

IMAP4 ACL extension

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

1. Abstract

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

Table of Contents

1.	Abstract.....	1
2.	Conventions Used in this Document.....	1
3.	Introduction and Overview.....	2
4.	Commands.....	3
4.1.	SETACL.....	3
4.2.	DELETEACL.....	4
4.3.	GETACL.....	4
4.4.	LISTRIGHTS.....	4
4.5.	MYRIGHTS.....	5
5.	Responses.....	5
5.1.	ACL.....	5
5.2.	LISTRIGHTS.....	6
5.3.	MYRIGHTS.....	6
6.	Formal Syntax.....	6
7.	References.....	7
8.	Security Considerations.....	7
9.	Author's Address.....	8

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 for the corresponding user. Identifiers starting with a dash ("-") are reserved for "negative rights", described below. All other identifier strings are interpreted in an implementation-defined manner.

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. The set of standard rights may only be extended by a standards-track document. Digits are reserved for implementation or site defined rights. The currently defined 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 to particular identifiers. For example, in an implementation that uses unix mode bits, the rights "wisd" are tied, the "a" right is always granted to the owner of a mailbox and is never granted to another user. If rights are tied in an implementation, the implementation must be conservative in granting rights in response to SETACL commands--unless all rights in a tied set are specified, none of that set should be included in the ACL entry for that identifier. A client may discover the set of rights which may be granted to a given identifier in the ACL for a given mailbox by using the LISTRIGHTS command.

It is possible for multiple identifiers in an access control list to apply to a given user (or other authentication identity). For example, an ACL may include rights to be granted to the identifier matching the user, one or more implementation-defined identifiers matching groups which include the user, and/or the identifier "anyone". How these rights are combined to determine the user's access is implementation-defined. An implementation may choose, for example, to use the union of the rights granted to the applicable identifiers. An implementation may instead choose, for example, to only use those rights granted to the most specific identifier present in the ACL. A client may determine the set of rights granted to the logged-in user for a given mailbox by using the MYRIGHTS command.

When an identifier in an ACL starts with a dash ("-"), that indicates that associated rights are to be removed from the identifier that is prefixed by the dash. For example, if the identifier "-fred" is granted the "w" right, that indicates that the "w" right is to be removed from users matching the identifier "fred". Implementations need not support having identifiers which start with a dash in ACLs.

4. Commands

4.1. SETACL

Arguments: mailbox name
 authentication identifier
 access right modification

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 permissions as specified in the third argument.

The third argument is a string containing an optional plus ("+") or minus ("-") prefix, followed by zero or more rights characters. If the string starts with a plus, the following rights are added to any existing rights for the identifier. If the string starts with a minus, the following rights are removed from any existing rights for the identifier. If the string does not start with a plus or minus, the rights replace any existing rights for the identifier.

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 <identifier, rights> pair for the specified identifier from the access control list for the specified mailbox.

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

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

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

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 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 these rights apply. 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)

mod_rights    ::= astring
                    ;; +rights to add, -rights to remove
                    ;; rights to replace

myrights      ::= "MYRIGHTS" SPACE mailbox

myrights_data ::= "MYRIGHTS" SPACE mailbox SPACE rights

rights        ::= astring

setacl        ::= "SETACL" SPACE mailbox SPACE identifier
                    SPACE mod_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

