dispatch
Internet-Draft
Intended status: Informational
Expires: 30 September 2026

P.P.V. Rajendra
29 March 2026

Policy-Controlled Handling of URLs in MIME Messages
draft-vinnakota-mime-url-policy-00

Abstract

This document defines a MIME-based mechanism for policy-controlled handling of URLs in email messages. The mechanism provides an alternative to hyperlink rewriting techniques used for click-time security enforcement. By avoiding modification of original URLs and introducing policy metadata, this approach improves interoperability, preserves message integrity, and enhances privacy.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 30 September 2026.

Table of Contents

1. Conventions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119, and RFC 8174 when, and only when, they appear in all capitals, as shown here.

2. Introduction

Email security systems commonly employ hyperlink rewriting to enforce click-time protection against malicious content. While effective in certain scenarios, this approach introduces complexity, interoperability issues, and privacy concerns.

This document proposes a MIME-based mechanism that enables policy-controlled handling of URLs without modifying the original message content.

Existing mechanisms such as message/external-body (RFC 2046) allow referencing external content but do not provide policy-controlled access or security enforcement.

3. Problem Statement

3.1. Reduced User Interpretability

Hyperlink rewriting replaces original URLs with transformed representations that are often significantly longer and less human-readable. This reduces the ability of users to inspect and verify link destinations.

3.2. Privacy and Metadata Exposure

Rewritten URLs may embed identifiers and routing metadata. Accessing such URLs may expose interaction data to intermediary systems.

URLs containing query parameters may include sensitive data such as tokens or identifiers, increasing the risk of exposure through logs or intermediaries.

3.3. Functional Interference

Rewriting may interfere with application behavior, including:

* One-time-use links (for example, password reset links)

* Token-based authentication flows

* URL fragments or client-side routing

Automated analysis systems may access such links prior to user interaction, potentially invalidating time-sensitive tokens.

3.4. Access Latency

Users may experience delays due to click-time analysis, affecting usability and productivity.

3.5. Evasion Techniques

Attackers may employ redirection chains or delayed activation techniques to bypass static or pre-access analysis.

3.6. Long-Term Stability

Rewritten URLs depend on intermediary infrastructure, including vendor-specific domains and token formats. Changes in such systems may cause previously delivered links to become non-functional over time.

3.7. Internationalization and Encoding Issues

Hyperlink rewriting may introduce errors when processing internationalized resource identifiers (IRIs) and Unicode-based URLs.

Transformations applied during rewriting, including encoding, normalization, or wrapping, may alter the original URL representation and lead to invalid or inaccessible resources in practice.

Such issues are particularly prevalent when URLs contain non-ASCII characters, percent-encoded sequences, or language-specific content.

By preserving the original URL without modification, the mechanism defined in this document avoids such interoperability issues.

4. Design Goals

The mechanism defined in this document aims to:

* Avoid modification of original URLs
* Preserve compatibility with existing email systems
* Support internationalized resource identifiers without transformation
* Minimize exposure of sensitive data
* Enable policy-controlled access by recipient domains
* Improve long-term reliability of message-contained references
* Preserve exact URL encoding and representation, including internationalized and Unicode URLs

5. Overview

This document defines an optional MIME mechanism for associating policy metadata with URLs contained in email messages.

If the mechanism is present, recipient systems MAY apply policy-controlled handling to URLs during user interaction.

In typical deployments, the URL-Policy and URL-Policy-Token header fields are inserted or validated by the recipient domain’s mail handling infrastructure. Sender-provided values MUST NOT be assumed to be authoritative without validation.

If the URL-Policy header field is present, it applies to all URLs contained within the associated message.

Enforcement of policy-controlled URL handling is performed by recipient-side software and not by the MIME header fields themselves.

6. MIME Header Field Definitions

6.1. URL-Policy

The URL-Policy header field specifies a policy-controlled endpoint.

If present, this field indicates that URLs contained in the message MAY be evaluated through a policy-controlled endpoint determined by the recipient domain.

Example:

URL-Policy: https://policy.example.com/url-check

6.2. URL-Policy-Token

The URL-Policy-Token header field contains an opaque identifier generated by the recipient domain.

The token is used to correlate message-specific context with the policy endpoint.

Example:

URL-Policy-Token: 4f9e7c12-8d11-47b1-a9c5-2d43c1f66e20

7. Processing Model

When a user activates a URL in a message containing the URL-Policy header field:

1. The client or recipient system identifies the presence of policy metadata.

2. The original URL remains unchanged.

3. The client or system MAY submit the URL and policy token to the policy endpoint.

4. The policy endpoint evaluates the request.

5. The endpoint returns a decision (for example, allow, block, or warn).

6. The original URL MUST NOT be modified as part of this process.

8. Example Flow

A message is received with the following header fields:

```
URL-Policy: https://policy.example.com/url-check
URL-Policy-Token: abc123
```

When a user clicks a URL:

```
https://example.com/reset?token=XYZ
```

The client MAY send:

```
POST /url-check HTTP/1.1
Host: policy.example.com

{
  "token": "abc123",
  "url": "https://example.com/reset?token=XYZ"
}
```

The policy endpoint evaluates the request and returns a decision.

9. Token and Trust Model

The URL-Policy-Token is an opaque identifier generated by the recipient domain.

The token does not contain user or URL information and MUST be treated as opaque by clients.

Enforcement of this mechanism is performed by recipient-side software that recognizes and enforces the defined header fields, such as webmail systems, mail clients, gateways, or associated enterprise security services.

The policy endpoint MUST be able to resolve or validate the token using implementation-specific mechanisms such as:

* Shared storage

* Pre-registration APIs

* Cryptographic validation

In deployments where a message is delivered to multiple recipients, recipient systems MAY generate distinct URL-Policy-Token values for each delivered message instance. This enables recipient-specific policy enforcement, tracking, and incident response.

Alternatively, a recipient domain MAY use a shared token representing a broader message context, provided that recipient identity is determined through local authenticated context at the time of URL activation.

Sender-provided tokens or endpoints MUST NOT be trusted without validation by the recipient domain.

Recipient systems MUST determine trusted policy endpoints based on local administrative policy. The presence of the URL-Policy header field alone MUST NOT be sufficient to establish trust.

10. Client Behavior

Clients that support this mechanism:

* SHOULD detect the URL-Policy header field

* MAY route URL activations through the policy endpoint

* MUST use secure transport (HTTPS)

Clients that do not support this mechanism:

* MUST ignore the header fields

* MUST process URLs normally

11. Security Considerations

This mechanism avoids modification of message content and therefore reduces interference with message authentication mechanisms such as DKIM.

Unlike URL rewriting approaches, this mechanism does not require modification of message body content and preserves compatibility with DKIM signatures.

This approach avoids issues related to URL normalization and encoding errors that may arise when processing internationalized URLs.

Implementations MUST:

* Ensure secure communication with policy endpoints

* Protect against unauthorized access and replay attacks

* Validate trust relationships for policy endpoints

12. Privacy Considerations

Implementations SHOULD:

* Minimize collection of user interaction data

* Avoid exposing sensitive information in URLs

* Allow recipient domains to retain control over telemetry

13. Backward Compatibility

This mechanism is optional.

Systems that do not recognize the defined header fields MUST ignore them and continue normal message processing.

14. IANA Considerations

This document requests registration of the following message header fields:

Header field name: URL-Policy
Applicable protocol: mail
Status: provisional
Author/Change controller: IETF
Specification document: this document

Header field name: URL-Policy-Token
Applicable protocol: mail
Status: provisional
Author/Change controller: IETF
Specification document: this document

Author’s Address

Phani Prasad Vinnakota Rajendra
Email: vrppinvictaralabs@gmail.com