Network Working Group   M. Crispin
INTERNET DRAFT: Message Submission   University of Washington
Document: draft-crispin-lemonade-pull-01.txt   February 2004

Message Submission

Status of this Memo

Abstract

This document describes a set of strategies to allow clients to submit new email messages incorporating content which resides on locations external to the client. This is the so-called "IMAP Pull" approach currently being considered by the LEMONADE working group as one solution to the "forward without download" problem (that is, as a means for clients to send new messages consisting of or containing all or parts of previously received messages without having to download and upload the data). Some of the strategies in this document rely upon extensions to other protocols; specifically URLAUTH and CATENATE in the IMAP protocol (RFC 3501) and BURL in the SUBMIT protocol (RFC 2476).

This document describes the strategies in multiple variants, as well as the implications on authorization and the "fcc problem".

Conventions Used in this Document

The key words "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", and "MAY" in this document are to be interpreted as described in [KEYWORDS].

The formal syntax use the Augmented Backus-Naur Form (ABNF) notation including the core rules defined in Appendix A of [ABNF].

In examples, "C:" and "S:" indicate lines sent by the client and server respectively. If a single "C:" or "S:" label applies to multiple lines, then the line breaks between those lines are for editorial clarity only and are not part of the actual protocol exchange.

Introduction

The advent of client/server email using the [IMAP] and [SMTP] protocols has changed what formerly were local disk operations to become excessive and repetitive network data transmissions. As noted in [LEMONADE], this is an onerous burden for clients operating over low-bandwidth or high-latency links.

Two major approach were identified in [LEMONADE] to address this problem: "pull" and "push". The strategy used to implement the push approach is the subject of another document ([PUSH]). This document discusses the pull approach, and describes three strategies in which the pull approach can be implemented.

All three of the pull strategies discussed below make use of the [BURL] SUBMIT extension to enable access to external sources during the submission of a message. In combination with the IMAP [URLAUTH] extension, inclusion of message parts or even entire messages from the IMAP mail store is possible with a minimal trust relationship between the IMAP and SUBMIT servers.

Pull has the distinct advantage of maintaining one submission protocol, and thus avoids the risk of having multiple parallel and possible divergent mechanisms for submission. Furthermore, by keeping the details of message submission in the SUBMIT server, the pull strategies are prepared to work with other message retrieval protocols such as POP, NNTP, or whatever else may be designed in the future.

Message Sending Overview

The act of sending an email message is best thought of as involving multiple steps: initiation of a new draft, draft editing, message assembly, and message submission.

Initiation of a new draft and draft editing take place on the MUA. Frequently, users choose to save more complex and/or time-consuming messages on an [IMAP] server (via the APPEND command with the \Draft flag) for later recall by the MUA and resumption of the editing process.

Message assembly is the process of producing a complete message from the final revision of the draft and external sources. At assembly time, external data is retrieved and inserted in the message.

Message submission is the process of inserting the assembled message into the [SMTP] infrastructure, typically using the [SUBMIT] protocol.

It should be noted that another way of handling external data is to send a pointer to the data, rather than the data itself, to the recipient. To be useful, this pointer must be something that the recipient can resolve and has authorization to reference. Common examples of this are the MESSAGE/EXTERNAL-BODY subtype defined in [MEDIA] and the practice of including public-access URLs in message texts.

Traditional Strategy

Traditionally, messages are initiated, edited, and assembled entirely within an MUA, although drafts may be saved to an [IMAP] server and later retrieved from the server. The completed text is then transmitted to an MSA for delivery.

There is often no clear boundary between the editing and assembly process. If a message is forwarded, its content is often retrieved immediately and inserted into the message text. Similarly, once external content is inserted or attached, the content is usually retrieved immediately and made part of the draft.

As a consequence, each save of a draft and subsequent retrieve of the draft transmits that entire (possibly large) content, as does message submission.

In the past, this was not much of a problem, because drafts, external data, and the message submission mechanism were typically located on the same system as the MUA. The most common problem was running out of disk quota.

[PUSH] Strategy

The [PUSH] strategy is the topic of another document, but bears brief mention for comparison purposes. In the [PUSH] strategy, messages are initiated and edited within an MUA. The proposed [CATENATE] extension to [IMAP] is then used to upload the message to the IMAP server and assemble it, and finally the proposed SUBMIT extension in [PUSH] is used to pass the message to an MSA.

[PUSH] explicitly limits external content to other messages in the currently selected mailbox on the IMAP server, although there is some discussion about allowing other content in [CATENATE] via URLs. The result of such an extension could be thought of as a hybrid pull-push model, since the IMAP server is used both to pull external references and push the resulting message to the MSA.

From the point of view of the "fcc problem" (see below) an advantage (and disadvantage) of this strategy is that a fully-assembled copy of the message is left on the IMAP server.

Pull Strategy -- [BURL] Variant

In the [BURL] variant of the pull strategy, messages are initiated and edited within an MUA. The MUA then uploads the message to a [SUBMIT] server with the [BURL] extension. Among the types of URIs that can be passed to the [BURL] command are [URLAUTH] format URLs, this satisfying the [LEMONADE] forward-without-download requirement.

Note that the assembly and submission operations are done in a single step, as opposed to the two-step process in the [PUSH] strategy; and that external content is not limited to other messages in the currently selected mailbox on the IMAP server.

The main disadvantage of this variant is that it does not address the "fcc problem" (see below) intrinsically, although it is suitable with a generalized solution to the fcc problem.

Pull Strategy -- [BURL]/[CATENATE] Variant

In the [BURL]/[CATENATE] variant of the pull strategy, messages are initiated and edited within an MUA. The proposed [CATENATE] extension to [IMAP] is then used to upload the message to the IMAP server and assemble it, and finally a [URLAUTH] format URL is given to a [SUBMIT] server with the [BURL] extension for submission.

If this strategy is chosen over the pure [BURL] variant, it is possible to simplify the [BURL] specification so that a single BURL command defines the message data without any BDAT.

This is another form of a hybrid pull-push model. As with the [PUSH] strategy, from the point of view of the "fcc problem" (see below) an advantage (and disadvantage) of this strategy is that a fully-assembled copy of the message is left on the IMAP server.

As noted under the [PUSH] strategy, to be fully useful the proposed [CATENATE] extension has to allow arbitrary URLs.

Pull Strategy -- [BURL]/Embedded URI Variant

In the embedded URI variant of the pull strategy, messages are initiated and edited within an MUA, and saved on the [IMAP] server via an APPEND command. External data is indicated by URIs which are embedded in the message in some easily-identifiable form, perhaps as an added parameter to the [MIME] Content-Type header. A [URLAUTH] format URL is given to a [SUBMIT] server with the [BURL] extension for assembly and submission.

If this strategy is chosen over the pure [BURL] variant, it is possible to simplify the [BURL] specification so that a single BURL command defines the message data without any BDAT.

The advantage of this strategy is that the client does not deal with message assembly at all; the entire burden is on the [SUBMIT] server.

This is the only strategy which is specifically draft-friendly, since drafts with pointers to the external data are suitable for submission without requiring the MUA to break up the draft between texts and pointers for message assembly. The client can simply download the draft from the IMAP server and re-edit it, without the burden of downloading the external data (since it hasn't yet been incorporated into the message) or fetching the message structure, parsing it, and then fetching the editable parts.

From the point of view of the "fcc problem" (see below) an advantage (and disadvantage) of this strategy is that a copy of the message with references to external content (but not the external content itself) is left on the IMAP server.

Authorization Strategies

In any of these strategies, authorization is necessary. It should be noted that in all strategies (including [PUSH]), the [SUBMIT] server is trusted with the data.

In the traditional strategy, authorization comes from authentication; the client authenticates to the [IMAP] server as a userid that is authorized to access the data, fetches the data from the [IMAP] server, authenticates to the [SUBMIT] server as a userid that is authorized to submit a message, and then sends the data to the [SUBMIT] server.

The [PUSH] strategy multiplexes authorization from a single credential derived from authentication to the [IMAP] server; that credential is assumed to be sufficient to access all data and to submit the data. There is no provision in [PUSH] for differential authorization; for example, it is not possible in [PUSH] to have authentication to the [IMAP] server using a general "role" account (e.g., "helpdesk") for access authorization, but submit as a specific individual.

The Pull strategies allow a wide range of authorization mechanisms to obtain data from the pulled URI:

(1) The simplest mechanism is a public-access URI. External data is commonly handled today this way, e.g., the MESSAGE/EXTERNAL-BODY subtype defined in [MEDIA] and the practice of including public-access URLs in message texts.

(2) Another simple mechanism is one in which the [IMAP] server trusts the [SUBMIT] server. The [SUBMIT] server is authenticated by the [IMAP] server by one of several means (e.g., IP address, Kerberos, SSL/TLS certificate validation, or special account), and is subsequently authorized by the [IMAP] server to access any data on the server. This is the trust model that the [PUSH] model uses, only in reverse.

(3) A variant of the trust mechanism is one in which the [SUBMIT] server uses a [SASL] authentication identity on the [IMAP] server that is permitted to use another identity as the authorization identity. The authorization identity passed to the [IMAP] server is the authentication identity used by the client to access the [SUBMIT] server. The advantage of this variant is that the [SUBMIT] server is only responsible for asserting the client's identity but not the client's authorization. Many existing webmail systems work this way today.

(4) The so-called "pawn-ticket" authorization mechanism uses a URI which contains its own authorization credentials using a scheme such as [URLAUTH]. The advantage of this mechanism is that the [SUBMIT] server can not access any data on the [IMAP] server without a "pawn-ticket" created by the client. The "pawn-ticket" grants acces only to the specific data that the [SUBMIT] server is authorized to access, can be revoked by the client, and can have a time-limited validity.

(5) A variant of the "pawn-ticket" authorization mechanism adds a requirement that a trust relationship, as described in mechanism (2) above, be established in order for the pawn-ticket to be accepted. The advantage of this variant is that even if an attacker learns the "pawn-ticket", it is useless to him because the attacker does not have the trust relationship. The AUTHID and AUTHROLE extensions in [URLAUTH] provide a means for doing this.

Of the above mechanisms, (5) is recommended, since it requires that an attacker must both discover a "pawn-ticket" and use the trust relationship of the [SUBMIT] server.

The "pawn-ticket" can be further secured by using the optional userid argument in AUTHROLE=compose in a [URLAUTH] format URL. For example, if example.com's [SUBMIT] server is given a URL of

```
   imap://joe@example.com/INBOX;uid=20;section=1.2;authrole=submit:smith
```

then the [SUBMIT] server MUST require that smith is the userid authenticated to the [SUBMIT] server. This allows the [SUBMIT] and the [IMAP] servers to be in different authentication realms; in the above example, the user is "joe" on the [IMAP] server and "smith" on the [SUBMIT] server.

Note that, even if the user doesn't authenticate to the [SUBMIT] server, there is no added risk over the push strategy. The "pawn-ticket" ensures authorized access to the desired data, and only that data.

The fcc Problem

The "fcc problem" refers to a frequent need to deliver a copy of the message to a "file carbon copy" receeipient. By far, the most common case of fcc is a client leaving a copy of outgoing mail in a "sent messages" or "outbox" mailbox.

In the traditional strategy, the MUA duplicates the effort spent in transmitting to the MSA by writing the message to the fcc destination in a separate step. This may be a write to a local disk file or an APPEND to a mailbox on an IMAP server. The latter is one of the "excessive and repretitive network data transmissions" which [LEMONADE] seeks to reduce or eliminate; and represents the "problem" aspect of the "fcc problem".

The proposed [CATENATE] extension to [IMAP] is a potentional solution to the fcc problem. It requires making several simplifying assumptions:
 (1a) there is one, and only one, fcc destination on a single server
 (2a) the server which holds the fcc is the same as the server which
       stages the outgoing message for submission
 (3a) it is desired that the fcc be a copy of the complete message text
       with all external data inserted in the message
As noted above, [CATENATE] is used by the [PUSH] strategy and by the [BURL]/[CATENATE] variant of the pull strategy.

The Embedded URI variant of the pull strategy represents an alternative potentional solution to the fcc problem. It requires making several simplifying assumptions:
 (1b) there is one, and only one, fcc destination on a single server
 (2b) the server which holds the fcc is the same as the server which
       stages the outgoing message for submission
 (3b) it is desired that the fcc be a copy of the message with all
       external data pointed to by URI references.

It can be noted that the first two simplifying assumptions are the same in both potential solutions. The difference is in the third assumption, which are complete opposites. Thus, the preferred solution to the fcc problems depends upon one's preferences in the third assumption.

Neither of these two potential solutions are satisfactory if the first two simplifying assumptions can not be made, or if it is a requirement that the third assumption be a matter of user preference. The developers of at least one MUA (Pine) found that it is very important for users to control whether the sent messages mailbox contains external data.

A more general solution to the fcc problem is possible with the [BURL] and Embedded URI variants of the pull strategy. The agent which does the message assembly is given the fcc (or fccs) as part of the list of message recipients, along with authorization information and whether the fcc should include external data or pointers. This simplifies to "no action needed" in the Embedded URI variant if the user wants pointers in the fcc.

Security Considerations

Security considerations are discussed throughout this memo.

Acknowledgements

Thanks to Randall Gellens and Rob Siemborski for numerous helpful suggestions.

References

The following references are normative:

[ABNF] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[BURL] Newman, C., "Message Composition", draft-newman-lemonade-compose-00.txt (work in progress), June 2003.

[CATENATE] Resnick, P., "Internet Message Access Protocol (IMAP) CATENATE Extension", draft-ietf-lemonade-catenate-01 (work in progress), January 2004.

[IMAP] Crispin, M., "Internet Message Access Protocol - Version 4rev1", RFC 3501, March 2003.

[IMAPURL] Newman, C., "IMAP URL Scheme", RFC 2192, September 1997.

[LEMONADE] Gellens, R., "IMAP Submit Without Download", draft-ietf-lemonade-submit-01.txt (work in progress), October 2003.

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

[MEDIA] Freed, N. and N. Borenstein, "MIME (Multipurpose Internet Mail Extensions) Part Two: Media Types", RFC 2046, November 1996.

[MIME] Freed, N. and N. Borenstein, "MIME (Multipurpose Internet Mail Extensions) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[SASL] Myers, J., "Simple Authentication and Security Layer (SASL)", RFC 2222, October 1997.

[SUBMIT]   Gellens, R. and Klensin, J., "Message Submission", RFC 2476, December 1998.

The following references are informative:

[PUSH]   Gellens, R., "IMAP Message Submission", draft-gellens-lemonade-push-00.txt (work in progress), December 2003.

[SMTP]   Klensin, J., "Simple Mail Transfer Protocol", RFC 2821, April 2001.

Author's Address

Mark R. Crispin
Networks and Distributed Computing
University of Washington
4545 15th Avenue NE
Seattle, WA 98105-4527

Phone: (206) 543-5762
EMail: MRC@CAC.Washington.EDU

Intellectual Property Statement

Full Copyright Statement

Acknowledgement

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