IMAPEXT Working Group
Internet Draft
Document: draft-ietf-imapext-2086upd-00.txt
Updates: 2086, <<3501?>>
Expires: March 2005

A. Melnikov
Editor

September 2004

IMAP4 ACL extension - updated list of rights

Status of this Memo

Abstract

The ACL (Access Control List) extension [RFC2086] of the Internet Message Access Protocol [IMAP4] permits mailbox access control lists to be manipulated through the IMAP protocol.

This document updates the list of rights defined in RFC 2086. It also clarifies which rights are required for different IMAP commands.

0. Open issues and ToDo list

This section will be removed when the draft will be published as RFC. It is intended to simplify discussion.

1). Do we want to add a requirement to send MYRIGHTS response on SELECT/EXAMINE?

1. Conventions Used in this Document

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

In all examples "/" character is used as hierarchy separator.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [KEYWORDS].

2. Introduction and Overview

The ACL (Access Control List) extension of the Internet Message Access Protocol [IMAP4] permits mailbox access control lists to be retrieved and manipulated through the IMAP protocol.

This document updates Section 3 of the RFC 2086. It also clarifies different aspects of the RFC 2086, in particular use of UTF-8 in identifiers, which rights are required for different IMAP4 commands; how READ-WRITE/READ-ONLY response codes are related to ACL>>

3. Access Control

This section replaces Section 3 of the RFC 2086.

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

A server implementation conformant to this document MUST also return rights (see below) not defined in RFC 2086 in the "RIGHTS=" capability response.

An access control list is an ordered list of <identifier,rights> pairs. An ACL applies to a mailbox.

Identifier is a UTF-8 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
(note, that the list below doesn't list all commands that use a
particular right):

l - lookup (mailbox is visible to LIST/LSUB commands, SUBSCRIBE mailbox)
r - read (SELECT the mailbox, perform STATUS)
s - keep seen/unseen information across sessions (set or clear \SEEN flag
    via STORE, APPEND or COPY)
w - write (set or clear flags other than \SEEN and \DELETED via STORE,
    APPEND or COPY)
i - insert (perform APPEND, COPY into mailbox)
p - post (send mail to submission address for mailbox,
    not enforced by IMAP4 itself.
c - create mailboxes (CREATE new sub-mailboxes in any
    implementation-defined hierarchy, parent mailbox for the new
    mailbox name in RENAME)
    When a new mailbox is created it SHOULD inherit rights from
    the parent mailbox (if one exists) in the defined hierarchy.
x - delete mailbox (DELETE mailbox, old mailbox name in RENAME)
t - delete messages (set or clear \DELETED flag via STORE, set \DELETED flag
    during APPEND/COPY)
e - perform EXPUNGE and expunge as a part of CLOSE
d - This right is defined for backward compatibility with ACL
    extension (RFC 2086). If a client sets "d" right, the server MUST
    set "x", "e" and "t" rights. When the client clears the "d" right,
    the server MUST clear "x", "e" and "t" rights. When all three of "x",
    "e" and "t" are set, the server MUST return "d" right in response to
    a LIST (ACL) command. If "x", "e" and "t" rights are not tied together,
    "d" right MUST NOT be returned in a LISTRIGHTS response.
a - administer (perform SETACL/DELETEACL/GETACL)
n - write shared annotations [ANNOTATE]

<<Add new right for private annotations, if required>>

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 "swite" 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. This is referred to as a "negative right". This differs from DELETEACL in that a negative right is added to the ACL, and is a part of the calculation of the rights.

Let's assume that an identifier "fred" refers to a user with login "fred". 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", even though the user "fred" might have the "w" right as a consequence of some other identifier in the ACL. A DELETEACL of "fred" simply deletes the identifier "fred" from the ACL; it does not affect any rights that the user "fred" may get from another entry in the ACL.

Server implementations are not required to support "negative right" identifiers.

4. Rights required to perform different IMAP4rev1 commands

Before executing a command an ACL compliant server must check which rights are required to perform it. This section groups command by functions they perform and list the rights required. It also gives the detailed description of any special processing required.

The table below summarizes different rights or their combinations that are required in order to perform different IMAP operations. As it is not always possible to express complex right checking and interactions, the description after the table should be used as the primary reference.

Operations\Rights	l	r	s	w	i	c	x	t	e	a	Any	None

LIST	+													
SUBSCRIBE	*												*	
UNSUBSCRIBE													+	
LSUB	*												*	
CREATE (for parent)							+							
DELETE								+						
RENAME							+	+						
COPY/APPEND				?	?	+				?				
EXPUNGE/CLOSE											+			
SELECT/EXAMINE/STATUS			+											
FETCH				?										
STORE flags				?	?					?				
SETACL/DELETEACL												+		
GETACL												+		
MYRIGHTS/LISTRIGHTS													+	

Legend:
+ - The right is required
* - Only one of the rights marked with * is required (see description below)
? - The right is optional (see description below)
"Any" - at least one of the "l", "r", "i", "c", "x", "e", "a" rights is required
"None" - No rights required to perform the command

Listing and subscribing/unsubscribing mailboxes:
LIST - "l" right is required.

Note, that if the user has "l" right to a mailbox "A/B", but not to its parent
mailbox "A", the LIST command should behave as if the mailbox "A" doesn't exist,
for example:

```
C: A777 LIST "" *
S: * LIST (\NoInferiors) "/" "A/B"
S: * LIST () "/" "C"
S: * LIST (\NoInferiors) "/" "C/D"
S: A777 OK LIST completed
```

SUBSCRIBE - "l" right is required only if the server checks for mailbox existence
when performing SUBSCRIBE.

UNSUBSCRIBE - no rights required to perform this operation.

LSUB - "l" right is required only if the server checks for mailbox existence when
performing SUBSCRIBE.

Mailbox management:
 CREATE - "c" right on a nearest existing parent mailbox. When a new
          mailbox is created it SHOULD inherit rights from the parent
          mailbox (if one exists) in the defined hierarchy.

 DELETE - "x" right on the mailbox.

 RENAME - Moving a mailbox from one parent to another
          requires "x" right on the mailbox itself and "c" right for
          the new parent. For example, if the user wants to rename
          mailbox named "A/B/C" to "D/E", the user must have "x" right
          for the mailbox "A/B/C" and "c" right for the mailbox "D".

Copying or appending messages:

 Before performing a COPY/APPEND command the server MUST check if the
 user has "i" right for the target mailbox. If the user doesn't have "i"
 right, the operation fails. Otherwise for each copied/appended message
 the server MUST check if the user has
  "t" right - when the message has \Deleted flag set
  "s" right - when the message has \Seen flag set
  "w" right for all other message flags.
 Only when the user has a particular right the corresponding flags are
 stored for the newly created message. The server MUST NOT fail
 a COPY/APPEND if the user has no rights to set a particular flag.

Example: C: A003 MYRIGHTS TargetMailbox
            S: * MYRIGHTS TargetMailbox rwis
            S: A003 OK Myrights complete

            C: A004 FETCH 1:3 (FLAGS)
            S: * 1 FETCH (FLAGS (\Draft \Deleted)
            S: * 2 FETCH (FLAGS (\Answered)
            S: * 3 FETCH (FLAGS ($Forwarded \Seen)
            S: A004 OK Fetch Completed

            C: A005 COPY 1:3 TargetMailbox
            S: A005 OK Copy completed

            C: A006 SELECT TargetMailbox
               ...
            S: A006 Select Completed

    Let's assume that the copied messages received message numbers 77:79.

            C: A007 FETCH 77:79 (FLAGS)
            S: * 77 FETCH (FLAGS (\Draft))
            S: * 78 FETCH (FLAGS (\Answered))
            S: * 79 FETCH (FLAGS ($Forwarded \Seen))
            S: A007 OK Fetch Completed

    \Deleted flag was lost on COPY, as the user has no "t" right in the

target mailbox.

If the MYRIGHTS command with the tag A003 would have returned:

```
S: * MYRIGHTS TargetMailbox rsti
```

the response from the FETCH with the tag A007 would have been:

```
C: A007 FETCH 77:79 (FLAGS)
S: * 77 FETCH (FLAGS (\Deleted))
S: * 78 FETCH (FLAGS ())
S: * 79 FETCH (FLAGS (\Seen))
S: A007 OK Fetch Completed
```

In the latter case \Answered, $Forwarded and \Draft flags were lost on COPY, as the user has no "w" right in the target mailbox.

Expunging the selected mailbox:
EXPUNGE - "e" right on the selected mailbox.

CLOSE - "e" right on the selected mailbox. If the server is unable to expunge the mailbox because the user doesn't have the "e" right, the server MUST ignore expunge request, close the mailbox and return tagged OK response.

Fetch information about a mailbox and its messages:
SELECT/EXAMINE/STATUS - "r" right on the mailbox.

FETCH - A FETCH request that implies setting \Seen flag MUST NOT set it, if the current user doesn't have "s" right.

Changing flags:
STORE - the server MUST check if the user has
"t" right - when the user modifies \Deleted flag
"s" right - when the user modifies \Seen flag
"w" right for all other message flags.
STORE operation SHOULD NOT fail if the user has rights to modify at least one flag specified in the STORE, as the tagged NO response to a STORE command is not handled very well by deployed clients.

Changing ACLs:
SETACL/DELETEACL - "a" right on the mailbox.

Reading ACLs:
GETACL - "a" right on the mailbox.

MYRIGHTS - any of the following rights is required to perform the operation: "l", "r", "i", "c", "x", "e", "a".

LISTRIGHTS - same as MYRIGHTS. <<?>> <<Same rights as for GETACL?>>

5. Formal Syntax

This document doesn't change the formal syntax of commands/ responses defined in the Section 6 of RFC 2086. However, the "identifier" production is now allowed to carry any UTF-8 string.

Formal syntax is defined using ABNF [ABNF] as modified by [IMAP4]. Non-terminals referenced but not defined below are as defined by [IMAP4] or [LISTEXT].

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.

```
rights_capa      = "RIGTHS=" new_rights
                   ;; RIGHTS=... capability

new_rights       = atom
                   ;; MUST include "t", "e", "x" and "n" <<ANNOTATE>>
```

6. 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 that the mailbox exists, much less return the mailbox's ACL.

LISTRIGHTS command MUST NOT check that a particular identifier exists, however it SHOULD recognize special identifiers like "anyone".

IMAP clients implementing ACL that are able to modify ACLs SHOULD warn a user that wants to give full access (or even just "a" right) to the special identifier "anyone".

7. Other considerations

7.1. Additional requirements and Implementation notes

This document defines an additional capability that is used to announce the list of extra rights (excluding the ones defined in the RFC 2086) supported by the server. The set of rights MUST include "t", "e", "x" and "n" <<ANNOTATE>>. Note, that the extra rights can appear in any order.

```
Example:    C: 1 capability
            S: * CAPABILITY IMAP4REV1 STARTTLS LITERAL+ ACL RIGTHS=texn<<>>
            S: 1 OK completed
```

A client implementation that allows a user to read and update ACL MUST preserve unrecognized rights that it doesn't allow the user to change

when updating the rights. Otherwise the client may unintentionally remove permissions.

Any server implementing an ACL extension MUST accurately reflect the current user's rights in FLAGS and PERMANENTFLAGS responses.

```
Example:    C: A141 MYRIGHTS INBOX
            S: * MYRIGHTS INBOX rwis
            S: A141 OK Myrights complete
            C: A142 SELECT INBOX
            S: * 172 EXISTS
            S: * 1 RECENT
            S: * OK [UNSEEN 12] Message 12 is first unseen
            S: * OK [UIDVALIDITY 3857529045] UIDs valid
            S: * OK [UIDNEXT 4392] Predicted next UID
            S: * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
            S: * OK [PERMANENTFLAGS (\Seen \Answered \Flagged \*)] Limited
            S: A142 OK [READ-WRITE] SELECT completed
```

An ACL server MAY modify one or more ACL for one or more identifier as a side effect of modifying the ACL specified in a SETACL/DELETEACL.
If the server does that it MUST send untagged ACL response to notify the client about the changes made.

7.2. Mapping of ACL rights to READ-WRITE and READ-ONLY response codes

A particular ACL server implementation may allow "shared multiuser access" to some mailboxes. "Shared multiuser access" to a mailbox means that multiple different users are able to access the same mailbox, if they have proper access rights. "Shared multiuser access" to the mailbox doesn't mean that the ACL for the mailbox is currently set to allow access by multiple users. Let's denote a "shared multiuser write access" as a "shared multiuser access" when a user may be granted flag modification rights (any of "w", "s" or "t").

Section 4 describes which rights are required for modifying different flags.

If the ACL server implements some flags as shared for a mailbox (i.e., the ACL for the mailbox may be set up so that changes to those flags are visible to another user), let's call the set of rights associated with these flags (as described in Section 4) for that mailbox collectively as "shared flag rights". Note, that "shared flag rights" set MAY be different for different mailboxes.

If the server doesn't support "shared multiuser write access" to a mailbox or doesn't implement shared flags on the mailbox, "shared flag rights" for the mailbox is defined to be the empty set.

Example 1: Mailbox "banan" allows "shared multiuser write access" and implements flags \Deleted, \Answered and $MDNSent as

shared flags. "Shared flag rights" for the mailbox "banan" is a set containing flags "t" (because system flag \Deleted requires "t" right) and "w" (because both \Answered and $MDNSent require "w" right).

Example 2: Mailbox "apple" allows "shared multiuser write access" and implements \Seen system flag as shared flag. "Shared flag rights" for the mailbox "apple" contains "s" right, because system flag \Seen requires "s" right.

Example 3: Mailbox "pear" allows "shared multiuser write access" and implements flags \Seen, \Draft as shared flags. "Shared flag rights" for the mailbox "apple" is a set containing flags "s" (because system flag \Seen requires "s" right) and "w" (because system flag \Draft requires "w" right).

The server MUST include a READ-ONLY prefix in the tagged OK response to a SELECT command if none of the following rights is granted to the current user:
"i", "e" and "shared flag rights".
The server SHOULD include a READ-WRITE prefix in the tagged OK response if at least one of the "i", "e" or "shared flag rights" is granted to the current user.

Example 1 (continued): The user that has "lrs" rights for the mailbox "banan". The server returns READ-ONLY response code on SELECT, as none of "iewt" rights is granted to the user.

Example 2 (continued): The user that has "rit" rights for the mailbox "apple". The server returns READ-WRITE response code on SELECT, as the user has "i" right.

Example 3 (continued): The user that has "rset" rights for the mailbox "pear". The server returns READ-WRITE response code on SELECT, as the user has "e" and "s" rights.

8. References

8.1. Normative References

[KEYWORDS] Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, Harvard University, March 1997.

[ABNF] Crocker, Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium, Demon Internet Ltd, November 1997.

[IMAP4] Crispin, M., "Internet Message Access Protocol - Version 4rev1", RFC 3501, University of Washington, March 2003.

[RFC2086] Myers, J., "IMAP4 ACL extension", RFC 2086, Carnegie Mellon, January 1997

[UTF-8] Yergeau, F., "UTF-8, a transformation format of IS0 10646", RFC 2279, Alis Technologies, January 1998.

8.2. Informative References

[ANNOTATE] Gellens, R. and C. Daboo, "IMAP ANNOTATE Extension", work in progress, draft-ietf-imapext-annotate-XX.txt

9. Aknowledgement

This document is a revision of the RFC 2086 written by John G. Myers.

Editor appreciates comments received from Mark Crispin, Chris Newman, Cyrus Daboo, John G. Myers, Steve Hole, Curtis King, Lyndon Nerenberg, Larry Greenfield, Robert Siemborski, Vladimir Butenko, Dave Cridland, Harrie Hazewinkel and other participants of the IMAPEXT working group.

10. Editor's Address

Alexey Melnikov
email: alexey.melnikov@isode.com

Isode Limited

11. IPR Disclosure Acknowledgement

By submitting this Internet-Draft, I certify that any applicable patent or other IPR claims of which I am aware have been disclosed, and any of which I become aware will be disclosed, in accordance with RFC 3668.

12. Intellectual Property Statement

be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

13. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

Appendix A. Changes since RFC 2086

1. Changed the charset of "identifier" from US-ASCII to UTF-8.

2. Specified that mailbox deletion is controled by the "x" right and EXPUNGE is controlled by "e" right.

3. Clarified that RENAME requires "c" right for the new parent and "x" right for the old name.

4. Added "t" right that controls STORE \Deleted. Redefined "d" to be a macro for "e", "x" and "t".

5. Specified that "a" right also controls DELETEACL.

6. Specified that "r" right also controls STATUS.

7. Removed the requirement to check the "r" right for CHECK, SEARCH and FETCH, as this is required for SELECT/EXAMINE to be successful.

8. LISTRIGHTS requires same rights as MYRIGHTS.

9. Deleted "PARTIAL", this is a deprecated feature of RFC1730.

10. Specified that "w" controls setting flags other than \Seen and \Deleted on APPEND. Same for "s" and "t" flags.

11. SUBSCRIBE is NOT allowed with "r" right.

12. Specified that "l" controls SUBSCRIBE.

13. GETACL is NOT allowed with "r" right, even though there are several implementations that allows that. If a user only has "r" right, GETACL can disclose information about identifiers existing on the mail system.

14. Added new section that describes which rights are required and/or checked when performing various IMAP commands.

15. Added mail client security considerations when dealing with special identifier "anyone".

16. Clarified that negative rights are not the same as DELETEACL.

17. Added note that a server can modify an ACL for any identifier(s) as a side effect of performing SETACL/DELETEACL. Also specified that the server MUST send untagged ACL response if it does that. <<Updated command definition to include optional ACL untagged response.>>

18. Added section about mapping of ACL rights to READ-WRITE and READ-ONLY response codes.

19. Added "Compatibility with RFC 2086" section.

20. Added "Implementation Notes" section.

21. Updated "References" section.

Appendix B. Compatibility with RFC 2086

This section gives guidelines how an existing RFC 2086 server implementation may be updated to comply with this document.

This document replaces "d" right with 3 new different rights "x", "t" and "e". The server should implement one of the following two approaches to handle "d" and the new rights that have replaced it.

a). Tie "x", "t" and "e" together - almost no changes
b). Implement separate "x", "t" and "e". Return "d" right in a LIST response containing ACL information when all three of "x", "t" and "e" are granted.

Also check Sections 7.1 and 7.2, as well as the appendix A to see

other changes required. Server implementors should check which rights are required to invoke different IMAP4 commands as described in Section 4.