Workgroup: Network Working Group
Internet-Draft:
draft-nottingham-feed-privacy-00
Published: 21 June 2022
Intended Status: Best Current Practice
Expires: 23 December 2022
Authors: M. Nottingham

Privacy Considerations for Web Feed Readers

Abstract

This specification collects privacy-enhancing guidelines for Web feed readers.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-nottingham-feed-privacy/.

information can be found at https://mnot.github.io/I-D/.

Source for this draft and an issue tracker can be found at https://github.com/mnot/I-D/labels/feed-privacy.

Note to Readers

This draft is a quick straw-man; it is intended to assess implementer and community interest in the topic, not to state concrete requirements (yet). Feedback much appreciated.

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 23 December 2022.

Copyright Notice

Table of Contents

1. Introduction

Many web sites offer a feed of updates to their content, using [ATOM] or [RSS]. While they are consumed in a variety of ways and for a variety of purposes, web feeds are often presented to users by dedicated software, colloquially known as a "feed reader."

Feed readers use HTML and HTTP, and can be considered as part of the web, but one that is distinct from web browsers. Unlike browsers, feed readers do not easily facilitiate cross-site tracking or behavioural advertising, because their capabilities are more

limited, thereby establishing an alternative, more privacy-respecting web platform.

At the same time, browsers are protecting privacy in increasingly sophisticated ways; for example, by taking steps to prevent active fingerprinting [FINGERPRINTING].

This specification seeks to codify these privacy-enhancing distinctions while incorporating browser's privacy advances by offering a definition for "feed reader" in Section 2, providing guidelines for how they make requests in Section 3, and providing guidelines for their handling of content in Section 4.

1.1. Notational Conventions

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.

2. Feed Readers

A feed reader acts as a user agent (per [HTTP]) that consumes and presents information from documents in [ATOM], [RSS], and/or similar formats to users.

A feed reader might be local software program on a host that the user controls, or a remote service that they access over the Internet, such as through a web browser. Typically, a feed reader will allow the user to subscribe to URIs that identify feeds, and regularly poll those URIs for new content. When a feed entry has already been seen, a reader might keep this state.

Feed readers make HTTP requests and parse, render and display HTML content (including some embedded content). Users can also follow links from content in a feed reader.

3. Making Feed Requests

When a feed reader makes a request for a feed document, privacy can be impacted in several ways. This section contains guidelines for such requests; note that they do not apply to requests for embedded content and user-initiated navigation to links in content (see Section 4).

3.1. Encryption

In HTTP, encryption protects communication from observation and modification, and is used to establish the identity of the server.

Feed readers, therefore, are expected to follow best current practice for encryption, as captured in the relevant RFCs and industry practice.

This includes implementation of the most recent version of TLS (as of this writing, [TLS13]), the Strict-Transport-Security mechanism [HSTS], and Certificate Transparency checking [TRANS].

3.2. Cookies

The HTTP Cookie mechanism has aspects that are problematic for privacy; see, eg., Part xx of [COOKIES]. Therefore, when making feed requests feed readers **MUST NOT** send the Cookie header field, and when receiving feed responses, they **MUST NOT** process the Set-Cookie header field.

3.3. ETags

HTTP ETags (see Part x.x of [HTTP]) are especially useful to feed readers, as they enable more efficient transfers when there have been no changes to a feed. However, they can also be used to track user activity. Therefore, feed readers **SHOULD** periodically send requests without If-None-Match header fields, to asure that ETags are changed.

3.4. User-Agent

Feed readers **SHOULD NOT** include more significant detail than an identifier for the software being used and its version. In particular, detail about libraries used and other aspects of the environment can contribute to the formation of an identifier for the user.

3.5. Client IP Address

Feed readers **SHOULD** take steps to prevent servers hosting feeds from using the client's IP address to identify them or track their activity. For example, [MASQUE] might be used to this end.

4. Handling Feed Content

When a feed reader displays a feed content (including an individual feed entry) to its user, interaction with the feed's server is limited in several ways to reduce privacy impact. This section outlines those limits.

4.1. Requesting Remote Resources

Feed readers **MAY** make requests for remote resources that are explicitly part of the feed or feed entry's metadata. For example, a

feed reader might fetch the URL in the atom:logo element (defined in Section 4.2.7 of [ATOM]) in order to present it to the user.

Feed readers **MAY** make requests for remote resources that are embedded in feed content. However, the user **MUST** be able to control this behaviour.

4.2. Executing Scripts

When handling feed content, feed readers **MUST NOT** execute embedded or linked scripts.

4.3. Reporting

Feed readers **MUST NOT** trigger reporting mechanisms designed for Web browsers when handing feed content. For example, [NEL], [CSP].

4.4. Following Links

When a user explicitly follows a link in a feed reader, their expectation will be that it either opens in their preferred Web browser, or that the resulting functionality is equivalent (e.g., a browser embedded in the feed reader). Once a link is followed, the feed reader is no longer handling feed content; the user's activity is now either in a separate Web browser, or in an embedded web browser that is considered a distinct context.

Therefore, the context used to follow a link **MUST** be separate from that used to make requests for feed documents. In particular, separate underlying connections are to be used, and no state such as cookies is to be shared.

5. IANA Considerations

This document has no actions for IANA.

6. Security Considerations

TBD

7. References

7.1. Normative References

[HSTS] Hodges, J., Jackson, C., and A. Barth, "HTTP Strict Transport Security (HSTS)", RFC 6797, DOI 10.17487/

RFC6797, November 2012, <https://www.rfc-editor.org/rfc/rfc6797>.

[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>.

[TLS13] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/rfc/rfc8446>.

[TRANS] Laurie, B., Messeri, E., and R. Stradling, "Certificate Transparency Version 2.0", RFC 9162, DOI 10.17487/RFC9162, December 2021, <https://www.rfc-editor.org/rfc/rfc9162>.

7.2. Informative References

[ATOM] Nottingham, M., Ed. and R. Sayre, Ed., "The Atom Syndication Format", RFC 4287, DOI 10.17487/RFC4287, December 2005, <https://www.rfc-editor.org/rfc/rfc4287>.

[COOKIES] Chen, L., Englehardt, S., West, M., and J. Wilander, "Cookies: HTTP State Management Mechanism", Work in Progress, Internet-Draft, draft-ietf-httpbis-rfc6265bis-10, 24 April 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-httpbis-rfc6265bis-10>.

[CSP] West, M., Barth, A., and D. Veditz, "Content Security Policy Level 2", World Wide Web Consortium Recommendation REC-CSP2-20161215, 15 December 2016, <https://www.w3.org/TR/2016/REC-CSP2-20161215>.

[FINGERPRINTING] Doty, N., "Mitigating Browser Fingerprinting in Web Specifications", World Wide Web Consortium NOTE NOTE-fingerprinting-guidance-20190328, 28 March 2019, <https://www.w3.org/TR/2019/NOTE-fingerprinting-guidance-20190328>.

[HTTP] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/rfc/rfc9110>.

[MASQUE] Schinazi, D., "Proxying UDP in HTTP", Work in Progress, Internet-Draft, draft-ietf-masque-connect-udp-15, 17 June 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-masque-connect-udp-15>.

[NEL] Creager, D., Grigorik, I., Tuttle, J., Reitbauer, A., Jain, A., and J. Mann, "Network Error Logging", World Wide Web Consortium WD WD-network-error-logging-1-20180925, 25 September 2018, <https://www.w3.org/TR/2018/WD-network-error-logging-1-20180925>.

[RSS] RSS Advisory Board, "RSS 2.0 Specification", March 2009, <https://www.rssboard.org/rss-specification>.

Author's Address

Mark Nottingham
Prahran
Australia

Email: mnot@mnot.net
URI: https://www.mnot.net/