Network Working Group
Request for Comments: 4468
Updates: 3463
Category: Standards Track

C. Newman
Sun Microsystems

May 2006

Message Submission BURL Extension

Status of This Memo

Copyright Notice

Abstract

The submission profile of Simple Mail Transfer Protocol (SMTP) provides a standard way for an email client to submit a complete message for delivery. This specification extends the submission profile by adding a new BURL command that can be used to fetch submission data from an Internet Message Access Protocol (IMAP) server. This permits a mail client to inject content from an IMAP server into the SMTP infrastructure without downloading it to the client and uploading it back to the server.

Table of Contents

1. Introduction

This specification defines an extension to the standard Message Submission [RFC4409] protocol to permit data to be fetched from an IMAP server at message submission time. This MAY be used in conjunction with the CHUNKING [RFC3030] mechanism so that chunks of the message can come from an external IMAP server. This provides the ability to forward an email message without first downloading it to the client.

2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", and "MAY" in this document are to be interpreted as defined in "Key words for use in RFCs to Indicate Requirement Levels" [RFC2119].

The formal syntax uses the Augmented Backus-Naur Form (ABNF) [RFC4234] notation including the core rules defined in Appendix B of RFC 4234.

3. BURL Submission Extension

This section defines the BURL submission extension.

3.1. SMTP Submission Extension Registration

1. The name of this submission extension is "BURL". This extends the Message Submission protocol on port 587 and MUST NOT be advertised by a regular SMTP [RFC2821] server on port 25 that acts as a relay for incoming mail from other SMTP relays.

2. The EHLO keyword value associated with the extension is "BURL".

3. The BURL EHLO keyword will have zero or more arguments. The only argument defined at this time is the "imap" argument, which MUST be present in order to use IMAP URLs with BURL. Clients MUST ignore other arguments after the BURL EHLO keyword unless they are defined by a subsequent IETF standards track specification. The arguments that appear after the BURL EHLO keyword may change subsequent to the use of SMTP AUTH [RFC2554], so a server that advertises BURL with no arguments prior to authentication indicates that BURL is supported but authentication is required to use it.

4. This extension adds the BURL SMTP verb. This verb is used as a replacement for the DATA command and is only permitted during a mail transaction after at least one successful RCPT TO.

3.2. BURL Transaction

A simple BURL transaction will consist of MAIL FROM, one or more RCPT TO headers, and a BURL command with the "LAST" tag. The BURL command will include an IMAP URL pointing to a fully formed message ready for injection into the SMTP infrastructure. If PIPELINING [RFC2920] is advertised, the client MAY send the entire transaction in one round trip. If no valid RCPT TO address is supplied, the BURL command will simply fail, and no resolution of the BURL URL argument will be performed. If at least one valid RCPT TO address is supplied, then the BURL URL argument will be resolved before the server responds to the command.

A more sophisticated BURL transaction MAY occur when the server also advertises CHUNKING [RFC3030]. In this case, the BURL and BDAT commands may be interleaved until one of them terminates the transaction with the "LAST" argument. If PIPELINING [RFC2920] is also advertised, then the client may pipeline the entire transaction in one round-trip. However, it MUST wait for the results of the "LAST" BDAT or BURL command prior to initiating a new transaction.

The BURL command directs the server to fetch the data object to which the URL refers and include it in the message. If the URL fetch fails, the server will fail the entire transaction.

3.3. The BURL IMAP Options

When "imap" is present in the space-separated list of arguments following the BURL EHLO keyword, it indicates that the BURL command supports the URLAUTH [RFC4467] extended form of IMAP URLs [RFC2192] and that the submit server is configured with the necessary credentials to resolve "urlauth=submit+" IMAP URLs for the submit server's domain.

Subsequent to a successful SMTP AUTH command, the submission server MAY indicate a prearranged trust relationship with a specific IMAP server by including a BURL EHLO keyword argument of the form "imap://imap.example.com". In this case, the submission server will permit a regular IMAP URL referring to messages or parts of messages on imap.example.com that the user who authenticated to the submit server can access. Note that this form does not imply that the submit server supports URLAUTH URLs; the submit server must advertise both "imap" and "imap://imap.example.com" to indicate support for both extended and non-extended URL forms.

When the submit server connects to the IMAP server, it acts as an IMAP client and thus is subject to both the mandatory-to-implement IMAP capabilities in Section 6.1.1 of RFC 3501, and the security considerations in Section 11 of RFC 3501. Specifically, this requires that the submit server implement a configuration that uses STARTTLS followed by SASL PLAIN [SASL-PLAIN] to authenticate to the IMAP server.

When the submit server resolves a URLAUTH IMAP URL, it uses submit server credentials when authenticating to the IMAP server. The authentication identity and password used for submit credentials MUST be configurable. The string "submit" is suggested as a default value for the authentication identity, with no default for the password. Typically, the authorization identity is empty in this case; thus the IMAP server will derive the authorization identity from the authentication identity. If the IMAP URL uses the "submit+" access identifier prefix, the submit server MUST refuse the BURL command unless the userid in the URL's <access> token matches the submit client's authorization identity.

When the submit server resolves a regular IMAP URL, it uses the submit client's authorization identity when authenticating to the IMAP server. If both the submit client and the submit server's embedded IMAP client use SASL PLAIN (or the equivalent), the submit

server SHOULD forward the client's credentials if and only if the submit server knows that the IMAP server is in the same administrative domain. If the submit server supports SASL mechanisms other than PLAIN, it MUST implement a configuration in which the submit server's embedded IMAP client uses STARTTLS and SASL PLAIN with the submit server's authentication identity and password (for the respective IMAP server) and the submit client's authorization identity.

3.4. Examples

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.

Two successful submissions (without and with pipelining) follow:

```
<SSL/TLS encryption layer negotiated>
C: EHLO potter.example.com
S: 250-owlry.example.com
S: 250-8BITMIME
S: 250-BURL imap
S: 250-AUTH PLAIN
S: 250-DSN
S: 250 ENHANCEDSTATUSCODES
C: AUTH PLAIN aGFycnkAaGFycnkAYWNjaW8=
S: 235 2.7.0 PLAIN authentication successful.
C: MAIL FROM:<harry@gryffindor.example.com>
S: 250 2.5.0 Address Ok.
C: RCPT TO:<ron@gryffindor.example.com>
S: 250 2.1.5 ron@gryffindor.example.com OK.
C: BURL imap://harry@gryffindor.example.com/outbox
       ;uidvalidity=1078863300/;uid=25;urlauth=submit+harry
       :internal:91354a473744909de610943775f92038 LAST
S: 250 2.5.0 Ok.

<SSL/TLS encryption layer negotiated>
C: EHLO potter.example.com
S: 250-owlry.example.com
S: 250-8BITMIME
S: 250-PIPELINING
S: 250-BURL imap
S: 250-AUTH PLAIN
S: 250-DSN
S: 250 ENHANCEDSTATUSCODES
C: AUTH PLAIN aGFycnkAaGFycnkAYWNjaW8=
```

```
C: MAIL FROM:<harry@gryffindor.example.com>
C: RCPT TO:<ron@gryffindor.example.com>
C: BURL imap://harry@gryffindor.example.com/outbox
       ;uidvalidity=1078863300/;uid=25;urlauth=submit+harry
       :internal:91354a473744909de610943775f92038 LAST
S: 235 2.7.0 PLAIN authentication successful.
S: 250 2.5.0 Address Ok.
S: 250 2.1.5 ron@gryffindor.example.com OK.
S: 250 2.5.0 Ok.
```

Note that PIPELINING of the AUTH command is only permitted if the selected mechanism can be completed in one round trip, a client initial response is provided, and no SASL security layer is negotiated. This is possible for PLAIN and EXTERNAL, but not for most other SASL mechanisms.

Some examples of failure cases:

```
C: MAIL FROM:<harry@gryffindor.example.com>
C: RCPT TO:<malfoy@slitherin.example.com>
C: BURL imap://harry@gryffindor.example.com/outbox
       ;uidvalidity=1078863300/;uid=25;urlauth=submit+harry
       :internal:91354a473744909de610943775f92038 LAST
S: 250 2.5.0 Address Ok.
S: 550 5.7.1 Relaying not allowed: malfoy@slitherin.example.com
S: 554 5.5.0 No recipients have been specified.

C: MAIL FROM:<harry@gryffindor.example.com>
C: RCPT TO:<ron@gryffindor.example.com>
C: BURL imap://harry@gryffindor.example.com/outbox
       ;uidvalidity=1078863300/;uid=25;urlauth=submit+harry
       :internal:71354a473744909de610943775f92038 LAST
S: 250 2.5.0 Address Ok.
S: 250 2.1.5 ron@gryffindor.example.com OK.
S: 554 5.7.0 IMAP URL authorization failed
```

3.5. Formal Syntax

The following syntax specification inherits ABNF [RFC4234] and Uniform Resource Identifiers [RFC3986].

```
burl-param  = "imap" / ("imap://" authority)
            ; parameter to BURL EHLO keyword

burl-cmd    = "BURL" SP absolute-URI [SP end-marker] CRLF

end-marker  = "LAST"
```

4. 8-Bit and Binary

A submit server that advertises BURL MUST also advertise 8BITMIME [RFC1652] and perform the down conversion described in that specification on the resulting complete message if 8-bit data is received with the BURL command and passed to a 7-bit server. If the URL argument to BURL refers to binary data, then the submit server MAY refuse the command or down convert as described in Binary SMTP [RFC3030].

The Submit server MAY refuse to accept a BURL command or combination of BURL and BDAT commands that result in un-encoded 8-bit data in mail or MIME [RFC2045] headers. Alternatively, the server MAY accept such data and down convert to MIME header encoding [RFC2047].

5. Updates to RFC 3463

SMTP or Submit servers that advertise ENHANCEDSTATUSCODES [RFC2034] use enhanced status codes defined in RFC 3463 [RFC3463]. The BURL extension introduces new error cases that that RFC did not consider. The following additional enhanced status codes are defined by this specification:

X.6.6 Message content not available

> The message content could not be fetched from a remote system. This may be useful as a permanent or persistent temporary notification.

X.7.8 Trust relationship required

> The submission server requires a configured trust relationship with a third-party server in order to access the message content.

6. Response Codes

This section includes example response codes to the BURL command. Other text may be used with the same response codes. This list is not exhaustive, and BURL clients MUST tolerate any valid SMTP response code. Most of these examples include the appropriate enhanced status code [RFC3463].

554 5.5.0 No recipients have been specified

> This response code occurs when BURL is used (for example, with PIPELINING) and all RCPT TOs failed.

503 5.5.0 Valid RCPT TO required before BURL

This response code is an alternative to the previous one when BURL is used (for example, with PIPELINING) and all RCPT TOs failed.

554 5.6.3 Conversion required but not supported

This response code occurs when the URL points to binary data and the implementation does not support down conversion to base64. This can also be used if the URL points to message data with 8-bit content in headers and the server does not down convert such content.

554 5.3.4 Message too big for system

The message (subsequent to URL resolution) is larger than the per-message size limit for this server.

554 5.7.8 URL resolution requires trust relationship

The submit server does not have a trust relationship with the IMAP server specified in the URL argument to BURL.

552 5.2.2 Mailbox full

The recipient is local, the submit server supports direct delivery, and the recipient has exceeded his quota and any grace period for delivery attempts.

554 5.6.6 IMAP URL resolution failed

The IMAP URLFETCH command returned an error or no data.

250 2.5.0 Waiting for additional BURL or BDAT commands

A BURL command without the "LAST" modifier was sent. The URL for this BURL command was successfully resolved, but the content will not necessarily be committed to persistent storage until the rest of the message content is collected. For example, a Unix server may have written the content to a queue file buffer, but may not yet have performed an fsync() operation. If the server loses power, the content can still be lost.

451 4.4.1 IMAP server unavailable

The connection to the IMAP server to resolve the URL failed.

250 2.5.0 Ok.

The URL was successfully resolved, and the complete message data has been committed to persistent storage.

250 2.6.4 MIME header conversion with loss performed

The URL pointed to message data that included mail or MIME headers with 8-bit data. This data was converted to MIME header encoding [RFC2047], but the submit server may not have correctly guessed the unlabeled character set.

7. IANA Considerations

The "BURL" SMTP extension as described in Section 3 has been registered. This registration has been marked for use by message submission [RFC4409] only in the registry.

8. Security Considerations

Modern SMTP submission servers often include content-based security and denial-of-service defense mechanisms such as virus filtering, size limits, server-generated signatures, spam filtering, etc. Implementations of BURL should fetch the URL content prior to application of such content-based mechanisms in order to preserve their function.

Clients that generate unsolicited bulk email or email with viruses could use this mechanism to compensate for a slow link between the client and submit server. In particular, this mechanism would make it feasible for a programmable cell phone or other device on a slow link to become a significant source of unsolicited bulk email and/or viruses. This makes it more important for submit server vendors implementing BURL to have auditing and/or defenses against such denial-of-service attacks including mandatory authentication, logging that associates unique client identifiers with mail transactions, limits on reuse of the same IMAP URL, rate limits, recipient count limits, and content filters.

Transfer of the URLAUTH [RFC4467] form of IMAP URLs in the clear can expose the authorization token to network eavesdroppers. Implementations that support such URLs can address this issue by using a strong confidentiality protection mechanism. For example, the SMTP STARTTLS [RFC3207] and the IMAP STARTTLS [RFC3501] extensions, in combination with a configuration setting that requires their use with such IMAP URLs, would address this concern.

Use of a prearranged trust relationship between a submit server and a specific IMAP server introduces security considerations. A compromise of the submit server should not automatically compromise all accounts on the IMAP server, so trust relationships involving super-user proxy credentials are strongly discouraged. A system that requires the submit server to authenticate to the IMAP server with submit credentials and subsequently requires a URLAUTH URL to fetch any content addresses this concern. A trusted third party model for proxy credentials (such as that provided by Kerberos 5 [RFC4120]) would also suffice.

When a client uses SMTP STARTTLS to send a BURL command that references non-public information, there is a user expectation that the entire message content will be treated confidentially. To address this expectation, the message submission server SHOULD use STARTTLS or a mechanism providing equivalent data confidentiality when fetching the content referenced by that URL.

A legitimate user of a submit server may try to compromise other accounts on the server by providing an IMAP URLAUTH URL that points to a server under that user's control that is designed to undermine the security of the submit server. For this reason, the IMAP client code that the submit server uses must be robust with respect to arbitrary input sizes (including large IMAP literals) and arbitrary delays from the IMAP server. Requiring a prearranged trust relationship between a submit server and the IMAP server also addresses this concern.

An authorized user of the submit server could set up a fraudulent IMAP server and pass a URL for that server to the submit server. The submit server might then contact the fraudulent IMAP server to authenticate with submit credentials and fetch content. There are several ways to mitigate this potential attack. A submit server that only uses submit credentials with a fixed set of trusted IMAP servers will not be vulnerable to exposure of those credentials. A submit server can treat the IMAP server as untrusted and include defenses for buffer overflows, denial-of-service slowdowns, and other potential attacks. Finally, because authentication is required to use BURL, it is possible to keep a secure audit trail and use that to detect and punish the offending party.

9. References

9.1. Normative References

[RFC1652] Klensin, J., Freed, N., Rose, M., Stefferud, E., and D. Crocker, "SMTP Service Extension for 8bit-MIMEtransport", RFC 1652, July 1994.

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

[RFC2192] Newman, C., "IMAP URL Scheme", RFC 2192, September 1997.

[RFC2554] Myers, J., "SMTP Service Extension for Authentication", RFC 2554, March 1999.

[RFC2821] Klensin, J., "Simple Mail Transfer Protocol", RFC 2821, April 2001.

[RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, February 2002.

[RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, March 2003.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC4234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

[RFC4409] Gellens, R. and J. Klensin, "Message Submission for Mail", RFC 4409, April 2006.

[RFC4467] Crispin, M., "Internet Message Access Protocol (IMAP) - URLAUTH Extension", RFC 4467, May 2006.

9.2. Informative References

[RFC2034] Freed, N., "SMTP Service Extension for Returning Enhanced Error Codes", RFC 2034, October 1996.

[RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[RFC2047] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, November 1996.

[RFC2920] Freed, N., "SMTP Service Extension for Command Pipelining", STD 60, RFC 2920, September 2000.

[RFC3030] Vaudreuil, G., "SMTP Service Extensions for Transmission of Large and Binary MIME Messages", RFC 3030, December 2000.

[RFC3463] Vaudreuil, G., "Enhanced Mail System Status Codes", RFC 3463, January 2003.

[RFC4120] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005.

[SASL-PLAIN] Zeilenga, K., "The Plain SASL Mechanism", Work in Progress, March 2005.

Appendix A. Acknowledgements

This document is a product of the lemonade WG. Many thanks are due to all the participants of that working group for their input. Mark Crispin was instrumental in the conception of this mechanism. Thanks to Randall Gellens, Alexey Melnikov, Sam Hartman, Ned Freed, Dave Cridland, Peter Coates, and Mark Crispin for review comments on the document. Thanks to the RFC Editor for correcting the author's grammar mistakes. Thanks to Ted Hardie, Randall Gellens, Mark Crispin, Pete Resnick, and Greg Vaudreuil for extremely interesting debates comparing this proposal and alternatives. Thanks to the lemonade WG chairs Eric Burger and Glenn Parsons for concluding the debate at the correct time and making sure this document got completed.

Author's Address

Chris Newman
Sun Microsystems
3401 Centrelake Dr., Suite 410
Ontario, CA 91761
US

EMail: chris.newman@sun.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).