

Update Report

Shared Resources in RELOAD

draft-knauf-p2psip-share-01

Usage for Distributed Conference Control

draft-knauf-p2psip-disco-03

Alexander Knauf, Gabriel Hege
Thomas Schmidt, Matthias Wählisch

alexander.knauf@haw-hamburg.de, hege@fhtw-berlin.de,
{t.schmidt,waehlich}@ieee.org

Agenda

- **Status Report:** Shared Resources (ShaRe) draft
- **ShaRe Updates:**
 - Redefinition of USER-CHAIN-ACL Access Policy
 - Mechanism for isolating stored data
 - Extensions of configuration document to include variable resource names
- **Distributed Conference Control (DisCo) Updates:**
 - Adaption to new requirements of ShaRe
 - Modified conference joining procedure

Status Report: Share(1)

- draft version -00: Initially presented at IETF 80 (Prague)
 - General consensus to continue this work
 - List feedback (by Marc): XML Config. Document does not allow multiple Access Control Policies per Kind-block – fixed in -01

Status Report: Share(2)

- draft version -01: Submitted 11. July 2011
 - Integrates former USER-PATTERN-MATCH into USER-CHAIN-ACL access control policy
 - Mechanisms for isolating stored data to avoid race conditions
 - Variable resource name XML element has own namespace
 - List feedback (Marc): Order of 'resource_name' and 'user_name' field in Kind structures not clear – discussion follows

USER-CHAIN-ACL Policy

- **Problem in -00:** Kinds proposed as Shared Resources should use several access control policies
 - Not allowed by the XML Config. Document
- **Proposal in -01:** USER-CHAIN-ACL access control policy *concatenates* several ACPs

A value **MUST** be written if:

 - USER-MATCH is true (for non-dictionary Kinds) **OR**
 - USER-NODE-MATCH is true (if Kind is of type dictionary) **OR**
 - Signers username matches *variable resource name pattern* **OR**
 - The corresponding *access control list* contains the signers username
- Validation of conditions at “*source code*” level

Isolating Stored Data

- **Problem:** Concurrent store requests on Shared Resources can cause race conditions
- **Proposal in -01:** Mechanisms for isolating stored data
 - **Case 1:** Shared Resource uses dictionary data model
 - Dictionary key **MUST** be equal to signers Node-ID
 - **Case 2:** Shared Resource uses array data model
 - Array indexes are built as a concatenation of the least significant 24 bits of the signers Node-ID + an 8 bit individual
 - Technique related to SSRC identifier generation in RTP (RFC3550)
 - **Case 3:** Shared Resource is a single value
 - Not allowed

XML Extension and Namespace

- **Changes in -01:**

- The `<variable-resource-name>` element is now sub-element of `<kind-block>`

- Uses its own namespace:


namespace share = "urn:ietf:params:xml:ns:p2p:config-base:share"

Order of Resource_Name and User_name in Kind structs (1)

- **Changes in -01:** Each Kind that uses USER-CHAIN-ACL access control policy **MUST** define:
 - An opaque $\langle 0..2^{16}-1 \rangle$ **initial** field within the Kind data structure definition containing the Resource Name
 - An opaque $\langle 0..2^{16}-1 \rangle$ as **second** field within the Kind data structure definition containing the username of the data signer


```
struct {  
    opaque resource_name<0..2^16-1>;  
    opaque user_name<0..2^16-1>;  
    opaque to_user<0..2^16-1>;  
    KindId kind;  
    Boolean allow_delegation;  
} AccessControlListData;
```

Meant as initial
and second field



```
struct {  
    uint16 length;  
    AccessControlListData data;  
} AccessControlListItem;
```

Meta data, not the
beginning of Kind data



Position of Resource_Name and User_name in Kind structs (2)

- **Problem:** Some RELOAD implementations might not be aware of the internal struct hierarchy
 - Try to read the preceding UInt16 length field as $\langle 0..2^{16}-1 \rangle$ opaque Resource Name
- **Proposal:** Resource_name and User_name at the very beginning

```
struct {
    opaque to_user<0..2^16-1>;
    KindId kind;
    Boolean allow_delegation;
} AccessControlListData;

struct {
    opaque resource_name<0..2^16-1>;
    opaque user_name<0..2^16-1>;
    uint16 length;
    AccessControlListData data;
} AccessControlListItem;
```

Agenda

- **Status Report:** Shared Resources (ShaRe) draft
- **ShaRe Updates:**
 - Redefinition of USER-CHAIN-ACL Access Policy
 - Mechanism for isolating stored data
 - Namespace and Position of XML Extension
- **DisCo Updates:**
 - Adaption to new requirements of ShaRe
 - Modified conference joining procedure

DisCo Updates

Changes in -03:

- DisCo uses only USER-CHAIN-ACL as access control policy
- A peer joining a distributed conference sends a Stat request **before** it fetches the DisCo-Registrations and Access Control List (ACL) Kinds
 - Meta data of Stat request used to obtain all indexes of the existing ACL items
 - Those indexes SHOULD be used in the subsequent Fetch request

Thanks for your attention!

Questions?

Alexander Knauf, Gabriel Hege, Thomas Schmidt, Matthias Wählisch

<http://inet.cpt.haw-hamburg.de/>