Network Working Group
Internet Draft
Document: draft-myers-imap-acl-02.txt

J. Myers
Carnegie Mellon
June 1996

IMAP4 ACL extension

Status of this Memo

1. Abstract

The ACL extension of the Internet Message Access Protocol [IMAP4] permits access control lists to be manipulated through the IMAP protocol.

2. Conventions Used in this Document

In examples, "C:" and "S:" indicate lines sent by the client and server respectively.

3. Introduction and Overview

The ACL extension is present in any IMAP4 implementation which returns "ACL" as one of the supported capabilities to the CAPABILITY command.

An access control list is a set of <identifier,rights> pairs.

Identifier is a US-ASCII string. The identifier anyone is reserved to refer to the universal identity (all authentications, including anonymous). All user name strings accepted by the LOGIN or AUTHENTICATE commands to authenticate to the IMAP server are reserved as identifiers to refer to the corresponding user. All other identifier strings have implementation-defined semantics.

Possible variations of identifier interpretation include, but are not limited to:

* Named groups of users, presumably managed by some authorization service.

* A prefix to the identifier specifying an "authentication type".

  As an example, an implementation may control posting to a group based on the contents of the From: header:

  from$user p

* Whether the union of rights for matching identifiers are granted to a user or whether the rights for the most specific matching identifier is granted.

As an example, for a mailbox with the following ACL:

```
user                    lrsa
group-user-is-in        lrsw
```


One implementation may grant the user 'lrswa' rights, another may only grant the user 'lrsa' rights.

* A prefix to an identifier name specifying the listed rights are to be removed from users who match the prefixed identifier.

  As an example, for a mailbox with the following ACL:

```
group-user-is-in        lrsw
-user                   w
```

  An implementation may grant the user 'lrs' rights.

Rights is a string listing a (possibly empty) set of alphanumeric characters, each character listing a set of operations which is being controlled. Letters are reserved for ``standard'' rights, listed below. Digits are reserved for implementation or site defined rights. The standard rights are:

l - lookup (mailbox is visible to LIST/LSUB commands)
r - read (SELECT the mailbox, perform CHECK, FETCH, PARTIAL, SEARCH, COPY from mailbox)
s - keep seen/unseen information across sessions (STORE \SEEN flag)
w - write (STORE flags other than \SEEN and \DELETED)
i - insert (perform APPEND, COPY into mailbox)
p - post (send mail to submission address for mailbox, not enforced by IMAP4 itself)
c - create (CREATE new sub-mailboxes in any implementation-defined hierarchy)
d - delete (STORE \DELETED flag, perform EXPUNGE)
a - administer (perform SETACL)

An implementation may tie rights together or may force rights to always or never be granted. For example, in an implementation that uses unix mode bits, the rights "wisd" are tied, the "a" right is always granted to the owner and is never granted to another user. If

rights are tied in an implementation, it should be conservative in granting rights in response to SETACL commands--unless all rights in a tied set are specified, none should be used.

4. Commands

4.1. SETACL

Arguments: mailbox name
authentication identifier
access rights

Data: no specific data for this command

Result: OK - setacl completed
NO - setacl failure: can't set acl
BAD - command unknown or arguments invalid

The SETACL command changes the access control list on the specified mailbox so that the specified identifier is granted the permissions enumerated in rights.

4.2. DELETEACL

Arguments: mailbox name
authentication identifier

Data: no specific data for this command

Result: OK - deleteacl completed
NO - deleteacl failure: can't delete acl

BAD - command unknown or arguments invalid

The DELETEACL command removes any portion of the access control list for mailbox for the specified identifier.

4.3. GETACL

Arguments: mailbox name

Data: untagged responses: ACL

Result: OK - getacl completed
NO - getacl failure: can't get acl
BAD - command unknown or arguments invalid

The GETACL command returns the access control list for mailbox in

an untagged ACL reply.

Example: C: A002 GETACL INBOX
S: * ACL INBOX Fred rwipslda
S: A002 OK Getacl complete

4.4. LISTRIGHTS

Arguments: mailbox name
authentication identifier

Data: untagged responses: LISTRIGHTS

Result: OK - listrights completed
NO - listrights failure: can't get rights list
BAD - command unknown or arguments invalid

Example: C: a001 LISTRIGHTS ~/Mail/saved smith
S: * LISTRIGHTS ~/Mail/saved smith la r swicd
S: a001 OK Listrights completed

```
C: a005 LISTRIGHTS archive.imap anyone
S: * LISTRIGHTS archive.imap anyone "" l r s w i p c d a 0 1
2 3 4 5 6 7 8 9
S: a005 OK Listrights completed
```

The LISTRIGHTS command takes a mailbox name and an identifier and returns information about what rights may be granted to the

4.5. MYRIGHTS

Arguments: mailbox name

Data: untagged responses: MYRIGHTS

Result: OK - myrights completed
NO - myrights failure: can't get rights
BAD - command unknown or arguments invalid

The MYRIGHTS command returns the set of rights that the user has to mailbox in an untagged MYRIGHTS reply.

Example: C: A003 MYRIGHTS INBOX
S: * MYRIGHTS INBOX rwipslda


```
S: A003 OK Myrights complete
```

5. Responses

5.1. ACL

Data: mailbox name
zero or more identifier rights pairs

The ACL response occurs as a result of a GETACL command. The first string is the mailbox name for which this ACL entry applies. This is followed by zero or more pairs of strings, each pair

contains the identifier for which the entry applies followed by the set of rights that the identifier has.

5.2. LISTRIGHTS

Data: mailbox name
identifier
required rights
list of optional rights

The LISTRIGHTS response occurs as a result of a LISTRIGHTS command. The first two strings are the mailbox name and identifier for which this rights list applies. Following the identifier is a string containing the (possibly empty) set of rights the identifier will always be granted in the mailbox.

Following this are zero or more strings each containing a set of rights the identifier may be granted in the mailbox. Rights mentioned in the same string are tied together--either all must be granted to the identifier in the mailbox or none may be granted.

The same right may not be listed more than once in the LISTRIGHTS command.

5.3. MYRIGHTS

Data: mailbox name
rights

The MYRIGHTS response occurs as a result of a MYRIGHTS command. The first string is the mailbox name for which this ACL entry

applies. The second string is the set of rights that the client has.

6. Formal Syntax

The following syntax specification uses the augmented Backus-Naur Form (BNF) notation as specified in [RFC-822] as modified by [IMAP4].

Non-terminals referenced but not defined below are as defined by [IMAP4].

Except as noted otherwise, all alphabetic characters are case-insensitive. The use of upper or lower case characters to define token strings is for editorial clarity only. Implementations MUST accept these strings in a case-insensitive fashion.

```
acl_data        ::= "ACL" SPACE mailbox *(SPACE identifier SPACE rights)

deleteacl       ::= "DELETEACL" SPACE mailbox SPACE identifier

getacl          ::= "GETACL" SPACE mailbox

identifier      ::= astring

listrights      ::= "LISTRIGHTS" SPACE mailbox SPACE identifier

listrights_data ::= "LISTRIGHTS" SPACE mailbox SPACE identifier
                        SPACE rights *(SPACE rights)

myrights        ::= "MYRIGHTS" SPACE mailbox

myrights_data   ::= "MYRIGHTS" SPACE mailbox SPACE rights

rights          ::= astring

setacl          ::= "SETACL" SPACE mailbox SPACE identifier SPACE rights
```

7. References

[IMAP4] Crispin, M., "Internet Message Access Protocol - Version 4", RFC 1730, University of Washington, December 1994.

[RFC-822] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", STD 11, RFC 822.

8. Security Considerations

An implementation must make sure the ACL commands themselves do not give information about mailboxes with appropriately restricted ACL's. For example, a GETACL command on a mailbox for which the user has insufficient rights should not admit the mailbox exists, much less return the mailbox's ACL.

9. Author's Address

John G. Myers
Carnegie-Mellon University
5000 Forbes Ave.
Pittsburgh PA, 15213-3890

Email: jgm+@cmu.edu

Table of Contents