<B. Grigg>
<University of Derby>
January 12, 2018

Filter Mail Retrieval Protocol

Status of this Memo

Copyright Notice

Abstract

This following document describes the requires and specifications for the creation of a FMRP server. As this is a self-contained protocol this should be the only documentation required for any implementations of the protocol outlined.

Table of Contents

1. Introduction

The purpose of Filter Mail Retrieval Protocol (FMRP) is to allow clients to interface with a server and access server based processing of mail efficiently. FMRP is a standalone protocol not

requiring any additional systems, relying only upon conditions described in the following RFC.

A important aspect of FMRP is that all mail remains on the server, and any functionality implemented should be server sided to reduce bandwidth consumption of any clients and increase the protocol flexibility when considering cross platform implementations. The protocol replies with limited Reponses, focusing primarily on the transfer of mail information than the status of the server.

2. Filter Mail Retrieval Protocol

FMRP is designed on the principle that clients are expected to do no processing other than issue commands to the server and specify “filters”. A filter is simply a parameter used to transform the server’s local storage of mails to the desired clients mail list, this clients mail list is held within server allocated resources until the client requests a retrieval of the list.

Most filters that are applied in this protocol are specified by command parameters from the client, the only filter that isn’t directly defined by command parameters from the client is filters regarding the “status” of mail. This filter is applied to the client’s mail list after the client session refreshes it mail list from the server’s mail list. This controls if a client is looking at “Inbox”, “Deleted”, or “Archived” mail. Whilst this is the only filter required from this specification, a filter allowing a client to only see mail where they are recipient of should be considered depending upon the desired implementation of FMRP.

FMRP requires no additional establishing or closing of connection other than checking if a client is connected. If a client disconnects the session should be terminated immediately and any resources of data obtained during the session could be released.

3. Procedure

This following section presents the procedures used in FMRP. All sessions must begin with client authentication, following this any of the other procedures and their associated commands may be used regardless of order. The client session following any procedure should be returned to a default state to allow for reliability and predictability.

While no encryption is expected for this RFC, it is advised all data transfer is at least encrypted in some form.

3.1. Authentication

Authentication should be the first procedure used before a client can use other commands. Authentication unlike the other procedures should not interact with any processes involving the mail and should instead be a single time process after a client server connection has been successfully established.

The LOGIN command should be used to begin the authentication process, the client should also provide the users username and password as arguments in the form shown following.

LOGIN <User> <Password>

In response to this command the server should respond with a 250OK reply if the user's credentials successfully match the server's records. Or decline access to the client prompting the user to retry the command. If the login is successful the authentication procedure should be considered complete, and the session can now be put into the default state in which any other procedure or commands can be executed at will by the client. If unsuccessful this access should be denied.

Example of FMRP Procedure

The following examples show a successful and unsuccessful execution of the associated LOGIN command.

S: LOGIN Smith@Alpha.ARPA Password123

R: Login successful

S: LOGIN Smith@Alpha.ARPA Password312

R: Login unsuccessful

End of Example 1

3.2. Retrieval

The retrieval procedure has the associated command "RETRIEVE" to which the server will return all mail that is considered to fit the client's filters defined in the arguments. The RETRIEVE command has been used in any of the following forms, each other which defines the filter to be applied to the sessions mail list before returning all mail resulting, including both headers and bodies.

RETRIEVE ALL

RETRIEVE ID <ID>

RETRIEVE RECIPIENT <RECIPIENT>

RETRIEVE SENDER <SENDER>

RETRIEVE KEYWORD <KEYWORD>

RETRIEVE SUBJECT <SUBJECT>

RETRIEVE DATE <START-DATE> <END-DATE>

Example of FMRP Procedure

The following examples show a successful and unsuccessful execution of the associated "RETRIEVE" command.

S: RETRIEVE ALL

R: <Retrieved Emails>

S: RETRIEVE ID 1

R: <Retrieved Emails>

S: RETRIEVE RECIPIENT Exampleemail@Gmail.com

R: <Retrieved Emails>

S: RETRIEVE SENDER ExampleSender@Gmaill.com

R: <Retrieved Emails>

S: RETRIEVE KEYWORD “Transaction”

R: <Retrieved Emails>

S: RETRIEVE SUBJECT Payment

R: <Retrieved Emails>

End of Example 2

3.3. Search

The search procedure has the associated command “SEARCH” to which the server will return mail headers fitting the defined parameters. The search command should not return any attachments or messages, and only the meta data of the mail(ID, Recipient, Sender, Keywords, Subject, Date). The following example shows all the valid syntax for the search command.

SEARCH ALL

SEARCH ID <ID>

SEARCH RECIPIENT <RECIPIENT>

SEARCH SENDER <SENDER>

SEARCH KEYWORD <KEYWORD>

SEARCH SUBJECT <SUBJECT>

SEARCH DATE <START-DATE> <END-DATE>

Example of FMRP Procedure

The following examples show a successful and unsuccessful execution of the associated “SEARCH” command.

S: SEARCH ALL

R: <Retrieved Emails>

S: SEARCH ID 1

R: <Retrieved Emails>

S: SEARCH RECIPIENT Exampleemail@Gmail.com

R: <Retrieved Emails>

S: SEARCH SENDER ExampleSender@Gmaill.com

R: <Retrieved Emails>

S: SEARCH KEYWORD “Transaction”

R: <Retrieved Emails>

S: SEARCH SUBJECT Payment

R: <Retrieved Emails>

End of Example 3

3.4. Domain

The domain procedure is used for changing the current domain the session is filtering by. By default, the session should be viewing the “INBOX” domain, this procedure allows the session to switch to any of domains from any other domains. The required domains are “INBOX”, “DELETED”, and “ARCHIVED”. The domain a client wishes their session to be changed to should be passed as an argument to the “CHANGEDOMAIN” command in the format shown following.

CHANGEDOMAIN <Domain>

Example of FMRP Procedure

The following examples show a successful and unsuccessful execution of the associated “SEARCH” command.

S: CHANGEDOMAIN INBOX

S: CHANGEDOMAIN DELETED

S: CHANGEDOMAIN ARCHIVED

End of Example 3

3.4.1. Mail Domain Control

The mail domain control procedure is a series of commands allowing for mail to be edited not only in the session but also the servers master list. The procedure should first update its mail before pushing any updates to the server. The server isn’t required to provide any respond to updating mail in this way. The commands associated with this procedure are “DELETE”, “ARCHIVE”, and “RESOTRE”. “DELETE” moving the item to the deleted domain, “ARCHIVE” for moving it to the archived domain, and “RESTORE” for moving an

item to the inbox domain. All of which are used in the same syntax as seen in the following form.

<COMMAND> ALL

<COMMAND> ID <ID>

<COMMAND> RECIPIENT <RECIPIENT>

<COMMAND> SENDER <SENDER>

<COMMAND> KEYWORD <KEYWORD>

<COMMAND> SUBJECT <SUBJECT>

<COMMAND> DATE <START-DATE> <END-DATE>

4. Specifications

4.1. Commands

4.1.1. Command Semantics

4.1.1.1. (LOGIN)

The login command is used to login in a user to start a session and to give them access to the other commands. Before a user has logged in successful all access should be denied to them.

4.1.1.2. (RETRIEVE)

The Retrieve command should return all mail (Both bodies and headers) matching a filter before refreshing the session list, returning to the default state.

4.1.1.3. (SEARCH)

The Retrieve command should return all mail headers matching a filter before refreshing the session list, returning to the default state.

4.1.1.4. (CHANGEDOMAIN)

The change domain command should update the client's domain filter to only show the desired domain before refreshing the session by pulling an instance of the server mail list and applying the newly chosen domain filter.

4.1.1.5. (DELETE)

The delete command set the status of all mail that fits the sessions selected filters to deleted, before then updating the servers master list.

4.1.1.6. (ARCHIVE)

The archive command set the status of all mail that fits the sessions selected filters to archived, before then updating the servers master list.

4.1.1.7. (RESTORE)

The restore command set the status of all mail that fits the sessions selected filters to inboxed, before then updating the servers master list.

4.1.2. Command Syntax

All commands and arguments should be case insensitive.

LOGIN <User> <Password>

RETRIEVE ALL

RETRIEVE ID <ID>

RETRIEVE RECIPIENT <RECIPIENT>

RETRIEVE SENDER <SENDER>

```
   RETRIEVE KEYWORD <KEYWORD>
   RETRIEVE SUBJECT <SUBJECT>
   RETRIEVE DATE <START-DATE> <END-DATE>

   SEARCH ALL
   SEARCH ID <ID>
   SEARCH RECIPIENT <RECIPIENT>
   SEARCH SENDER <SENDER>
   SEARCH KEYWORD <KEYWORD>
   SEARCH SUBJECT <SUBJECT>
   SEARCH DATE <START-DATE> <END-DATE>

   CHANGEDOMAIN <Domain>

   DELETE ALL
   DELETE ID <ID>
   DELETE RECIPIENT <RECIPIENT>
   DELETE SENDER <SENDER>
   DELETE KEYWORD <KEYWORD>
   DELETE SUBJECT <SUBJECT>
   DELETE DATE <START-DATE> <END-DATE>

   ARCHIVE ALL
   ARCHIVE ID <ID>
```

ARCHIVE RECIPIENT <RECIPIENT>

ARCHIVE SENDER <SENDER>

ARCHIVE KEYWORD <KEYWORD>

ARCHIVE SUBJECT <SUBJECT>

ARCHIVE DATE <START-DATE> <END-DATE>

RESTORE ALL

RESTORE ID <ID>

RESTORE RECIPIENT <RECIPIENT>

RESTORE SENDER <SENDER>

RESTORE KEYWORD <KEYWORD>

RESTORE SUBJECT <SUBJECT>

RESTORE DATE <START-DATE> <END-DATE>

4.1.2.1. Argument Syntax

All Arguments unless specified below should be in string format without any spaces, they must conform to UTF-8 to ensure they can be parsed correctly.

4.1.2.1.1.

<ID> should be parse able as an integer value.

<Keyword> should follow the previously mentioned standard however should be enclosed by quotation marks such as “keyword”.

<START-DATE> should be in the format dd/mm/yyyy.

<END-DATE> should be in the format dd/mm/yyyy.

5. Security Considerations

Due to the nature of this system, the most venerable vector for any attack would likely to be to intercept the traffic between the client and the server, it is for this reason it is suggested in other implementations to include encryption of all traffic to allow user data to remain safe.

Another potential security concern is problems caused by several users changing the status of their mail at the same time, as this requires a complete update of the server mail list, resulting in a potential race condition if implemented without safe guards, this would likely result in lost mail.

6. References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[2] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2234] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997.