HTTP Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 27, 2019

S. Ludin
Akamai Technologies
M. Nottingham
Fastly
N. Sullivan
Cloudflare

October 24, 2018

CDN Loop Prevention
draft-ietf-httpbis-cdn-loop-01

Abstract

This specification defines the CDN-Loop request header field for HTTP.

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 April 27, 2019.

Copyright Notice

Table of Contents

1. Introduction

In modern deployments of HTTP servers, it is common to interpose Content Delivery Networks (CDNs) in front of origin servers to improve end-user perceived latency, reduce operational costs, and improve scalability and reliability of services.

Often, more than one CDN is in use by a given origin. This happens for a variety of reasons, such as cost savings, arranging for failover should one CDN have issues, or to directly compare their services.

As a result, it is not unknown for forwarding CDNs to be configured in a "loop" accidentally; because routing is achieved through a combination of DNS and forwarding rules, and site configurations are sometimes complex and managed by several parties.

When this happens, it is difficult to debug. Additionally, it sometimes isn't accidental; loops between multiple CDNs be used as an attack vector (e.g., see [loop-attack]), especially if one CDN unintentionally strips the loop detection headers of another.

This specification defines the CDN-Loop request header field for HTTP to enable secure interoperability of forwarding CDNs. Having a header that is guaranteed not to be modified by other CDNs that are used by a shared customer helps give each CDN additional confidence that any purpose (debugging, data gathering, enforcement) that they use this header for is free from tampering due to how that customer configured the other CDNs.

1.1. Relationship to Via

HTTP defines the Via header field in [RFC7230], Section 5.7.1 for "tracking message forwards, avoiding request loops, and identifying

the protocol capabilities of senders along the request/response chain."

In theory, Via could be used to identify these loops. However, in practice it is not used in this fashion, because some HTTP servers use Via for other purposes - in particular, some implementations disable some HTTP/1.1 features when the Via header is present.

1.2. Conventions and Definitions

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.

This specification uses the Augmented Backus-Naur Form (ABNF) notation of [RFC5234] with a list extension, defined in Section 7 of [RFC7230], that allows for compact definition of comma-separated lists using a '#' operator (similar to how the '*' operator indicates repetition). Additionally, it uses the OWS rule from [RFC7230] and the parameter rule from [RFC7231].

2. The CDN-Loop Request Header Field

The CDN-Loop request header field is intended to help a Content Delivery Network identify when an incoming request has already passed through that CDN's servers, to prevent loops.

```
CDN-Loop = #cdn-id
cdn-id   = token *( OWS ";" OWS parameter )
```

Conforming Content Delivery Networks SHOULD add a value to this header field to all requests they generate or forward (creating the header if necessary).

The token identifies the CDN as a whole. Chosen token values SHOULD be unique enough that a collision with other CDNs is unlikely. Optionally, the token can have semicolon-separated key/value parameters, to accommodate additional information for the CDN's use.

As with all HTTP headers defined using the "#" rule, the CDN-Loop header can be added to by comma-separating values, or by creating a new header field with the desired value.

For example:

```
CDN-Loop: FooCDN, barcdn; host="foo123.bar.cdn"
CDN-Loop: baz-cdn; abc="123"; def="456", anotherCDN
```

Note that the token syntax does not allow whitespace, DQUOTE or any of the characters "(),/:;<=>?@[]{}". See [RFC7230], Section 3.2.6. Likewise, note the rules for when parameter values need to be quoted in [RFC7231], Section 3.1.1.

To be effective, intermediaries - including Content Delivery Networks - MUST NOT remove this header field, or allow it to be removed (e.g., through configuration) and servers (including intermediaries) SHOULD NOT use it for other purposes.

3. Security Considerations

The threat model that the CDN-Loop header field addresses is a customer who is attempting to attack a service provider by configuring a forwarding loop by accident or malice. For it to function, CDNs cannot allow it to be modified by customers (see Section 2).

The CDN-Loop header field can be generated by any client, and therefore its contents cannot be trusted. CDNs who modify their behaviour based upon its contents should assure that this does not become an attack vector (e.g., for Denial-of-Service).

It is possible to sign the contents of the header (either by putting the signature directly into the field's content, or using another header field), but such use is not defined (or required) by this specification.

4. IANA Considerations

This document registers the "CDN-Loop" header field in the Permanent Message Header Field Names registry.

- Header Field Name: CDN-Loop
- Protocol: http
- Status: standard
- Reference: (this document)

5. References

5.1. 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/info/rfc2119>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <https://www.rfc-editor.org/info/rfc7230>.

[RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <https://www.rfc-editor.org/info/rfc7231>.

[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/info/rfc8174>.

5.2. Informative References

[loop-attack] Chen, J., Jiang, J., Zheng, X., Duan, H., Liang, J., Li, K., Wan, T., and V. Paxson, "Forwarding-Loop Attacks in Content Delivery Networks", ISBN 1-891562-41-X, DOI 10.14722/ndss.2016.23442, February 2016, <http://www.icir.org/vern/papers/cdn-loops.NDSS16.pdf>.

Authors' Addresses

Stephen Ludin
Akamai Technologies

Email: sludin@akamai.com

Mark Nottingham
Fastly

Email: mnot@fastly.com

Nick Sullivan
Cloudflare

Email: nick@cloudflare.com