Internet-Draft
LDAP Extensions WG
Intended Category: Informational
Expires: 25 December 1999

E. Stokes
D. Byrne
IBM
B. Blakley
Dascom
P. Behera
Netscape

25 June 1999

Access Control Requirements for LDAP

<draft-ietf-ldapext-acl-reqts-02.txt>

STATUS OF THIS MEMO

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts. Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet- Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Comments and suggestions on this document are encouraged. Comments on this document should be sent to the LDAPEXT working group discussion list:

ietf-ldapext@netscape.com

ABSTRACT

This document describes the fundamental requirements of an access control list (ACL) model for the Lightweight Directory Application Protocol (LDAP) directory service. It is intended to be a gathering place for access control requirements needed to provide authorized access to and interoperability between directories. The RFC 2119 terminology is used in this document.

1. Introduction

The ability to securely access (replicate and distribute) directory information throughout the network is necessary for successful deployment. LDAP's acceptance as an access protocol for directory information is driving the need to provide an access control model definition for LDAP directory content among servers within an enterprise and the Internet. Currently LDAP does not define an access control model, but is needed to ensure consistent secure access across heterogeneous LDAP implementations. The requirements for access control are critical to the successful deployment and acceptance of LDAP in the market place.

The RFC 2119 terminology is used in this document.

2. Objectives

The major objective is to provide a simple, but secure, highly efficient access control model for LDAP while also providing the appropriate flexibility to meet the needs of both the Internet and enterprise environments and policies.

This generally leads to several general requirements that are discussed below.

3. Requirements

This section is divided into several areas of

requirements: general, semantics/policy, usability, and nested groups (an unresolved issue). The requirements are not in any priority order. Examples and explanatory text is provided where deemed necessary. Usability is perhaps the one set of requirements that is generally overlooked, but must be addressed to provide a secure system. Usability is a security issue, not just a nice design goal and requirement. If it is impossible to set and manage a policy for a secure situation that a human can understand, then what was set up will probably be non-secure. We all need to think of usability as a functional security requirement.

3.1 General

G1. Model SHOULD be general enough to support extensibility to add desirable features in the future.

G2. When in doubt, safer is better, especially when establishing defaults.

G3. ACL administration SHOULD be part of the LDAP protocol. Access control information MUST be an LDAP attribute.

G4. Object reuse protection SHOULD be provided and MUST NOT inhibit implementation of object reuse. The directory SHOULD support policy controlling the re-creation of deleted DNs, particularly in cases where they are re-created for the purpose of assigning them to a subject other than the owner of the deleted DN.

3.2 Semantics / Policy

S1. Omitted as redundant; see U8.

S2. More specific policies must override less specific ones (e.g. individual user entry in ACL SHOULD take precedence over group entry) for the evaluation of an ACL.

S3. Multiple policies of equal specificity SHOULD be combined in some easily-understood way (e.g. union or intersection). This is best understood by example. Suppose user A belongs to 3 groups and those 3 groups are

listed on the ACL. Also suppose that the permissions for each of those groups are not identical. Each group is of equal specificity (e.g. each group is listed on the ACL) and the policy for granting user A access (given the example) SHOULD be combined in some easily understood way, such as by intersection or union. For example, an intersection policy here may yield a more limited access for user A than a union policy.

S4. Newly created directory entries SHOULD be subject to a secure default policy.

S5. Access policy SHOULD NOT be expressed in terms of attributes which the directory administrator or his organization cannot administer (e.g. groups whose membership is administered by another organization).

S6. Access policy SHOULD NOT be expressed in terms of attributes which are easily forged (e.g. IP addresses). There may be valid reasons for enabling access based on attributes that are easily forged and the behavior/implications of doing that should be documented.

S7. Humans (including administrators) SHOULD NOT be required to manage access policy on the basis of attributes which are not "human-readable" (e.g. IP addresses).

S8. It MUST be possible to deny a subject the right to invoke a directory operation. The system SHOULD NOT require a specific implementation of denial (e.g. explicit denial, implicit denial).

S9. The system MUST be able (semantically) to support either default-grant or default-deny semantics (not simultaneously).

S10. The system MUST be able to support either union semantics or intersection semantics for aggregate subjects (not simultaneously).

S11. Absence of policy SHOULD be interpretable as grant or deny. Deny takes precedence over grant among entries of equal specificity.

S12. ACL policy resolution MUST NOT depend on the order of entries in the ACL.

S13. Rights management MUST have no side effects. Granting a subject one right to an object MUST NOT implicitly grant the same or any other subject a different right to the same object. Granting a privilege attribute to one subject MUST NOT implicitly grant the same privilege attribute to any other subject. Granting a privilege attribute to one subject MUST NOT implicitly grant a different privilege attribute to the same or any other subject. Definition: An ACL's "scope" is defined as the set of directory objects governed by the policy it defines; this set of objects is a sub-tree of the directory. Changing the policy asserted by an ACL (by changing one or more of its entries) MUST NOT implicitly change the policy governed by an ACL in a different scope.

S14. It SHOULD be possible to apply a single policy to multiple directory entries, even if those entries are in different subtrees. Applying a single policy to multiple directory entries SHOULD NOT require creation and storage of multiple copies of the policy data. The system SHOULD NOT require a specific implementation (e.g. nested groups, named ACLs) of support for policy sharing.

3.3 Usability (Manageability)

U1. When in doubt, simpler is better, both at the interface and in the implementation.

U2. Subjects MUST be drawn from the "natural" LDAP namespace; they should be DNs.

U3. It SHOULD NOT be possible via ACL administration to lock all users, including all administrators, out of the directory.

U4. Administrators SHOULD NOT be required to evaluate arbitrary Boolean predicates in order to create or understand policy.

U5. Administrators SHOULD be able to administer access to directories and their attributes based on their

sensitivity, without having to understand the semantics of individual schema elements and their attributes (see U9).

U6. Management of access to resources in an entire subtree SHOULD require only one ACL (at the subtree root). Note that this makes access control based explicitly on attribute types very hard, unless you constrain the types of entries in subtrees. For example, another attribute is added to an entry. That attribute may fall outside the grouping covered by the ACL and hence require additional administration where the desired affect is indeed a different ACL. Access control information specified in one administrative area MUST NOT have jurisdiction in another area. You SHOULD NOT be able to control access to the aliased entry in the alias. You SHOULD be able to control access to the alias name.

U7. Override of subtree policy MUST be supported on a per-directory-entry basis.

U8. Control of access to individual directory entry attributes (not just the whole directory entry) MUST be supported.

U9. Administrator MUST be able to coarsen access policy granularity by grouping attributes with similar access sensitivities.

U10. Control of access on a per-user granularity MUST be supported.

U11. Administrator MUST be able to aggregate users (for example, by assigning them to groups or roles) to simplify administration.

U12. It MUST be possible to review "effective access" of any user, group, or role to any entry's attributes. This aids the administrator in setting the correct policy.

U13. A single administrator SHOULD be able to define policy for the entire directory tree. An administrator MUST be able to delegate policy administration for specific subtrees to other users. This allows for the partitioning of the entire directory tree for policy

administration, but still allows a single policy to be defined for the entire tree independent of partitioning. (Partition in this context means scope of administration). An administrator MUST be able to create new partitions at any point in the directory tree, and MUST be able to merge a superior and subordinate partition. An administrator MUST be able to configure whether delegated access control information from superior partitions is to be accepted or not.

U14. It MUST be possible to authorize users to traverse directory structure even if they are not authorized to examine or modify some traversed entries; it MUST also be possible to prohibit this. The tree structure MUST be able to be protected from view if so desired by the administrator.

U15. It MUST be possible to create publicly readable entries, which may be read even by unauthenticated clients.

U16. The model for combining multiple access control list entries referring to a single individual MUST be easy to understand.

U17. Administrator MUST be able to determine where inherited policy information comes from, that is, where ACLs are located and which ACLs were applied. Where inheritance of ACLs is applied, it must be able to be shown how/where that new ACL is derived from.

U18. It SHOULD be possible for the administrator to configure the access control system to permit users to grant additional access control rights for entries which they create.

4. Security Considerations

Access control is a security consideration. This documents addresses the requirements.

5. Glossary

This glossary is intended to aid the novice not versed in depth about access control. It contains a list [2] of terms and their definitions that are commonly used in discussing access control.

Access control - The prevention of use of a resource by unidentified and/or unauthorized entities in any other that an authorized manner.

Access control list - A set of control attributes. It is a list, associated with a security object or a group of security objects. The list contains the names of security subjects and the type of access that may be granted.

Access control policy - A set of rules, part of a security policy, by which human users, or their representatives, are authenticated and by which access by these users to applications and other services and security objects is granted or denied.

Access context - The context, in terms of such variables as location, time of day, level of security of the underlying associations, etc., in which an access to a security object is made.

Authorization - The granting of access to a security object.

Authorization policy - A set of rules, part of an access control policy, by which access by security subjects to security objects is granted or denied. An authorization policy may be defined in terms of access control lists, capabilities, or attributes assigned to security subjects, security objects, or both.

Control attributes - Attributes, associated with a security object that, when matched against the privilege attributes of a security subject, are used to grant or deny access to the security object. An access control list or list of rights or time of day range are examples of control attributes.

Credentials - Data that serve to establish the claimed identity of a security subject relative to a given security domain.

Privilege attributes - Attributes, associated with a security subject that, when matched against control attributes of a security object, are used to grant or deny access to that subject. Group and role memberships are examples of privilege attributes.

Security attributes - A general term covering both privilege attributes and control attributes. The use of security attributes is defined by a security policy.

Security object - An entity in a passive role to which a security policy applies.

Security policy - A general term covering both access control policies and authorization policies.

Security subject - An entity in an active role to which a security policy applies.

6. References

[1] Steve Kille, Tim Howes, M. Wahl, "Lightweight Directory Access Protocol (v3)", RFC 2251, August 1997.

[2] ECMA, "Security in Open Systems: A Security Framework" ECMA TR/46, July 1988

AUTHOR(S) ADDRESS

Bob Blakley
Dascom
5515 Balcones Drive
Austin, TX 78731
USA
mail-to: blakley@dascom.com
phone: +1 512 458 4037 ext 5012
fax: +1 512 458 2377

Ellen Stokes
IBM
11400 Burnet Rd
Austin, TX 78758
USA
mail-to: stokes@austin.ibm.com
phone: +1 512 838 3725
fax: +1 512 838 0156

Debbie Byrne
IBM
11400 Burnet Rd
Austin, TX 78758
USA
mail-to: djbyrne@us.ibm.com
phone: +1 512 838 1930
fax: +1 512 838 8597

Prasanta Behera
Netscape
501 Ellis Street
Mountain View, CA 94043
USA
mail-to: prasanta@netscape.com
phone: +1 650 937 4948
fax: +1 650 528-4164

7. Full Copyright Statement