HTTP Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 8, 2019

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

February 04, 2019

CDN Loop Detection
draft-ietf-httpbis-cdn-loop-02

Abstract

This document defines the CDN-Loop request header field for HTTP. CDN-Loop addresses an operational need that occurs when an HTTP request is intentionally forwarded between Content Delivery Networks (CDNs), but is then accidentally or maliciously re-routed back into the original CDN causing a non-terminating loop. The new header field can be used to identify the error and terminate the loop.

Status of This Memo

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 can 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 HTTP request header field to help detect such attacks and accidents among implementing forwarding CDNs, by disallowing its modification by their customers.

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 token, OWS, uri-host and port rules 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 detect loops.

```
CDN-Loop  = #cdn-info
cdn-info  = cdn-id *( OWS ";" OWS parameter )
cdn-id    = ( uri-host [ ":" port ] ) / pseudonym
pseudonym = token
```

The cdn-id identifies the CDN using either a hostname under its control or a pseudonym. Hostnames are preferred, to help avoid accidental collisions. If a pseudonym is used, unintentional collisions are more likely, and therefore values should be carefully chosen to prevent them; for example, using a well-known value (such as the recognized name of the CDN in question), or a generated value with enough entropy to make collisions unlikely (such as a UUID [RFC4122]).

Optionally, cdn-info can have semicolon-separated key/value parameters, to accommodate additional information for the CDN's use.

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

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

For example:

```
GET /image.jpg HTTP/1.1
Host: cdn-customer.example
User-Agent: ExampleBrowser/5
CDN-Loop: foo123.foocdn.example, barcdn.example; trace="abcdef"
CDN-Loop: AnotherCDN; abc=123; def="456"
```

Note that the pseudonym 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.

The effectiveness of this mechanism relies on all intermediaries preserving the header field, since removing (or allowing it to be removed, e.g., by customer configuration) would prevent downstream CDNs from using it to detect looping. In general, unknown header fields are not removed by intermediaries, but there may be need to add CDN-Loop to an implementation's list of header fields that are not to be removed under any circumstances. The header field SHOULD NOT be used for other purposes.

3. Security Considerations

The threat model that the CDN-Loop header field addresses is a customer who is attacking a service provider by configuring a forwarding loop by accident or malice. For it to function, CDNs cannot allow customers to modify or remove it in their configuration (see Section 2).

Note that a CDN that allows customers to remove or modify the CDN-Loop header field (i.e., they do not implement this specification) remains an attack vector against both implementing and non-implementing CDNs.

A CDN's use of the CDN-Loop header field might expose its presence. For example, if CDN A is configured to forward its requests to CDN B for a given origin, CDN B's presence can be revealed if it behaves differently based upon the presence of the CDN-Loop header field.

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 field (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.

Depending on how it is used, CDN-Loop can expose information about the internal configuration of the CDN; for example, the number of hops inside the CDN, and the hostnames of nodes.

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", DOI 10.14722/ndss.2016.23442, February 2016, <http://www.icir.org/vern/papers/cdn-loops.NDSS16.pdf>.

[RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, DOI 10.17487/RFC4122, July 2005, <https://www.rfc-editor.org/info/rfc4122>.

Authors' Addresses

Stephen Ludin
Akamai Technologies

Email: sludin@akamai.com

Mark Nottingham
Fastly

Email: mnot@fastly.com

Nick Sullivan
Cloudflare

Email: nick@cloudflare.com