Conference Policy Authorization Rules

XCON Interim Meeting
May 2004

Aki Niemi
aki.niemi@nokia.com

Overview of CPCP Authorization

- Access control list for controlling participation
  - <ACL> element includes rules for URIs
  - Allows wildcarding
- Privilege control list for controlling other conference related permissions
  - <PCL> element includes privileges for URIs
  - Allows wildcarding
  - Currently only one privilege defined for subscribing to conference state

Overview of Common Policy

- <conditions> represent the 'if' part
- <actions> represent the 'then'
- <transformations> give more instructions for the action, almost like a 'how'
- Rule ordering unimportant
- Actions/transformations are positive grants of permission
  - Way to resolve in case multiple rules result in multiple actions/transformations

Applying Common Policy in CPCP

- Extending the common policy specification to conferencing application domain
- Permissions for conference participation and access to conferencing related data
  - Conference state
  - Floor events
- Two models: “active request-response” and “event notification”
  - Possible to use the same ruleset for both

Conditions

- <identity>
  - <uri> is the username or asserted ID of caller
  - <domain> matches for domains
  - <any> matches any ID
- <validity>
  - Open as to how this really applies to conferencing
- Others possible if needed

Actions

- <allow-conference-state>
  - Boolean action for allowing receipt of conference-state notification
- <allow-floor-events>
  - Boolean action for allowing receipt of floor event notifications
- <join-handling>
  - Enumerated integer type for “block”, “confirm” or “allow”

Transformations

- <is-key-participant> for indicating that a joining user is a “key” participant
- <show-conference-info>
- <show-floor-holder>
- <show-floor-requests>

Examples

```
<rule="ff405">
  <conditions>
    <identity>
      <uri>lisa@example.com</uri>
    </identity>
  </conditions>
    <actions>
      <join-handling>accept</join-handling>
    </actions>
    <transformations>
      <is-key-participant>true</is-key-participant>
    </transformations>
</rule>
```

Examples

```
<rule="7a8c">
  <conditions>
    <identity>
      <any />
        <except>joe@example.com</except>
    </identity>
  </conditions>
  <actions>
    <join-handling>confirm</join-handling>
    <allow-conference-state>true</allow-conference-state
  </actions>
  <transformations>
    <show-conference-info>true</show-conference-info>
  </transformations>
</rule>
```

Reasons for Having <any>

- Need to be able to have a conference by default open to anyone
- Without <any> coupled with <except> it is impossible to expel users in public conferences
  - <identity> really needs a way of describing an “all-except” construct
- Same issue for conferences that by default requiring moderator confirmation for joining participants

Conclusion

- This is only a first stab
  - Many open issues
  - Comments needed
- Looks promising though
  - Reuse of already used rule syntax
  - Enables permissions per groups
  - Enables resolving conflicting rules