IMAP4 ID extension

Status of this Memo

Copyright Notice

Abstract

   The ID extension to the Internet Message Access Protocol - Version
   4rev1 (IMAP4rev1) protocol allows the server and client to exchange
   identification information on their implementation in order to make
   bug reports and usage statistics more complete.

1. Introduction

   The IMAP4rev1 protocol described in [IMAP4rev1] provides a method for
   accessing remote mail stores, but it provides no facility to
   advertise what program a client or server uses to provide service.
   This makes it difficult for implementors to get complete bug reports
   from users, as it is frequently difficult to know what client or
   server is in use.

   Additionally, some sites may wish to assemble usage statistics based
   on what clients are used, but in an an environment where users are
   permitted to obtain and maintain their own clients this is difficult
   to accomplish.

   The ID command provides a facility to advertise information on what
   programs are being used along with contact information (should bugs
   ever occur).

2. Conventions Used in this Document

   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 [KEYWORDS].

   The conventions used in this document are the same as specified in
   [IMAP4rev1]. In examples, "C:" and "S:" indicate lines sent by the
   client and server respectively. Line breaks have been inserted for
   readability.

3. Specification

   The sole purpose of the ID extension is to enable clients and servers
   to exchange information on their implementations for the purposes of
   statistical analysis and problem determination.

   This information is be submitted to a server by any client wishing to
   provide information for statistical purposes, provided the server
   advertises its willingness to take the information with the atom "ID"
   included in the list of capabilities returned by the CAPABILITY
   command.

   Implementations MUST NOT make operational changes based on the data
   sent as part of the ID command or response. The ID command is for
   human consumption only, and is not to be used in improving the
   performance of clients or servers.

   This includes, but is not limited to, the following:

      Servers MUST NOT attempt to work around client bugs by using
      information from the ID command. Clients MUST NOT attempt to work
      around server bugs based on the ID response.

      Servers MUST NOT provide features to a client or otherwise
      optimize for a particular client by using information from the ID
      command. Clients MUST NOT provide features to a server or
      otherwise optimize for a particular server based on the ID
      response.

      Servers MUST NOT deny access to or refuse service for a client
      based on information from the ID command. Clients MUST NOT refuse
      to operate or limit their operation with a server based on the ID
      response.

Rationale: It is imperative that this extension not supplant IMAP's
CAPABILITY mechanism with a ad-hoc approach where implementations
guess each other's features based on who they claim to be.

Implementations MUST NOT send false information in an ID command.

Implementations MAY send less information than they have available or
no information at all. Such behavior may be useful to preserve user
privacy. See Security Considerations, section 7.

3.1. ID Command

Arguments: client parameter list or NIL

Responses: OPTIONAL untagged response: ID

Result: OK   identification information accepted
            BAD   command unknown or arguments invalid

Implementation identification information is sent by the client with
the ID command.

This command is valid in any state.

The information sent is in the form of a list of field/value pairs.
Fields are permitted to be any IMAP4 string, and values are permitted
to be any IMAP4 string or NIL. A value of NIL indicates that the
client can not or will not specify this information. The client may
also send NIL instead of the list, indicating that it wants to send
no information, but would still accept a server response.

The available fields are defined in section 3.3.

Example: C: a023 ID ("name" "sodr" "version" "19.34" "vendor"
             "Pink Floyd Music Limited")
          S: * ID NIL
          S: a023 OK ID completed

3.2. ID Response

Contents: server parameter list

In response to an ID command issued by the client, the server replies
with a tagged response containing information on its implementation.
The format is the same as the client list.

```
Example:  C: a042 ID NIL
          S: * ID ("name" "Cyrus" "version" "1.5" "os" "sunos"
             "os-version" "5.5" "support-url"
             "mailto:cyrus-bugs+@andrew.cmu.edu")
          S: a042 OK ID command completed
```

A server MUST send a tagged ID response to an ID command. However, a
server MAY send NIL in place of the list.

3.3. Defined Field Values

Any string may be sent as a field, but the following are defined to
describe certain values that might be sent. Implementations are free
to send none, any, or all of these. Strings are not case-sensitive.
Field strings MUST NOT be longer than 30 octets. Value strings MUST
NOT be longer than 1024 octets. Implementations MUST NOT send more
than 30 field-value pairs.

```
    name                Name of the program
    version             Version number of the program
    os                  Name of the operating system
    os-version          Version of the operating system
    vendor              Vendor of the client/server
    support-url         URL to contact for support
    address             Postal address of contact/vendor
    date                Date program was released, specified as a date-time
                          in IMAP4rev1
    command             Command used to start the program
    arguments           Arguments supplied on the command line, if any
                          if any
    environment         Description of environment, i.e., UNIX environment
                          variables or Windows registry settings
```

Implementations MUST NOT use contact information to submit automatic
bug reports. Implementations may include information from an ID
response in a report automatically prepared, but are prohibited from
sending the report without user authorization.

It is preferable to find the name and version of the underlying
operating system at runtime in cases where this is possible.

Information sent via an ID response may violate user privacy. See
Security Considerations, section 7.

Implementations MUST NOT send the same field name more than once.

4. Formal Syntax

   This syntax is intended to augment the grammar specified in
   [IMAP4rev1] in order to provide for the ID command. This
   specification uses the augmented Backus-Naur Form (BNF) notation as
   used in [IMAP4rev1].

      command_any ::= "CAPABILITY" / "LOGOUT" / "NOOP" / x_command / id
         ;; adds id command to command_any in [IMAP4rev1]

      id ::= "ID" SPACE id_params_list

      id_response ::= "ID" SPACE id_params_list

      id_params_list ::= "(" #(string SPACE nstring) ")" / nil
         ;; list of field value pairs

      response_data ::= "*" SPACE (resp_cond_state / resp_cond_bye /
         mailbox_data / message_data / capability_data / id_response)

5. Use of the ID extension with Firewalls and Other Intermediaries

   There exist proxies, firewalls, and other intermediary systems that
   can intercept an IMAP session and make changes to the data exchanged
   in the session. Such intermediaries are not anticipated by the IMAP4
   protocol design and are not within the scope of the IMAP4 standard.
   However, in order for the ID command to be useful in the presence of
   such intermediaries, those intermediaries need to take special note
   of the ID command and response. In particular, if an intermediary
   changes any part of the IMAP session it must also change the ID
   command to advertise its presence.

   A firewall MAY act to block transmission of specific information
   fields in the ID command and response that it believes reveal
   information that could expose a security vulnerability. However, a
   firewall SHOULD NOT disable the extension, when present, entirely,
   and SHOULD NOT unconditionally remove either the client or server
   list.

   Finally, it should be noted that a firewall, when handling a
   CAPABILITY response, MUST NOT allow the names of extensions to be
   returned to the client that the firewall has no knowledge of.

6. References

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

   [IMAP4rev1] Crispin, M., "Internet Message Access Protocol - Version
               4rev1", RFC 2060, October 1996.

   [RFC-822]   Crocker, D., "Standard for the Format of ARPA Internet
               Text Messages", STD 11, RFC 822, August 1982.

7. Security Considerations

   This extension has the danger of violating the privacy of users if
   misused. Clients and servers should notify users that they implement
   and enable the ID command.

   It is highly desirable that implementations provide a method of
   disabling ID support, perhaps by not sending ID at all, or by sending
   NIL as the argument to the ID command or response.

   Implementors must exercise extreme care in adding fields sent as part
   of an ID command or response. Some fields, including a processor ID
   number, Ethernet address, or other unique (or mostly unique)
   identifier allow tracking of users in ways that violate user privacy
   expectations.

   Having implementation information of a given client or server may
   make it easier for an attacker to gain unauthorized access due to
   security holes.

   Since this command includes arbitrary data and does not require the
   user to authenticate, server implementations are cautioned to guard
   against an attacker sending arbitrary garbage data in order to fill
   up the ID log. In particular, if a server naively logs each ID
   command to disk without inspecting it, an attacker can simply fire up
   thousands of connections and send a few kilobytes of random data.
   Servers have to guard against this. Methods include truncating
   abnormally large responses; collating responses by storing only a
   single copy, then keeping a counter of the number of times that
   response has been seen; keeping only particularly interesting parts
   of responses; and only logging responses of users who actually log
   in.

   Security is affected by firewalls which modify the IMAP protocol
   stream; see section 5, Use of the ID Extension with Firewalls and
   Other Intermediaries, for more information.

8. Author's Address

Tim Showalter
Mirapoint, Inc.
909 Hermosa Ct.
Sunnyvale, CA 94095

EMail: tjs@mirapoint.com

9. Full Copyright Statement

Acknowledgement