sml
Internet-Draft
Intended status: Standards Track
Expires: 14 November 2025

H.-J. Happel
audriga
A. Gulbrandsen
ICANN

13 May 2025

Trust and security considerations for Structured Email
draft-happel-structured-email-trust-04

Abstract

This document discusses trust and security considerations for structured email and provides recommendations for message user agents on how to deal with structured data in email messages.

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 14 November 2025.

Table of Contents

1. Introduction

Structured email, as described in [I-D.sml-structured-email], makes the content of some email messages machine-readable, such that user agents can provide higher-level functions than displaying/replying, for example "add this to calendar".

Naturally, new functions bring new trust and security considerations, or bring new urgency to existing issues. This document discusses issues related to trust and security of structured email, and provides advice in some cases.

2. Requirements Language

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 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Types of security concerns

This section gives an overview of the various types of security and privacy concerns that arise when email messages contain structured data. The same concerns often arise for other messages, of course.

3.1. Spam/virus filters

Structured email increases the syntactical and semantic complexity of email messages. If a spam/virus filter parses structured email in order to block malevolent messages, the filter’s parser will necessarily differ from that of the MUA that will finally act on the structured data, creating a risk of misclassification.

These risks are elevated when a structured data format has complex syntax, syntax that offers several optional or alternative ways to express the same substance, and of course by parsers that deviate from the specification for better bug compatibiloty.

3.2. Formal display of data

A common example is displaying a received calendar invitation using dates/times in the recipient’s timezone, in a fixed format.

Formal display introduces additional possibilities of discrepancy between the different representations. For example, a single message might contains a multipart/alternative containing a text/plain description of a flight itinerary, a text/html description of the same itinerary, and a structured representation. All three may be different, leading to confusion (and in this example, perhaps to missing a flight).

Unintentional discrepancy is a risk for senders; some recipients may be misinformed.

If a message is sent to a group and there is a discrepancy, different members of the group may see it differently.

If a particular MUA displays the formal representation within the message, a malevolent sender could try to mimic the visual representation using HTML with CSS, but with misleading content.

3.3. Automated processing

Automated processing covers actions that are taken as soon as the message arrives rather than when a human user reads the message. For example, a user might want flight reservations to be automatically added to a travel itinerary application and/or a calendar.

Such automation could be a custom MUA feature or a future extension of the Sieve email filtering language ([RFC5228]). A related example for abuse in automated processing is calendar spam ([CalSpam]).

3.4. External references

Email messages with a text/html body part ("HTML email messages") may contain image resources that link to web servers. Such links can be used for tracking user interaction with the message.

Similar concerns apply to structured data types which include image references, such as the cover image of a music album or the teaser image of a news article.

RDF structured data can be partial by design and include references to additional data. Using a "follow your nose" approach, tools can follow URL references to obtain further structured data concerning a resource. For example, a piece of structured data about an article could reference the article’s authors only by URL. For a meaningful processing of author information, one might try to obtain further data using that URL.

3.5. Social engineering

While the risks of social engineering are hardly new and the human-readable text in a message can in principle be used to persuade the human reader to do anything, structured data widens the variety of actions the human reader can easily perform. If there are more buttons to click, then there’s also a greater variety of attacks.

Put differently: A user who might not be able to follow the instructions in a long and involved text-based social engineering attack may be able to follow simple instructions such as "click this ten that".

4. Trust mechanisms

Several implementations of structured email restrict processing to messages that are particularly trusted. That is to say, an incoming message is in one of these three categories:

1. Spam. Structured data is not processed.

2. Ordinary. Structured data is not processed.

3. Trusted. Structured data is processed.

This section gives an overview of the trust mechanisms used at the time of writing.

It does not attempt to describe whether a trust-based mechanism is appopriate in a particular case.

It is RECOMMENDED to restrict trust to a simple "or", rather than e.g. an elaborate scoring mechanism that’s difficult to understand for users.

4.1. Trusted senders

For the case of displaying remote images embedded in HTML email messages, MUAs often allow users to manually choose if they trust a certain sender. Sometimes, addresses in the MUA’s address book are considered trusted, or in a special list in the address book. While this is mentioned in [RFC6132] and [RFC6134], this mechanism is currently not standardized.

Several services manage trust centrally: trusted senders are trusted by the mail service rather than by the individual users.

Some services differentiate between local mail and mail that arrived across the internet, and trust local mail. In particular, one author has seen that attachments sent by remote senders are distrusted ("warning! do you really want to download this?") while attachments from senders within the recipient’s organization are assumed to be benevolent.

OPEN ISSUE: Whether this should discuss combining sender trust with sender/domain signatures. Pro: A sender that has used DKIM may be assumed trustworthy if subsequent message also do it, less so if not. Contra: Isn’t this what DMARC does, and best regulated only there?

4.2. Sender signatures

Sender signatures such as PGP or S/MIME could potentially be used as a trust indicator. The authors aren't aware of any concrete examples of this in the past, perhaps because of limited deployment.

4.3. Domain signatures

DKIM defines a signature on sender domain that may be used to verify that a message was sent by the same sender as an earlier message.

Some mail hosts restrict structured processing to messages with DKIM signatures, or to a set of senders who are identified by their DKIM signatures.

4.4. Recipient-generated identifiers

Part of the simplicity of email is the fact that just the email address is required to reach out to a recipient. This however requires the the recipient to discern whether a message is desirable or abusive.

Recipient-generated transaction identifiers aim to pass a certain secret to the sender, which helps to prove legitimacy. One such approach are one-time email aliases, which are generated for a single transaction or series of transactions.

4.5. Secrets and related messages

A message may contain a secret or hard-to-guess identifier.

A message may leverage another message's trust level by referencing an identifier from the other message.

For example, a ticket sold by a concert venue may reference the payment transaction ID that's also mentioned in a message from the payment processor. If these IDs are difficult to guess and the receiver trusts the payment processor, then this gives some reason to trust that the ticket is bona fide and could safely be added to the user's ticket/wallet app.

In another example, a message with a calendar update may reference an identifier from the original calendar entry. If this ID is hard to guess and the recipient acted on the message containing the original entry, then this gives some reason to trust that the change is bona fide.

OPEN ISSUE: How to specify the secret.

One possible solution: The leveraging message contains a specification of sender domain and optinal xpath query, and the MUA decides whether the sender is on its list. But I don’t like having xpath, its interop is said to be poor and a malevolent sender could specify something easily guessed. For example, the messages I receive from a popular payment processor contain my name; an attacker could use an xpath query that results in my name.

Another possible solution: The leveraging message specifies the sender of the leveraged message and the secret, and each MUA contains a hardcoded xpath query to find the ONLY kind of secret it trusts for that particular sender.

5. Categories of use cases

Certain types of structured data might need to be kept more secure than others. For instance, the structured data representation of a music album shared by a friend would not contain major personal information, while e.g., medical records or financial statements certainly would.

6. Implementation guidelines

6.1. Processing structured data

MUAs SHOULD consider structured data in incoming email messages only if either of these criteria hold:

* The sender is trusted (e.g., part of the user’s address book) and the messsage either contains a valid personal or domain signature

* The message is part of an ongoing thread with a trusted sender

If none of these criteria is fulfilled, MUAs should fall back to alternative presentations (e.g., "text/html" or "text/plain").

Open issue 2: Really SHOULD? I (Arnt) don’t see this guidance as clearly right. It doesn’t match existing code well, and doesn’t seem to be where we want to go.

6.2. Inlining data

Structured data included in an email message should be self-contained in order to avoid privacy problems. This implies that if an MUA is able to provide meaningful user interaction (rather than mere display), then it should do that without loading additional referenced resources from the web.

7. Implementation status

RFC Editor: before publication please remove this section and the reference to [RFC7942].

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

7.1. Structured Email plugin for Roundcube Webmail

The plugin currently uses the "trusted sender" feature of Roundcube to determine if structured data should be processed.

8. Security considerations

Security considerations are a core subject of this document.

9. Privacy considerations

Privacy considerations are a core subject of this document.

10. IANA Considerations

This document has no IANA actions at this time.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

11.2. Informative References

[I-D.sml-structured-email] "*** BROKEN REFERENCE ***".

[RFC5228] Guenther, P., Ed. and T. Showalter, Ed., "Sieve: An Email Filtering Language", RFC 5228, DOI 10.17487/RFC5228, January 2008, <https://www.rfc-editor.org/rfc/rfc5228>.

[RFC6132] George, R. and B. Leiba, "Sieve Notification Using Presence Information", RFC 6132, DOI 10.17487/RFC6132, July 2011, <https://www.rfc-editor.org/rfc/rfc6132>.

[RFC6134] Melnikov, A. and B. Leiba, "Sieve Extension: Externally Stored Lists", RFC 6134, DOI 10.17487/RFC6134, July 2011, <https://www.rfc-editor.org/rfc/rfc6134>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <https://www.rfc-editor.org/rfc/rfc7942>.

[CalSpam] "Calendar operator practices—Guidelines to protect against calendar abuse", n.d., <https://standards.calconnect.org/csd/cc-18003.html>.

[MachineReadable] "NIST IR 7511 Rev 4", n.d., <https://csrc.nist.gov/glossary/term/Machine_Readable>.

Appendix A. Acknowledgements

The authors wish to thank Ben Bucksch, Alexey Melnikov, Phillip Tao and others whose suggestions were made before this paragraph was started.

Authors' Addresses

Hans-Joerg Happel
audriga
Email: happel@audriga.com
URI: https://www.audriga.com

Arnt Gulbrandsen
ICANN
6 Rond Point Schumann, Bd. 1
1040 Brussels
Belgium
Email: arnt@gulbrandsen.priv.no
URI: https://icann.org/ua