An Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Usages for Conference Policy Manipulation and Conference Policy Privelges Manipulation

draft-ietf-xcon-cpcp-xcap-02

Status of this Memo

Copyright Notice

Abstract

The Conference Policy is defined as the complete set of rules for a particular conference manipulated by the conference policy server. The Conferece Policy Control Protocol (CPCP) is the protocol used by client to manipulate the conference policy. This document defines an XML Configuration Access Protocol (XCAP) application usage that may be used to store and manipulate a conference policy.

There also exists an Extensible Markup Language (XML) Schema that

enumerates the conference policy meta data that enable a user to assign privileges to users that enables them to read and/or manipulate parts of or the entirety of a conference policy. This document defines an XML Configuration Access Protocol (XCAP) application usage that may be used to store and manipulate a conference policy priveleges XML document.

Table of Contents

1. Introduction

The SIP conferencing framework [8] defines the mechanisms for multi-party centralized conferencing in a SIP environment.

Existing SIP mechanisms allow users, for example, to join and leave a conference, as described in [5]. A centralised server, called focus, can expel and invite users, and may have proprietary access control lists and user privilege definitions. The Conference Policy Control Protocol [1] defines an XML Schema that enumerates the conference policy data elements that enable a user to define a conference policy. This policy document may be given to a focus using a number of transports. Mechanisms such as a web page or a voice response system can also be used to manipulate conference policy data.

Similarily, Privileges for Manipulating a Conference Policy [2] defines an Extensible Markup Language (XML) Schema that enumerates the conference policy meta data that enable a user to assign privileges to users that enables them to read and/or manipulate a conference policy. Mechanims are also needed to manipulate such data.

In many cases it is useful to have standardised means to manipulate conference policy elements and conference policy privileges elements. Two XML Configuration Access Protocol (XCAP) [6] application usages are defined that allow for the real-time manipulation of conference policy and conference policy privileges and meets the requirements in [4] to store and manipulate a conference policy object and a conference policy privileges object.

XCAP has many advantages in its use for conference policy control protocol. It is a HTTP 1.1 based protocol that allows clients to read, write, modify and delete application data stored in XML format at a server. XCAP maps XML document elements and attributes to HTTP URIs that can be directly accessed by HTTP. One application area which has already adopted XCAP is the manipulation of event lists [7].

For manipulation of the Conference Policy XML object, the system MAY support the XCAP usage defined in Section 4. For manipulation of the Conference Policy Privileges XML object, the system MAY support the XCAP usage defined in Section 5.

2. Conventions Used in This Document

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 [3].

3. Terminology

This document uses terminology from [8]. Some additional definitions are introduced in [1].

4. An XCAP Usage for Conference Policy Manipulation

4.1 Application Unique ID

XCAP requires application usages to define a unique application usage ID (AUID) in either the IETF tree or a vendor tree. This specification defines the "conference-policies" AUID within the IETF tree, via the IANA registration in Section 8.

4.2 Resource Interdependencies

The conference policy server MAY fill the conference URI(s), but the client MUST propose a conference URI. If the CPS does not allow assignments of URIs by the client, it rejects the request with a "409" response and SHOULD include a body in the response detailing the error. XCAP Base document [6] section 7.2.1 explains how such a response body is constructed. The CPS MAY assign multiple conference URIs to a conference, one for each call signaling protocol that it supports. Section xx of [1] (Conference Settings) discusses this is more detail.

Sidebar URIs are subject to the same behaviour.

4.3 Additional Constraints

These are defined within the XML structure definition in [1].

4.4 Naming Conventions

There are no naming conventions that need to be defined for this application usage.

4.5 Authorization Policies

A server can allow privileged users to modify documents that they don't own. The establishment and indication of such policies is done by setting the authorization rules as described in [2].

4.6 MIME Type for CPCP XML Document

The MIME type for the CPCP XML document is defined in [1].

5. An XCAP Usage for Conference Policy Privileges Manipulation

5.1 Application Unique ID

XCAP requires application usages to define a unique application usage ID (AUID) in either the IETF tree or a vendor tree. This specification defines the "conference-policy-privileges" AUID within the IETF tree, via the IANA registration in Section 8.

5.2 Resource Interdependencies

There are no resource interdependencies that need to be defined fo this application usage.

5.3 Additional Constraints

These are defined within the XML structure definition in [2].

5.4 Naming Conventions

There are no naming conventions that need to be defined for this application usage.

5.5 Authorization Policies

This application usage does not modify the default XCAP authorization policy, which is that only a user can read, write or modify their own documents.

5.6 MIME Type for CPCP XML Document

The MIME type for the Conference Policy Privileges XML document is defined in [2]

6. Examples

6.1 Conference Policy Manipulation

6.1.1 Creating a Conference

Continuing with the example in Section xx of [1], Alice's client uses XCAP to transport the conference policy to the conference policy server

```
      PUT
      http://xcap.example.com/services/conference-policies/users/Alice/
conference.xml HTTP/1.1
```

```
      Content-Type: application/conference-policy+xml

      [conference policy from [1] example goes here].
```

At exactly 2004-12-17T09:30:00-05:00, the focus sends SIP INVITE request to Alice and a SIP REFER request to Sarah. At 2004-12-17T09:25:00-05:00, SIP INVITE requests can be accepted from anyone at domain example.com. Any attempts to join the conference by users in other domains are rejected.

6.1.2 Expelling a User

After the conference has started, Alice decides to expel Bob who has joined the conference. So she modifies the authorization rule that allows everyone at example.com to join:

```
      PUT
      http://xcap.example.com/services/conference-policies/users/Alice/
conference.xml/~~/conference/authorization-rules/rule[@id=""]/conditions/
identity/ HTTP/1.1

      Content-Type:text/plain



        <identity>
                <domain>example.com</domain>
                <except>bob@example.com</except>
        </identity>
```

At this point, the focus sends a SIP BYE request to Bob ending Bob's participation in the conference. This also guarantees that Bob cannot rejoin the conference since he is explicitly blocked. Any attempt Bob makes in rejoining the conference will fail.

6.1.3 Allowing An Expelled Participant To Join Again

Continuing with the example above, Alice now decides to allow Bob to join again after a period of time. She does so by rewriting parts of the rule that blocks him from joining.

```
      PUT
      http://xcap.example.com/services/conference-policies/users/Alice/
conference.xml/~~/conference/authorization-rules/rule[@id=""]/conditions/
identity/ HTTP/1.1
```

```
      Content-Type:text/plain
```

```
        <identity>
                <domain>example.com</domain>
        </identity>
```

Bob can now rejoin the conference by sending a SIP INVITE request.

6.1.4 Allowing Sarah to Refer Users

Alice now decides that Sarah can ask the focus to refer users to the conference:

```
      PUT
      http://xcap.example.com/services/conference-policies/users/Alice/
conference.xml/~~/conference/authorization-rules/rule[@id="3"] HTTP/1.1

      Content-Type:text/plain



        <rule id="3">
                <conditions>
                        <identity>
                                <uri>sarah@example.com</uri>
                        </identity>
                </conditions>
                <actions>
                        <allow-refer-users-dynamically>true</allow-refer-users-
dynamically>
                </actions>
                <transformations/>
        </rule>
```

6.1.5 Removing A Conference

Alice now decides she no longer wants this conference to exist and therefore deletes the conference:

```
      DELETE
      http://xcap.example.com/services/conference-policies/users/Alice/
conference.xml
```

As a result of this action, the focus sends SIP BYE requests to all current participants in the conference. The conference server terminates the focus thereafter.

6.2 Conference Policy Privileges Manipulation

6.2.1 Creating Conference Policy Privilegtes

Continuing with the example in Section xx of [2], Alice's client uses XCAP to transport the conference policy privileges to the conference policy server

```
      PUT
      http://xcap.example.com/services/conference-policy-privileges/users/
Alice/cp-privileges.xml HTTP/1.1

      Content-Type: application/privileges+xml

      [conference policy privileges from [2] example goes here].
```

7. Security Considerations

A conference document may contain information that is highly sensitive. Its delivery to the conference server needs to happen strictly, paying special attention to integrity and confidentiality. Reading the document is also a security concern since the conference policy contains sensitive information like the topic of the conference, who is allowed to join and the URIs of the users that can participate.

Manipulations of the conference policy have similar security issues. Users with relevant privileges can manipulate parts of the conference policy giving themselves and others privileges to manipulate the conference policy, including the dial-out list and the security level settings for a conference. This can happen because the conference policy itself carries the identities and the authorization rules that apply to those identities. Those authorization rules carry the privileges that certain identities have. If an unauthorized user gets access to this document (pretending to be someone else), s/he can manipulate those rules giving himself and other unauthorized users access to the conference policy. S/he can also manipulate other parts of the conference policy under a false identity. Some of the things that a malicious user can do include: denying users certain privileges, giving himself floor moderation, removing users from lists, removing rules for certain identities, giving privileges to other malicious users, changing the media streams and changing conference time. Therefore, it is very important that only authorized clients are able to manipulate the conference policy. Any conference policy transport protocol MUST provide authentication,

confidentiality and integrity.

In the case that XCAP is used to create and manipulate a conference policy, the XCAP base specification mandates that all XCAP servers MUST implement HTTP Authentication: Basic and Digest Access Authentication [9]. Furthermore, XCAP servers MUST implement HTTP over TLS [10]. It is recommended that administrators of XCAP servers use an HTTPS URI as the XCAP root services URI, so that the digest client authentication occurs over TLS. By using these means, XCAP client and server can ensure the confidentiality and integrity of the XCAP created conference policy document and its manipulation operations, and that only authorized clients are allowed to perform them.

8. IANA Considerations

8.1 XCAP Application Usage IDs

8.1.1 conference-policies

Name of the AUID: conference-policies
Description: Conference policy application manipulates conference policy at a server.

8.1.2 conference-policy-privielges

Name of the AUID: conference-policy-privileges
Description: Conference policy privileges application manipulates conference policy privielges at a server.

9. Acknowledgements

The authors would like to thank Alan Johnston and the IETF XCON working group for their feedback and suggestions.

10 Normative References

[1] Khartabil, H., Koskelainen, P. and A. Niemi, "The Conference Policy Control Protocol (CPCP)", Internet-Draft I-D.draft-ietf-xcon-cpcp, September 2004.

[2] Khartabil, H. and A. Niemi, "Privileges for Manipulating a Conference Policy", Internet-Draft I-D.draft-ietf-xcon-conference-policy-privileges, September 2004.

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

[4] Koskelainen, P. and H. Khartabil, "Requirements for conference policy control protocol", draft-ietf-xcon-cpcp-req-01 (work in progress), January 2004.

[5] Johnston, A. and O. Levin, "Session Initiation Protocol Call Control - Conferencing for User Agents", draft-ietf-sipping-cc-conferencing-03 (work in progress), February 2004.

[6] Rosenberg, J., "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)", draft-ietf-simple-xcap-02 (work in progress), February 2004.

[7] Rosenberg, J., "An Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Usage for Presence Lists", draft-ietf-simple-xcap-list-usage-02 (work in progress), February 2004.

[8] Rosenberg, J., "A Framework for Conferencing with the Session Initiation Protocol", draft-ietf-sipping-conferencing-framework-01 (work in progress), October 2003.

[9] Franks, J., "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

[10] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

Author's Address

Hisham Khartabil
Nokia
P.O. Box 321
Helsinki FIN-00045
Finland

EMail: hisham.khartabil@nokia.com

Intellectual Property Statement

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.