Open Software Foundation
Request For Comments: 87.0
December 1995

H. Yu (Hewlett Packard)
M. Burati (Hewlett Packard)

DCE 1.2 GLOBAL GROUPS FUNCTIONAL SPECIFICATION

1. INTRODUCTION

It is a limitation of the current implementation of DCE security that groups can contain only *local* members, i.e., can contain only principals that are registered in the same cell as the group. The purpose of the DCE 1.2 Global Groups work is to remove this limitation, i.e., to provide the ability for any DCE group to contain *foreign* principals — and for that membership to be reflected in EPAC's in such a way that it can be interpreted globally by ACL managers (i.e., is usable in ACL's on protected objects in any arbitrary DCE cell).

This feature responds to requirement I1 of OSF-RFC 8.1, and it completes the DCE authorization model to be truly global and uniform (modulo some release-schedule limitations — see below).

2. TARGET

The target audience for this functional specification is vendors supporting multi-cell group environment and customers planning to use this functionality.

3. GOALS AND NON-GOALS

It is a goal of this work to provide membership operations for a foreign principal as a member of a local group.

It is a goal of this work to provide the Privilege Server functionality for inclusion of foreign groups in EPAC's on PTGT requests.

It is not a goal of this work, for DCE 1.2.2 release at least, to validate the membership in a third party cell group (see section 6.2, below).

It is not a goal of this work, for DCE 1.2.2 release at least, to provide Least Privilege functionality (see section 6.1, below).

4. TERMINOLOGY

(a) **Global group**

A group that includes both a Cell identifier and a Group identifier (where identifier is a name or UUID).

(b) **Foreign**

Originating from a cell other than the local one.

(c) **PAC**

Privilege Attribute Certificate. DCE 1.0.x-based DCE credentials.

(d) **EPAC**

Extended Privilege Attribute Certificate. DCE 1.1-based DCE credentials.

(e) **Pseudo-principal node**

A fake principal node in a Registry datastore, used to implement the global group feature. See section 6 for details.

(f) **Third-party group**

When a client principal in CellA makes a request to a server in CellB, claiming membership in a group in CellC, then CellC is called a third-party group. See section 6.2 for details.

(g) **Migration**

In releases since OSF DCE 1.1, there is support for migration between releases in the security server to ensure that new features are disabled until all security replicas within a cell can implement them.

5. ADDITIONAL REQUIREMENTS

This functionality will not affect the existing security behavior, however servers will now be able to acquire foreign group memberships in EPAC's.

6. FUNCTIONAL DEFINITION

With this work, a registry administrator will be able to add a foreign principal as a member of a local group. To accomplish this, a **pseudo-principal** node is created in the local cell registry for the foreign principal, to maintain its membership information. This pseudo-node is intended to be implemented as internal registry data, therefore is not visible to the administrators or users. However this may change in the future. During an intercell PAC or EPAC request, this pseudo-node is examined and its membership information returned to the Privilege Server for inclusion in the requested EPAC.

Work for DCE 1.2.2 includes:

(a) DCE administration enhancements for group membership commands (`dcecp`).

(b) `secd` support for foreign principals as members of local groups.

(c) Privilege server support.

(d) Database conversion, migration, and `sec_salvage_db` support.

6.1. Intercell PTGT Requests and Least Privilege.

By default, when a user logs in to CellA, the security runtime obtains a list of all groups the user is a member of in that cell, and specifies that list of groups in the request for a PTGT.

When obtaining an intercell PTGT, we don't want the CellA privilege server to have to talk to CellB's privilege server: once to obtain the list of foreign groups that principalA is a member of, and then to make the intercell PTGT request. By default, the intercell PTGT request to CellB will result in CellB's privilege server adding all CellB-groups that principalA is a member of to the generated EPAC. This doesn't provide for the ''Least Privilege'' case, where the caller does not want any privileges, except for those that are specifically being requested, to be granted. This ''Least Privilege'' functionality is considered a future (post-1.2.2) enhancement (perhaps via a well-known ERA?).

6.2. Third-Party Group Summary

CellB can believe the group membership specified in a PTGT request from a privilege server in CellA for the case of groups local to CellA, and can verify membership for foreign groups that are actually from CellB (i.e., foreign with respect to that EPAC, but local with respect to CellB's registry). A problem arises only when the request from CellA to CellB contains groups from CellC. We call these **third-party groups**.

Possible solutions to this problem include, but may not be limited to:

(a) Reject all requests containing third-party groups.

(b) Allow the request to continue after silently removing third-party groups from the request.

(c) Don't verify the third-party groups, but include them anyway.

(d) Require CellB's privilege server to contact all third-party (CellC) registries to verify all third-party group memberships.

(e) Create a mechanism whereby the request can specify whether all requested groups are necessary. By default, the privilege server would do the equivalent of (b) above. If the request specified that it needed all groups to be verified to complete its work, then CellB's privilege server would perform the equivalent of (d) above. In that case, if it couldn't verify the membership of any third-party group, the entire request would be denied. One possible mechanism for specifying that third-party groups be verified is via a well-known ERA attached to the EPAC.

Solution (a) above may be too limiting, in that the caller may not have known (or cared) that it was specifying a third-party group, but the request is rejected anyway. In that respect, (a) is an undesirable solution.

Solution (b) allows the request to continue in the event of an incidental (or don't-care) inclusion of a third-party group, but it doesn't provide a mechanism for a client to verify a third-party group if that is actively desired. We think this approach can serve reasonably well as a first cut for the global group functionality, and also meet DCE 1.2.2 schedule demand.

Solution (c) is too insecure and should not be considered.

Solution (d) is too expensive in the default case, and thus should not be considered.

Solution (e) is potentially a better solution than (b), providing valuable extra functionality. But due to DCE 1.2.2 release time frame, we have determined that the value of adding this functionality is not worth the cost of the registry performance hit, and of the extra development time necessary to provide this functionality.

In the cases of (c), (d) and (e) above, whether or not the privilege server will even allow third-party groups in the request will be determined by that privilege server's own policy (possibly via an ERA on the policy object).

For the DCE 1.2.2 release, the only solution that provides the desired functionality and fits within the project's time frame is solution (b), so that is what will be provided.

NOTE: Providing any solution like (d) or (e) above means figuring out how a privilege server in CellB could even possibly locate a privilege server in CellC, given that the only information that is available at that point is the UUID of the other cell. The UUID can be used to figure out the `krbtgt/CellC` cellname. It is possible to just convert `krbtgt/CellC` to `/.../CellC`, and use that to locate CellC's privilege server in the presence of the Hierarchical Cells feature.

LIMITATION: When principalA in CellA is a member of a CellB group, then by default principalA's EPAC will not include that CellB group in its membership list. So if, say, FileA in CellA allows access to that CellB-group, principalA may not be able to access FileA since there is no indication in CellA that principalA has group membership in Cellb.

7. USER INTERFACES

`dcecp` must be enhanced for this work. The command syntax is similar to local group member operations. The difference is that *`princ`* can be a valid global principal name, e.g., `/.../`*`cell_name`*`/`*`principal_name`*. Pre-DCE 1.2.2, `dcecp` member operation implementation parses out a principal's cell name, and only the local principal name is passed to the security server. With DCE 1.2.2, `dcecp` will pass the principal name from its commands, without any parsing, to the security server. This may require updating existing tests if they have different assumptions. Following are the `dcecp` commands that must be extended for supporting foreign principal member operations.

(a) `group add` *`grp`* `-member` *`princ`*

(b) `group remove` *`grp`* `-member` *`princ`*

(c) `group list` *`grp`*

`rgy_edit` will also support foreign principals as local group members with the following commands:

(a) `member group_name [-a member_list] [-r member_list]`

(b) `view pgo_name -m`

8. API'S

We reuse the existing API's: `sec_rgy_pgo_add_member()`, `sec_rgy_pgo_delete_member()`, `sec_rgy_pgo_is_member()`, and `sec_rgy_pgo_get_members()`.

For these API's, the `person_name` parameter, for pre-DCE 1.2.2, can only be a local principal name; now it can be a global name, indicating both cell name and principal name. For `sec_rgy_pgo_get_members()`, the returned `member_list` parameter holds a list of principal names, or global names for foreign principals.

9. REMOTE INTERFACES

No new interfaces are introduced for this work in this release. We are reusing the existing member propagation protocols for servers. See the Compatibility section, below, for more information.

10. MANAGEMENT INTERFACES

`dcecp` and `rgy_edit` commands for multi-cell group operations are the same as local group operations.

11. RESTRICTIONS AND LIMITATIONS

As noted above (in section 6), this work (for release 1.2.2 at least) will not validate the membership of a third-party cell group, and will not provide Least Privilege functionality.

12. OTHER COMPONENT DEPENDENCIES

`dcecp` needs to be enhanced as described above.

13. COMPATIBILITY

All security servers in a cell that is running DCE 1.2.2 need to migrate forward to support this new functionality. Pre-DCE 1.2.2 security servers don't support multi-cell group operations; replication of such events and new data to those servers would not succeed. Because of that, migration is required to ensure all servers running in a cell can implement this new functionality. More details about migration can be found in DCE 1.1 Warranty Patch release document.

For a multi-cell group application program communicating with a DCE 1.2.2 security server via a pre-DCE 1.2.2 client, users may get a ''Registry object not found'' error from a global name in the member operations. Taking the example of `dcecp` as the client, pre-DCE 1.2.2 `dcecp` parses out the cell name from the global name and passes only the shorthand local name to the server. If locally such a shorthand principal name exists, then this will create an incorrect result (e.g., adding this local principal as a member of the group). If no such shorthand principal exists locally, the error mentioned (''Registry object not found'') will be returned.

There has always been a placeholder for foreign groups in PAC's and EPAC's, and the 1.0.x and 1.1 pickling/encoding routines are already there. Therefore a pre-DCE 1.2.2 client should be able to acquire foreign group in its PAC's or EPAC's from a DCE 1.2.2 security server. Since there has been a bug found in the 1.0.x foreign group pickling routine (fixed in DCE1.1), suppliers of DCE 1.0.x will need to ship a patch of `libdce` with this fix. Without this fix, DCE 1.0.x based applications may crash if they encounter a foreign group in a PAC.

14. STANDARDS

This work is compatible with the AES/DC Security Volume.

15. OPEN ISSUES

None identified.

AUTHORS' ADDRESSES

Hanfei Yu
Hewlett Packard
300 Apollo Drive
Chelmsford, MA 01824
USA
Internet email: hanfei@apollo.hp.com
Telephone: +1-508-436-4935

Michael Burati
Hewlett Packard
300 Apollo Drive
Chelmsford, MA 01824
USA
Internet email: burati@apollo.hp.com
Telephone: +1-508-436-4351