6.1. Access Control Principles

www.centos.org/docs/5/html/CDS/ag/8.0/Managing_Access_Control-Access_Control_Principles.html

The mechanism which defines user access is called *access control*. When the server receives a request, it uses the authentication information provided by the user in the bind operation and the access control instructions (ACIs) defined in the server to allow or deny access to directory information. The server can allow or deny permissions for actions on entries like read, write, search, and compare. The permission level granted to a user may depend on the authentication information provided.

Access control in Directory Server is flexible enough to provide very precise rules on when the ACIs are applicable:

- For the entire directory, a subtree of the directory, specific entries in the directory (including entries defining configuration tasks), or a specific set of entry attributes.
- For a specific user, all users belonging to a specific group or role, or all users of the directory.
- For a specific location such as an IP address or a DNS name.

6.1.1. ACI Structure

Access control instructions are stored in the directory as attributes of entries. The `aci` attribute is an operational attribute; it is available for use on every entry in the directory, regardless of whether it is defined for the object class of the entry. It is used by the Directory Server to evaluate what rights are granted or denied when it receives an LDAP request from a client. The `aci` attribute is returned in an `ldapsearch` operation if specifically requested.

The three main parts of an ACI statement are:

- Target
- Permission
- Bind Rule

The permission and bind rule portions of the ACI are set as a pair, also called an *access control rule* (ACR). The specified permission is granted or denied depending on whether the accompanying rule is evaluated to be true.

6.1.2. ACI Placement

If an entry containing an ACI does not have any child entries, the ACI applies to that entry only. If the entry has child entries, the ACI applies to the entry itself and all entries below it. As a direct consequence, when the server evaluates access permissions to any given entry, it verifies the ACIs for every entry between the one requested and the directory suffix, as well as the ACIs on the entry itself.

The `aci` attribute is multi-valued, which means that you can define several ACIs for the same entry or subtree.

An ACI created on an entry can be set so it does not apply directly to that entry but to some or all of the entries in the subtree below it. The advantage of this is that general ACIs can be placed at a high level in the directory tree that effectively apply to entries more likely to be located lower in the tree. For example, an ACI that targets entries that include the `inetorgperson` object class can be created at the level of an `organizationalUnit` entry or a `locality` entry.

Minimize the number of ACIs in the directory tree by placing general rules at high level branch points. To limit the scope of more specific rules, place them as close as possible to leaf entries.

NOTE

ACIs placed in the root DSE entry apply only to that entry.

6.1.3. ACI Evaluation

To evaluate the access rights to a particular entry, the server compiles a list of the ACIs present on the entry itself and on the parent entries back up to the top level entry stored on the Directory Server. ACIs are evaluated across all of the databases for a particular Directory Server but not across all Directory Server instances.

The evaluation of this list of ACIs is done based on the semantics of the ACIs, not on their placement in the directory tree. This means that ACIs that are close to the root of the directory tree do not take precedence over ACIs that are closer to the leaves of the directory tree.

For Directory Server ACIs, the *precedence rule* is that ACIs that deny access take precedence over ACIs that allow access. Between ACIs that allow access, union semantics apply, so there is no precedence.

For example, if you deny write permission at the directory's root level, then none of the users can write to the directory, regardless of the specific permissions you grant them. To grant a specific user write permissions to the directory, you have to restrict the scope of the original denial for write permission so that it does not include the user.

6.1.4. ACI Limitations

When creating an access control policy for your directory service, you need to be aware of the following restrictions:

- If your directory tree is distributed over several servers using the chaining feature, some restrictions apply to the keywords you can use in access control statements:

  - ACIs that depend on group entries (`groupdn` keyword) must be located on the same server as the group entry. If the group is dynamic, then all members of the group must have an entry on the server, too. If the group is static, the members' entries can be located on remote servers.

  - ACIs that depend on role definitions (`roledn` keyword) must be located on the same server as the role definition entry. Every entry that is intended to have the role must also be located on the same server.

  However, you can match values stored in the target entry with values stored in the entry of the bind user; for example, using the `userattr` keyword. Access is evaluated normally even if the bind user does not have an entry on the server that holds the ACI.

  For more information on how to chain access control evaluation, see Section 3.3.5, “Database Links and Access Control Evaluation”.

- Attributes generated by class of service (CoS) cannot be used in all ACI keywords. Specifically, you should not use attributes generated by CoS with the following keywords:

  If you create target filters or bind rules that depend on the value of attributes generated by CoS, the access control rule will not work. For more information on CoS, see Chapter 5, *Managing Entries with Roles, Class of Service, and Views*.

- Access control rules are always evaluated on the local server. Therefore, it is not necessary to specify the hostname or port number of the server in LDAP URLs used in ACI keywords. If you do, the LDAP URL is not taken into account at all. For more information on LDAP URLs, see Appendix C, *LDAP URLs*.