Network Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: April 11, 2021

M. Nottingham
October 8, 2020

Greasing HTTP
draft-nottingham-http-grease-01

Abstract

Like many network protocols, HTTP is vulnerable to ossification of its extensibility points. This draft explains why HTTP ossification is a problem and establishes guidelines for exercising those extensions by 'greasing' the protocol to combat it.

Note to Readers

RFC EDITOR: please remove this section before publication

The issues list for this draft can be found at https://github.com/mnot/I-D/labels/http-grease [1].

The most recent (often, unpublished) draft is at https://mnot.github.io/I-D/http-grease/ [2].

Recent changes are listed at https://github.com/mnot/I-D/commits/gh-pages/http-grease [3].

See also the draft's current status in the IETF datatracker, at https://datatracker.ietf.org/doc/draft-nottingham-http-grease/ [4].

Status of This Memo

Table of Contents

1. Introduction

Like many network protocols, HTTP is vulnerable to ossification of its extensibility points. Ossification happens when a significant number of the systems that generate, transmit, handle, or consume the protocol don't accept a new extension, thereby making it more difficult to deploy extensions.

For example, TCP has effectively been ossified by middleboxes that assume that new TCP options will not be deployed; likewise, the Protocol field in IP has been effectively ossified as well, since so many networks will only accept TCP or UDP traffic.

Addressing this issue is important; protocol extensibility allows adaptation to new circumstances as well as application to new use cases. Inability to deploy new extensions creates pressure to misuse the protocol - often leading to undesirable side effects - or to use other protocols, reducing the value that the community gets from a shared, standard protocol.

While there are a few ways that protocol designers can mitigate ossification, this document focuses on a technique that's well suited to many of the ossification risks in HTTP: 'greasing' extensibility points by exercising them, so that they don't become 'rusted shut.'

[RFC8701]) pioneered greasing techniques in IETF protocols; this document explains how they apply to HTTP. It focuses on generic HTTP features; other documents cover versioned extensibility points (e.g., see [I-D.bishop-httpbis-grease]).

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. Ossification and HTTP

As an application protocol, HTTP has several extensibility points. For example, methods, status codes, header and trailer fields, cache directives, range units and content codings are all HTTP extension points.

Each extension point defines how unrecognised values should be handled; in most cases, they should be ignored (e.g. header fields, cache directives and range units), while in a few cases they have other handling (e.g., unrecognised methods result in a 405 status code; unrecognised status codes devolve to a more generic x00 status code).

Implementations and other components that diverge from these defined behaviours risk ossifying that extensibility point.

For example, it is increasingly common for Web Application Firewalls (WAFs), bot detection services and similar components to reject HTTP requests that contain header fields with certain characters or strings, even though syntactically valid, and even though the header fields are not necessarily recognised by the recipient.

This behaviour has become prevalent enough to make it difficult for Web browsers and other clients to introduce new request header fields. That difficulty is aggravated by two factors:

1. A relatively large number of vendors create these components, but have little coordination between them, leading to wide variances in behaviour, and

2. Many of these components' deployments are not updated regularly and reliably, leading to difficulty in addressing ossification issues even when they are identified.

To avoid ossification of request header fields, it is Best Current Practice to grease them, as explained below. Other HTTP extensibility points might be added in the future, and it is not to be inferred that greasing other HTTP extensibility points is not good practice.

2.1. Greasing HTTP Request Header Fields

HTTP clients SHOULD grease request header fields. There are two aims in doing so:

1. Preserving the ability to add new request header fields over time

2. Preserving the ability to add new request header fields with values containing common syntax

Clients can grease a given request at their discretion. For example, a client implementation might add one or more grease request header fields to every request it makes, or it might add one to every third or tenth request.

Depending on the deployment model of the client, it might do this in production releases automatically (especially if there are ways that it can modify how grease values are sent with a high degree of control, in case too many errors are encountered), or it might do so only in pre-releases.

Grease field names SHOULD be hard to predict; e.g., they SHOULD NOT have any identifying prefix, suffix, or pattern. However, they MUST NOT be likely to conflict with unregistered or future field names, and the grease coordinator MUST avoid potentially offensive or confusing terms. They also MUST conform to the syntactic requirements for field names in HTTP ([I-D.ietf-httpbis-semantics], Section 4.3).

This can be achieved in different ways (which SHOULD vary from time to time), for example:

- Combine two or three dictionary words or proper nouns with a hyphen (e.g., 'Skateboard-Clancy', 'Murray-Fortnight-Scout')

- Append digits to a dictionary word (e.g., 'Turnstile23')

- Generate a string using a hash or similar function (e.g., 'dd722785c01b')

Grease field names are not required to be registered in the IANA HTTP Field Name Registry, unless they are intended to be used over an extended period of time (e.g., more than one year). However, they MAY be registered as Provisional with a reference to this RFC or another explanatory resource, to help interested parties to find out what they are used for. Such registered values SHOULD be removed after the client stops using that field.

Greasing clients SHOULD not reuse other clients' grease fields names, unless they coordinate.

Grease field values can be fixed strings, or dynamically generated at runtime. It is RECOMMENDED that greasing clients exercise the various types in [I-D.ietf-httpbis-header-structure].

If an error is encountered by a greasing client, it SHOULD NOT re-issue the request without the grease value, since hiding the consequences of the failure doesn't serve the purpose of greasing.

Greasing clients SHOULD announce new field names they intend to grease on the http-grease@ietf.org mailing list.

3. Security Considerations

Some HTTP extensibility points are becoming (or have become) ossified because of security considerations; receiving implementations believe that it is more secure to reject unknown values, or that they can identify undesirable peers through their use of extensions.

This document does not directly address these concerns, nor does it directly disallow such behaviour. Instead, it aims to encourage the ability to accommodate new extensions more quickly than is now possible.

4. References

4.1. Normative References

[I-D.ietf-httpbis-semantics]
Fielding, R., Nottingham, M., and J. Reschke, "HTTP Semantics", draft-ietf-httpbis-semantics-12 (work in progress), October 2020.

[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>.

[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>.

4.2. Informative References

[I-D.bishop-httpbis-grease] Bishop, M., "GREASE for HTTP/2", draft-bishop-httpbis-grease-01 (work in progress), June 2020.

[I-D.ietf-httpbis-header-structure] Nottingham, M. and P. Kamp, "Structured Field Values for HTTP", draft-ietf-httpbis-header-structure-19 (work in progress), June 2020.

[RFC8701] Benjamin, D., "Applying Generate Random Extensions And Sustain Extensibility (GREASE) to TLS Extensibility", RFC 8701, DOI 10.17487/RFC8701, January 2020, <https://www.rfc-editor.org/info/rfc8701>.

4.3. URIs

[1] https://github.com/mnot/I-D/labels/http-grease

[2] https://mnot.github.io/I-D/http-grease/

[3] https://github.com/mnot/I-D/commits/gh-pages/http-grease

[4] https://datatracker.ietf.org/doc/draft-nottingham-http-grease/

Author's Address

Mark Nottingham
made in
Prahran, VIC
Australia

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