draft-urpalainen-sip-xcap-diff-event-03

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on May 19, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

This document describes an "xcap-diff" SIP event package, with the aid of which clients can receive notifications of the partial changes of Extensible Markup Language (XML) Configuration Access Protocol (XCAP) resources. The initial synchronization and document updates are based on using the XCAP-Diff format.
1. Introduction


While XCAP allows several authorized users or devices to modify the same XML document, XCAP does not provide an effective synchronization mechanism (except polling) to keep resources equivalent between the server and the client. This memo defines an "xcap-diff" event package that, together with the SIP event notification framework [RFC3265] and the XCAP-diff format [I-D.ietf-simple-xcap-diff], allows a user to subscribe to changes in an XML document, and to receive notifications whenever a change in an XML document takes place. It is also possible to subscribe to documents within some collection, the notifier provides those documents where the user has read privilege.

Before being able to apply any patches into documents, a client needs to first retrieve initial reference documents. With XCAP, this is done with the HTTP [RFC2616] protocol. The first "xcap-diff" notification thus contains references to subscribed documents, thereafter notifications MAY contain patches to these documents. Optionally some clients MAY want to retrieve only specific element or attribute content from these XCAP documents. All this information can be transformed with the XCAP-Diff [I-D.ietf-simple-xcap-diff] format.

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, BCP 14 [RFC2119] and indicate requirement levels for compliant implementations.

3. Definitions

The following terms are used in this document:
XCAP Component: An XML element or an attribute, which can be updated with the XCAP protocol.

Aggregating: While XCAP clients update only single XCAP components at a time, several of these modifications can be aggregated together with the XML-Patch-Ops semantics.

4. XCAP-Diff Event Package

4.1. XCAP-Diff Processing

When a client starts the "xcap-diff" subscription they may not be aware of all the XCAP resources they are subscribing to. This may for instance happen when the user subscribes to his/her collection [RFC4918] of a given XCAP Application Usage. The initial notification will then contain references to documents the user is authorized to read. After receiving these documents and resolving the initial synchronization stage, the subsequent notifications MAY contain patches to these documents.

While the initial document synchronization is based on separate HTTP retrievals of full documents, XML elements or attributes may be received "in-band", that is within the <xcap-diff> notification format. For example, the subscribed element has different semantics than XML-Patch-Ops [I-D.ietf-simple-xml-patch-ops] elements, because the intent is just to get the content of an element without the need of a separate HTTP GET request. If during the initial subscription stage the requested node can not be found, there is not any need for a result element or an attribute either. Once they will be created, the notify will contain their content. And similarly, their removals will be reported, for example after a successful HTTP DELETE request.

In another usage scenario, a subscriber to the "xcap-diff" event might not need XML-Patch operations at all. Indeed, the subscriber just wants to be informed that an update has happened, but the subscriber is not interested in learning the actual changes in the document(s). The XCAP-Diff [I-D.ietf-simple-xcap-diff] format will then only indicate document creations, updates and removals.

4.2. Event Package Name

The name of this event package is "xcap-diff". This package name is carried in the Event and Allow-Events header, as defined in RFC3265 [RFC3265].

Urpalainen                Expires May 19, 2008                  [Page 4]
4.3. ’diff-processing’ Event Package Parameter

The optional "diff-processing" parameter will tell the notifier how to apply the "XML differencing process". The possible values are "no-patching", "xcap-patching", "aggregate". The "no-patching" value means that only document or XCAP component existences MUST be indicated. The "xcap-patching" value means that all individual XCAP component updates with ETag changes MUST be indicated. The "aggregate" value means that the notifier is free to aggregate several individual XCAP component updates into a single XCAP-Diff <document> element. If the subscription does not contain this additional "diff-processing" parameter, the notifier MUST send all individual changes so that the client receives the full ETag change history of a document. In other words, "xcap-patching" is the default mode.

The formal grammar ([RFC4234](https://datatracker.ietf.org/doc/rfc4234)) of the "diff-processing" parameter:

diff-processing = "diff-processing" EQUALS
               "aggregate" / "no-patching" /
               "xcap-patching" / token

4.4. SUBSCRIBE Bodies

When generating the subscribe request, the subscriber needs to define the resources it is interested in getting information. This can be done simply by sending a URI list to the SIP notifier. This list can be described with the XCAP resource list ([RFC4826](https://datatracker.ietf.org/doc/rfc4826)) document format. Only a simple subset of that format is needed here, i.e. a flat list of XCAP URIs. The usage of hierarchical lists and <entry> references, etc. can thus be discarded. However, using this format allows adding some future semantics to these subscriptions. Also it is anticipated that the XCAP server will be collocated with the SIP notifier so relative XCAP URIs MAY be used. Note that although the node-selector of XCAP allows requesting all in-scope namespaces of an element, it is disallowed to subscribe to them with this package.
Figure 1 shows an example to a subscription of several XCAP resources: a "resource-list" document, a specific element in a "rls-services" document and a collection in "pidf-manipulation" Application Usage. For all of these resources the client MUST have read privilege in order to actually receive them in a NOTIFY request. The "Content-Type" header of this SUBSCRIBE request is "application/resource-lists+xml".

```xml
<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists">
  <list>
    <entry uri="resource-lists/users/sip:joe@example.com/index"/>
    <entry uri="rls-services/users/sip:joe@example.com/index~/*/service%5b@uri='sip:marketing@example.com'/%5d"/>
    <entry uri="pidf-manipulation"/>
  </list>
</resource-lists>
```

Figure 1: Example subscription body

Note that when collections are being selected as "pidf-manipulation" in the example, the conventions applied follow the WebDAV [RFC4918] semantics, that is, the subscriber MUST add the forward slash "/" character to the end of the path segment.

4.5. Subscription Duration

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

4.6. NOTIFY Bodies

The NOTIFY bodies will follow the conventions of the XCAP-Diff format [I-D.ietf-simple-xcap-diff].

4.7. Notifier Generation of NOTIFY Requests

During the initial subscription the notifier must first resolve the requested XCAP resources. Once there are superfluous resource selections in the requested URI list, the notifier SHOULD not provide overlapping similar responses for these resources. Only the resources where the authenticated user has read privilege will be included in the XCAP-Diff format. Note that for example, an XCAP component which could not be located with XCAP semantics, does not produce an error. Instead, the request remains in "pending" state,
that is, waiting for this resource to be created. Subscriptions to collections have a similar property: once a new document is created into the subscribed collection, the creation of a new resource is notified with the next NOTIFY request. After the authorized XCAP resources are known, the notifier will generate the first full response with the list of found and authorized resources. At this time depending on the "diff-processing" parameter, the notifier typically starts also the follow-up of XCAP component updates and unless otherwise directed, it reports all individual XCAP component updates with ETag changes to the subscriber. If the notifier process receives then a re-subscription with the diff-processing=aggregate value it MUST re-send the current full XML-Diff content unless the request or body can be suppressed with the SubNot-Etags [I-D.ietf-sip-subnot-etags] semantics by using e.g. the header Suppress-Notify-If-Match: xxxx. The notifier SHOULD then start to aggregate "diff-processing".

It MAY happen in some corner cases that the notifier can not or will not provide patches with the XML-Patch-Ops [I-D.ietf-simple-xml-patch-ops] semantics. One example of this is when the diff format produces a larger content than the original document is. When this happens, and if the server has been in this diff "aggregation" mode, it MUST fall back to the "xcap-patching" mode for this particular resource.

It should be noted that the whole diff-processing is truly implementation specific and in essence, also OPTIONAL. If the server so desires, it MAY elect not to produce XML-diffing at all although it is RECOMMENDED to implement it. Support for XCAP component subscriptions is mandatory in the server, however. Note that if for example, the subscriber has selected too many elements to subscribe, so that the notification body becomes impractically large, the notifier MAY discard the element content. The existence of elements is indicated with an empty element but the content is not shown for those resources.

Event packages like this require in practice a reliable transfer of events. This means that all events MUST be successfully transferred as otherwise patching will most likely fail or at least the document content becomes to be different. RFC 3265 [RFC3265] proposes utilization of a "version" attribute information to state deltas in chapter 4.4. Partial-PIDF-Notify [I-D.ietf-simple-partial-notify] requires that notifiers will not send a new request to the same dialog unless a successful response (200 OK) has been received for the last request. The latter applies also to this "xcap-diff" event package. If the NOTIFY request fails due to a timeout condition, the notifier MUST remove the subscription.
4.8. Subscriber Processing of NOTIFY Requests

The first NOTIFY request will typically contain references to HTTP resources including their strong ETag values. If the subscriber does not have similar locally cached versions, it will start an unconditional HTTP GET request for those resources. During this HTTP retrieval time it MAY also receive patches to these documents (notifications) if they are changing frequently. So it MAY happen that the subscriber receives newer versions (with HTTP) than what was indicated in the initial notification. However, once all atomic XCAP modifications are indicated with both previous and new ETags of each resource, it is easy to chain the modification list for a document and possibly omit some of the patches based on the received ETag (with HTTP) of a document. After that the subscriber MAY send a re-subscription to start the diff "aggregation" on the server.

The subscriber can use CSeq values to keep track of possible missing requests. These values can be used to detect missing events also if there are no multiple usages in the current "xcap-diff" dialog. The event bodies, i.e. failing patches and especially the consistent ETag value changes MAY also be used to detect possible missing events. Once the client detects an error it MUST renew the subscription.

If a diff format can not be applied because of some patching errors, look Section 5.1 of [I-D.ietf-simple-xml-patch-ops], the subscriber SHOULD fall back to not to enable any patching within the subscription. It is hardly reasonable to signal this error to the notifier even if the error exists in the notifier process.

4.9. Handling of Forked Requests

This specification only allows a single dialog to be constructed as a result of emitting an initial SUBSCRIBE request. In case a SUBSCRIBE request is forked and the subscriber receives forked responses, the subscriber MUST apply the procedures indicated in Section 4.4.9 of RFC 3265 [RFC3265] for handling non-allowed forked requests.

4.10. Rate of Notifications

Notifiers of "xcap-diff" event package SHOULD NOT generate notifications for a single user at a rate of more than once every five seconds.

4.11. State Agents

State agents play no role in this package.
5. An Initial Example NOTIFY document

Figure 2 shows an example initial XCAP-Diff document provided by the first NOTIFY request. The subscriber used the list as in the example in Figure 1. An example event header of this SUBSCRIBE request:

Event: xcap-diff; diff-processing=aggregate

The subscriber requests the notifier to actually "aggregate" XCAP component updates together. It is anticipated that the subsequent notifications would contain aggregated patches to these documents.

Note: If these documents are changing frequently during the initial synchronization stage, it may happen that the subscriber can not synchronize the contents of all documents properly. However, the subscriber can always begin with the default "xcap-patching" mode where all individual changes with the full ETag change history are shown and this issue does not occur. Also a future extension to XCAP specification MAY solve this versioning issue in a better way.
<?xml version="1.0" encoding="UTF-8"?>
<xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff"
          xcap-root="http://xcap.example.com/root/">
<document new-etag="7ahggs"
         sel="resource-lists/users/sip:joe@example.com/index"/>
<document new-etag="30376adf"
         sel="pidf-manipulation/users/sip:joe@example.com/index"/>
<d:element sel="rls-services/users/sip:joe@example.com/index/~~/*/service[@uri='sip:marketing@example.com']"
           xmlns:d="urn:ietf:params:xml:ns:xcap-diff"
           xmlns="urn:ietf:params:xml:ns:rls-services"
           xmlns:rl="urn:ietf:params:xml:ns:resource-lists">
   <service uri="sip:marketing@example.com">
      <list name="marketing">
         <rl:entry uri="sip:joe@example.com"/>
         <rl:entry uri="sip:sudhir@example.com"/>
      </list>
      <packages>
         <package>presence</package>
      </packages>
   </service>
</d:element>
</xcap-diff>

Figure 2: An example initial XCAP-Diff document

Note that the resource-list "index" document included only the new ETag value, as the document existed during the subscription time (resource was not created, it just exists by the subscription time). In the "pidf-manipulation" collection there was only a single document where the user had read privilege. The <services> element existed within the rls-services "index" document and its content was shown.

6. IANA Considerations

This memo calls for IANA to:

- register a new package name per [RFC3265].
6.1. Registration of the "xcap-diff" Event Package

This specification instructs IANA to register an event package in the SIP Event Types Namespace, based on the registration procedures defined in RFC 3265 [RFC3265]. The following is the information required for such a registration:

Package Name: xcap-diff
Package or Template-Package: This is a package.
Published Document: RFCXXXX
Person to Contact: Jari Urpalainen, jari.urpalainen@nokia.com

7. Security Considerations

This document defines a new SIP event package for the SIP event notification framework specified in RFC 3265 [RFC3265]. As such, all the Security considerations of RFC 3265 [RFC3265] apply. The configuration data can contain sensitive information, and both the client and the server need to authenticate each other. XCAP provides basic authorization policy for resources. Notifiers and subscribers MAY use S/MIME feature to provide authentication and message integrity. This is described in Section 23 of RFC 3261 [RFC3261].

8. Acknowledgments

The author would like to thank Jonathan Rosenberg for his valuable comments and providing the initial event package, and Miguel Garcia, Pavel Dostal, Krisztian Kiss, Anders Lindgren and Sofie Lassborn for their valuable comments.

9. References

9.1. Normative References


9.2. Informative References


(SIP) Events for Conditional Event Notification",
draft-ietf-sip-subnot-etag-01 (work in progress),
August 2007.

Author’s Address

Jari Urpalainen
Nokia
Itamerenkatu 11-13
Helsinki 00180
Finland

Phone: +358 7180 37686
Email: jari.urpalainen@nokia.com