cid:abcdef2@example.com;purpose=EmergencyCallData.cap
   CSeq: 1 MESSAGE
   Content-Type: multipart/mixed; boundary=boundary1
   Content-Length: ...

   --boundary1

   Content-Type: application/EmergencyCallData.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@example.com</sender>
```

```
     <sent>2020-01-04T20:57:35Z</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@example.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>44.85249659 -93.2386657124</gml:pos>
                 </gml:Point>
               </gml:location>
             </gp:location-info>
             <gp:usage-rules>
               <gbp:retransmission-allowed>false
               </gbp:retransmission-allowed>
               <gbp:retention-expiry>2020-02-04T20:57:25Z
               </gbp:retention-expiry>
             </gp:usage-rules>
             <gp:method>802.11</gp:method>
           </gp:geopriv>
           <dm:timestamp>2020-01-04T20:57:25Z</dm:timestamp>
         </dm:device>
       </presence>
    --boundary1--
```

Figure 4: Example Message Conveying an Alert to a PSAP

9. 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 Emergency Context Resolution with Internet Technologies (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 could get triggered based on certain sensor input that might 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 could 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 authenticated sensor input will be processed. For this reason, it needs to be possible to refuse to accept alert messages from unknown origins. 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, such as P-Asserted-Identity [RFC3325] or SIP Identity [RFC8224]. The latter provides a cryptographic assurance while the former relies on a chain-of-trust model. These mechanisms can be reused.

2. CAP provides additional security mechanisms and the ability to carry further information about the sender's identity. Section 3.3.4.1 of [CAP] specifies the signing algorithms of CAP documents.

The specific policy and mechanisms used in a given deployment are out of scope for this document.

There is no rate limiting mechanisms in SIP, and all kinds of emergency calls, including those defined in this document, could be used by malicious actors or misbehaving devices to effect a denial-of-service attack on the emergency services. The mechanism defined in this document does not introduce any new considerations, although it may be more likely that devices that place non-interactive emergency calls without a human initiating them may be more likely than those that require a user to initiate them.

Implementors should note that automated emergency calls may be prohibited or regulated in some jurisdictions, and there may be penalties for "false positive" calls.

This document describes potential retrieval of information by dereferencing URIs found in a Call Info header of a SIP MESSAGE. These may include a CAP alert as well as other additional data [RFC7852] blocks. The domain of the device sending the SIP MESSAGE; the domain of the server holding the CAP alert, if sent by reference; and the domain of other additional data blocks, if sent by reference, may all be different. No assumptions can be made that there are trust relationships between these entities. Recipients MUST take precautions in retrieving any additional data blocks passed by reference, including the CAP alert, because the URI may point to a malicious actor or entity not expecting to be referred to for this purpose. The considerations in handling URIs in [RFC3986] apply.

Use of timestamps to prevent replay is subject to the availability of accurate time at all participants. Because emergency event notification via this mechanism is relatively low frequency and generally involves human interaction, implementations may wish to consider messages with times within a small number of seconds of each other to be effectively simultaneous for the purposes of detecting replay. Implementations may also wish to consider that most deployed time distribution protocols likely to be used by these systems are not presently secure.

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>, and <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 alert 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 the SIP Identity PASSporT [RFC8225], to tie the CAP alert to the SIP message. To provide protection of the entire SIP message exchange between neighboring SIP entities, the usage of TLS is RECOMMENDED. [RFC6443] discusses the issues of using TLS with emergency calls, which are equally applicable to non-interactive emergency calls.

Note that none of the security mechanisms in this document protect against a compromised sensor sending crafted alerts. Confidentiality provided for any emergency calls, including non-interactive messages, is subject to local regulations. Privacy issues are discussed in [RFC7852] and are applicable here.

10. IANA Considerations

10.1. 'application/EmergencyCallData.cap+xml' Media Type

Type name: application

Subtype name: EmergencyCallData.cap+xml

Required parameters: N/A

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

Encoding considerations: 7bit, 8bit, or binary. See Section 3.2 of [RFC7303].

Security considerations: This content type is designed to carry payloads of the Common Alerting Protocol (CAP). RFC 8876 discusses security considerations for this.

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

Published specification: RFC 8876

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

Fragment identifier considerations: N/A

Additional information: OASIS has published the Common Alerting Protocol at <https://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.pdf>

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

Intended usage: Limited use

Author/Change controller: The IESG

Other information: This media type is a specialization of 'application/xml' [RFC7303], and many of the considerations described there also apply to application/EmergencyCallData.cap+xml.

10.2. 'cap' Additional Data Block

Per this document, IANA has registered a new block type in the "Emergency Call Data Types" subregistry of the "Emergency Call Additional Data" registry defined in [RFC7852]. The token is "cap", the Data About is "The Call", and the reference is this document.

10.3. 425 Response Code

In the SIP "Response Codes" registry, the following has been added under Request Failure 4xx.

Response Code	Description	Reference
425	Bad Alert Message	RFC 8876

Table 1: Response Codes Registry Addition

This SIP Response code is defined in Section 5.

10.4. AlertMsg-Error Header Field

The SIP AlertMsg-Error header field is created by this document, with its definition and rules in Section 5. The IANA "Session Initiation Protocol (SIP) Parameters" registry has been updated as follows.

1. In the "Header Fields" subregistry, the following has been added:

Head Name	compact	Reference
AlertMsg-Error		RFC 8876

Table 2: Header Fields Registry Addition

2. In the "Header Field Parameters and Parameter Values" subregistry, the following has been added:

Header Field	Parameter Name	Predefined Values	Reference
AlertMsg-Error	code	no	RFC 8876

Table 3: Header Field Parameters and Parameter Values Registry Addition

10.5. SIP AlertMsg-Error Codes

This document creates a new registry called "SIP 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. The registration procedure is Specification Required [RFC8126].

Code	Default Reason Phrase	Reference
100	"Cannot process the alert payload"	RFC 8876
101	"Alert payload was not present or could not be found"	RFC 8876
102	"Not enough information to determine the purpose of the alert"	RFC 8876
103	"Alert payload was corrupted"	RFC 8876

Table 4: SIP AlertMsg-Error Codes Registry Creation

Details of these error codes are in Section 5.

11. References

11.1. Normative References

[CAP] Jones, E. and A. Botterell, "Common Alerting Protocol Version 1.2", OASIS Standard CAP-V1.2, July 2010, <https://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.pdf>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

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

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <https://www.rfc-editor.org/info/rfc2818>.

[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, <https://www.rfc-editor.org/info/rfc3261>.

[RFC3262] Rosenberg, J. and H. Schulzrinne, "Reliability of Provisional Responses in Session Initiation Protocol (SIP)", RFC 3262, DOI 10.17487/RFC3262, June 2002, <https://www.rfc-editor.org/info/rfc3262>.

[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, <https://www.rfc-editor.org/info/rfc3428>.

[RFC4119] Peterson, J., "A Presence-based GEOPRIV Location Object Format", RFC 4119, DOI 10.17487/RFC4119, December 2005, <https://www.rfc-editor.org/info/rfc4119>.

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

[RFC7303] Thompson, H. and C. Lilley, "XML Media Types", RFC 7303, DOI 10.17487/RFC7303, July 2014, <https://www.rfc-editor.org/info/rfc7303>.

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

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[RFC6442] Polk, J., Rosen, B., and J. Peterson, "Location Conveyance for the Session Initiation Protocol", RFC 6442, DOI 10.17487/RFC6442, December 2011, <https://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, <https://www.rfc-editor.org/info/rfc6881>.

[RFC7852] Gellens, R., Rosen, B., Tschofenig, H., Marshall, R., and J. Winterbottom, "Additional Data Related to an Emergency Call", RFC 7852, DOI 10.17487/RFC7852, July 2016, <https://www.rfc-editor.org/info/rfc7852>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8225] Wendt, C. and J. Peterson, "PASSporT: Personal Assertion Token", RFC 8225, DOI 10.17487/RFC8225, February 2018, <https://www.rfc-editor.org/info/rfc8225>.

11.2. Informative References

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

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8224] Peterson, J., Jennings, C., Rescorla, E., and C. Wendt, "Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 8224, DOI 10.17487/RFC8224, February 2018, <https://www.rfc-editor.org/info/rfc8224>.

[RFC5031] Schulzrinne, H., "A Uniform Resource Name (URN) for Emergency and Other Well-Known Services", RFC 5031, DOI 10.17487/RFC5031, January 2008, <https://www.rfc-editor.org/info/rfc5031>.

[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, <https://www.rfc-editor.org/info/rfc3325>.

[RFC5222] Hardie, T., Newton, A., Schulzrinne, H., and H. Tschofenig, "LoST: A Location-to-Service Translation Protocol", RFC 5222, DOI 10.17487/RFC5222, August 2008, <https://www.rfc-editor.org/info/rfc5222>.

[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, <https://www.rfc-editor.org/info/rfc6443>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <https://www.rfc-editor.org/info/rfc3550>.

Acknowledgments

The authors would like to thank the participants of the Early Warning ad hoc 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, Marc Linsner, Christer

Holmberg, and Ivo Sedlacek for their review comments.

Authors' Addresses

Brian Rosen
470 Conrad Dr
Mars, PA 16046
United States of America

Email: br@brianrosen.net

Henning Schulzrinne
Columbia University
Department of Computer Science
450 Computer Science Building
New York, NY 10027
United States of America

Phone: +1 212 939 7004
Email: hgs+ecrit@cs.columbia.edu
URI: https://www.cs.columbia.edu

Hannes Tschofenig
Austria

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

Randall Gellens
Core Technology Consulting

Email: rg+ietf@coretechnologyconsulting.com
URI: http://www.coretechnologyconsulting.com