Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 20, 2012

A. Petersson
M. Nilsson
Opera Software

June 18, 2012

Forwarded HTTP Extension
draft-ietf-appsawg-http-forwarded-04

Abstract

This document standardizes an HTTP extension header field that allows proxy components to disclose information lost in the proxying process, e.g., the originating IP address of a request or IP number of the proxy on the user-agent-facing interface. Given a trusted path of proxying components, this makes it possible to arrange so that each subsequent component will have access to e.g., all IP addresses used in the chain of proxied HTTP requests.

This document also specifies guidelines for a proxy administrator to anonymize the origin of a request.

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 http://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 December 20, 2012.

Table of Contents

1. Introduction

In today's HTTP landscape, there are a multitude of different applications acting as a proxy for the user agent and effectively anonymizing the requests to look as if they originated from the proxy IP address or in other ways changing the information in the original request. Examples of such applications include caching, content filtering, content compression, crypto offload, and load balancing. As most of the time this loss of information is not the primary purpose, or even a desired effect, a way of disclosing the original information at HTTP level instead of depending on the TCP/IP connection remote IP address and transport port number is needed.

In addition to the above mentioned problems, there may also be issues due to the use of NAT. This is further discussed in [RFC6269].

A common way to disclose this information is by using the de facto standard header fields such as X-Forwarded-For, X-Forwarded-By, and X-Forwarded-Proto. This document intends to standardize syntax and semantics for disclosing such information. The header field also combines all information within one single header field, making it possible to correlate the header fields to each other. With the header field format described in this document, it is possible to know what information belongs together, given that the proxies are trusted. Such conclusions are not possible to make with the X-Forwarded class of header fields. This new header field also extends the de facto standard of, e.g., X-Forwarded-For with features for which real life deployments have shown a need.

2. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Syntax Notations

This specification uses the Augmented Backus-Naur Form (ABNF) notation of [RFC5234] with the list rule extension defined in Section 3.2.5 of [I-D.ietf-httpbis-p1-messaging].

4. Forwarded HTTP header field

The Forwarded HTTP header field is an OPTIONAL header field that, when used, contains a list of parameter-identifier pairs that

disclose information that is altered or lost when a proxy is involved in the path of the request. This applies to forwarding proxies, as well as reverse proxies. Information passed in this header can be, e.g., the source IP address of the request, the IP address of the incoming interface on the proxy, or whether HTTP or HTTPS was used. If the request is passing through several proxies, each proxy MAY add a set of parameters; it MAY also remove earlier added Forwarded-header fields.

The top-level list is represented as a list of HTTP header field-value's as defined in Section 3.2 of [I-D.ietf-httpbis-p1-messaging]. The first element in this list holds information added by the first proxy, followed by information added by any subsequent proxy. Each field-value is a semicolon-separated list; this sub-list consists of parameter-identifier pairs. Parameter-identifier pairs are grouped together by an equals sign. The parameter names are case-insensitive. The header field can be defined in augmented BNF syntax as:

```
    Forwarded   = "Forwarded" ":" LWS Forwarded-v
    Forwarded-v = 1#forwarded-element

    forwarded-element =
            [ forwarded-pair ] *( ";" [ forwarded-pair ] )

    forwarded-pair = token "=" value
    value          = token / quoted-string
```

Examples:

```
    Forwarded: For="[2001:db8:cafe::17]:4711"
    Forwarded: for=192.0.2.60;proto=http;by=203.0.113.43
    Forwarded: for=192.0.2.43, for=198.51.100.17
```

Note that as ":" and "[]" are not valid characters in token, IPv6 addresses are written as quoted-string.

Given that a proxy wishes to add a Forwarded header field to the outgoing request, if the incoming request has no such header field, the proxy simply adds the header field with the list of parameters desired. If, on the other hand, the incoming request has such a header field, the proxy can either add a comma and the list of parameters, or add a new instance of the header field. A proxy MAY remove all Forwarded header fields from a request. It MUST, however, ensure that the correct header field is updated in case of multiple Forwarded header fields.

5. Parameters

This document specifies a number of parameters and valid values for each of them. Each parameter MUST NOT occur more than once per "forwarded-element" as defined in ABNF in Section 4.

- "by" identifies the user-agent facing interface of the proxy.
- "for" identifies the node making the request to the proxy.
- "host" is the host request header-field as received by the proxy.
- "proto" indicates what protocol was used to make the request.

5.1. Forwarded by

The "by" parameter is used to disclose the interface where the request came in to the proxy server. Typically, the value of this parameter is an IP address and optionally a port number; however, it can also be some other kind of identifier.

The syntax of a "by" value, after potential quoted-string unescaping, MUST conform to the "node" ABNF described in Section 6.

This is primarily added by reverse proxies that wish to forward this information to the backend server. It can also be interesting in a multi-homed environment to signal to backend servers where the request came from.

5.2. Forwarded for

The "for" parameter is used to disclose information about the client that initiated the request and following proxies in a chain of proxies. Typically, the value of this parameter is an IP address, but it can also be some other kind of identifier.

The syntax of a "for" value, after potential quoted-string unescaping, MUST conform to the "node" ABNF described in Section 6.

In a chain of proxy servers where this is fully utilized, the first for-parameter will disclose the user agent where the request was first made, followed by any subsequent proxy identifiers. The last proxy in the chain is not part of the list of for-parameters. The last proxy's IP address, and optionally a port number, are, however, readily available as the remote IP address of the TCP/IP connection. It can, however, be more relevant to read information about the last proxy from preceding Forwarded header field's by-parameter, if present.

5.3. Forwarded host

The "host" parameter is used to forward the original value of the "Host" header field. This can be used, for example, by the origin server if a reverse proxy is rewriting the "Host" header field to some internal host name.

The syntax for a "host" value, after potential quoted-string unescaping, MUST conform to the Host ABNF described in Section 5.4 of [I-D.ietf-httpbis-p1-messaging].

5.4. Forwarded proto

The "proto" parameter has the value of the used protocol type. The syntax of a "proto" value, after potential quoted-string unescaping, MUST conform to the URI scheme name as defined in Section 3.1 in [RFC3986] and registered to IANA according to [RFC4395]. Typical values are "http" or "https".

For example, in an environment where a reverse proxy is also used as a crypto offloader, this allows the origin server to rewrite URLs in a document to match the type of connection as the user agent requested, even though all connections to the origin server are unencrypted HTTP.

5.5. Extensions

Private extensions allow for adding own parameters and values. This can be particularly useful in a reverse proxy environment. If these extensions are to be widely spread, it is RECOMMENDED that they are standardized. It is possible to register additional parameters to the register HTTP Forwarded Parameters. This is further discussed in Section 9.

6. Node identifiers

The node identifiers are the IP address, and, optionally port number, of the network node, a predefined token hiding the real identity, but signaling that such a component exists in the network path, or a generated token allowing for tracing and debugging without revealing network internals.

```
nodename = IPv4address / "[" IPv6address "]" /
           "unknown" / obfnode
```

All of the identifiers may optionally have the port identifier, for example, allowing the identification of the end point in a NATted

environment.

```
    node            = nodename [ ":" node-port ]
```

The node-port can be identified either by its TCP port number or by a generated token obfuscating the real port number.

```
    node-port       = port / obfport
    port            = 1*5DIGIT
    obfport         = "_" 1*(ALPHA / DIGIT / "." / "_" / "-")
```

Note that this also allows port numbers to be appended to the the "unknown" identifier. Interpretation of such notation is, however, left to the possessor of a proxy adding such a value to the header field. To distinguish an obfport from a port, obfport MUST have a leading underscore. Further, it MUST also consist of only US-ASCII letters, US-ASCII digits and the characters ".", "_" and "-".

It is important to note that an IPv6 address and any nodename with node-port specified MUST be quoted, since ":" is not an allowed character in token.

Examples:

```
          "192.0.2.43:47011"
          "[2001:db8:cafe::17]:47011"
```

6.1. IPv4 and IPv6 identifiers

The ABNF rules for "IPv6address" and "IPv4address" are defined in [RFC3986] The IPv6address SHOULD comply with textual representation recommendations [RFC5952] (e.g., lowercase, compression of zeroes).

Note that the IP address may be one from the internal nets, as defined in [RFC1918] and [RFC4193]. Also, note that an IPv6 address is always enclosed in square brackets.

6.2. The "unknown" identifier

The "unknown" identifier is used when the identity of the preceding entity is not known. One example would be a proxy server process generating an outgoing request without direct access to the incoming request TCP socket.

6.3. Obfuscated identifier

A generated identifier may be used where there is a wish to keep the internal IP addresses secret, while still allowing the Forwarded

header field to be used for tracing. This can also be useful if the proxy uses some sort of interface labels and it is desired to pass them rather than an IP address. The identifiers can be randomly generated for each request and do not need to be statically assigned to resources. To distinguish the obfuscated identifier from other identifiers, it MUST have a leading underscore "_". Further, it MUST also consist of only US-ASCII letters, US-ASCII digits and the characters ".", "_" and "-".

```
obfnode        = "_" 1*( ALPHA / DIGIT / "." / "_" / "-")
```

Example:

```
Forwarded: for=_hidden, for=_SEVKISEK
```

7. Implementation considerations

7.1. HTTP lists

Note that an HTTP list allows white spaces to occur between the identifiers, and the list may be split over multiple header fields. As an example, the header field

```
Forwarded: for=192.0.2.43,for="[2001:db8:cafe::17]",for=unknown
```

is equivalent to the header field

```
Forwarded: for=192.0.2.43, for="[2001:db8:cafe::17]", for=unknown
```

which is equivalent to the header fields

```
Forwarded: for=192.0.2.43
Forwarded: for="[2001:db8:cafe::17]", for=unknown
```

7.2. Header field preservation

There are some cases when this header field should be kept and some cases where it should not be kept. A directly forwarded request should preserve and possibly extend it. If a single incoming request causes the proxy to make multiple outbound requests, special care must be taken to decide whether the header field should be preserved or not. In many cases the header field should be preserved, but if the outbound request is not a direct consequence of the incoming request, the header field should not be preserved. Consider also the case when a proxy has detected a content mismatch in a 304 response and is following the instructions in [I-D.ietf-httpbis-p4-conditional] Section 4.1 to repeat the request

unconditionally, in which case the new request is still basically a direct consequence of the origin request, and the header should probably be kept.

7.3. Relation to Via

Note that it is not possible to combine information from this header field with the information from the Via header field. Some proxies will not update the Forwarded header field, some proxies will not update the Via header field, and some proxies will update both.

7.4. Transition

If a proxy gets incoming requests with X-Forwarded-* header fields present, it is encouraged to convert these into the header field described in this document, if it can be done in a sensible way. If the request only contains one type, ex. X-Forwarded-For, this can be translated to Forwarded, by prepending each element with "for=". Note that IPv6-addresses may not be quoted in X-Forwarded-For, but they are quoted in Forwarded.

```
X-Forwarded-For: 192.0.2.43, [2001:db8:cafe::17]
```

becomes:

```
Forwarded: for=192.0.2.43, for="[2001:db8:cafe::17]"
```

Special care must, however, be taken if, for example, both X-Forwarded-For and X-Forwarded-By exists. In such cases, it may not be possible to do a conversion, since it is not possible to know in what order the already existing fields were added. Also, note that removing the X-Forwarded-For header field may cause issues for parties that have not yet implemented support for this new header field.

7.5. Example usage

A request from a client with IP address 192.0.2.43 passes through a proxy with IP address 198.51.100.17, then through another proxy with IP address 203.0.113.60 before reaching a origin server. This could, for example, be an office client behind a corporate malware filter talking to a origin server through a reverse proxy.

- The HTTP request between the client and the first proxy has no Forwarded header field.

- The HTTP request between the first and second proxy has a "Forwarded: for=192.0.2.43" header field.

- The HTTP request between the second proxy and the origin server has a "Forwarded: for=192.0.2.43, for=198.51.100.17;by=203.0.113.60;proto=http;host=example.com" header field.

Note that, at some points in a connection chain, the information might not be updated in the Forwarded header field, either because of lack of support of this HTTP extension or because of a policy decision not to disclose information about this network component.

8. Security considerations

8.1. Header validity and integrity

The Forwarded HTTP header field cannot be relied upon to be correct, as it may be modified, whether mistakenly or for malicious reasons, by every node on the way to the server, including the client making the request.

One approach is to verify the correctness of proxies and to whitelist them as trusted. This approach has at least two weaknesses. First, the chain of IP addresses listed before the request came to the proxy cannot be trusted. Second, unless the communication between proxies and the end point is secured, the data can be modified by an attacker with access to the network.

8.2. Information leak

The Forwarded HTTP header field can reveal internal structures of the network setup behind the NAT or proxy setup, which may be undesired. This can be addressed either by using obfuscated elements, preventing the internal nodes from updating the HTTP header field, or by having an egress proxy removing entries that reveals internal network information.

This header field should never be copied into response messages by origin servers or intermediaries, as it can reveal the whole proxy chain to the client. As a side effect, special care must be taken in hosting environments not to allow the TRACE request where the Forwarded field is used, as it would appear in the body of the response message.

8.3. Privacy considerations

From an end-user perspective, intermediate proxies in the request path are either known or unknown. Hidden proxies using this extension will preserve the information of a direct connection, and

thus it has no end-user privacy impact.

Proxies that are known to the end user, such as explicitly configured proxies, using this extension will not anonymize the end-user IP address. This extension, however, only standardizes the format for forwarding client connection information. There are already deployed proxies supporting this feature, so there is no new privacy risk in that proxies are thought to be anonymizing, but in reality are not.

A proxy that is intended to anonymize the request should be very careful to use this header field at all.

9. IANA considerations

This document specifies the HTTP header listed below, which should be added to the permanent HTTP header registry defined in [RFC3864].

Header field: Forwarded
Applicable protocol: http/https
Status: standard
Author/Change controller:
    IETF (iesg@ietf.org)
    Internet Engineering Task Force
Specification document(s): this specification (Section 4)
Related information: None

The Forwarded header field contains parameters for which IANA is to create and maintain a new registry entiteled "HTTP Forwarded parameters". Initial registrations are given below; for future assignments, RFC is required [RFC5226]. The author should consider security- and privacy aspects and, if there are any, include such sections in the RFC. New parameters and their values MUST conform the forwarded-pair as defined in ABNF in Section 4. Further, a short description should be provided in the registration.

Parameter name	Description	Definition
by	IP-address of incoming interface of a proxy	Section 5.1
for	IP-address of client making a request through a proxy	Section 5.2
host	Host header field of the incoming request	Section 5.3

proto	Application protocol used for incoming request	Section 5.4

Table 1: Initial assignments

10. References

10.1. Normative references

[I-D.ietf-httpbis-p1-messaging]
Fielding, R., Lafon, Y., and J. Reschke, "HTTP/1.1, part 1: URIs, Connections, and Message Parsing", draft-ietf-httpbis-p1-messaging-19 (work in progress), March 2012.

[I-D.ietf-httpbis-p4-conditional]
Fielding, R., Lafon, Y., and J. Reschke, "HTTP/1.1, part 4: Conditional Requests", draft-ietf-httpbis-p4-conditional-19 (work in progress), March 2012.

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3864] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", BCP 90, RFC 3864, September 2004.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.

[RFC4395] Hansen, T., Hardie, T., and L. Masinter, "Guidelines and Registration Procedures for New URI Schemes", BCP 35, RFC 4395, February 2006.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, August 2010.

10.2. Informative references

[RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, June 2011.

Appendix A. Change Log (to be removed by RFC Editor before publication)

A.1. Since draft-petersson-forwarded-for-00

Added IANA considerations.

Expanded scope and add parameterized list.

A.2. Since draft-petersson-forwarded-for-01

Removed "x-" from private extensions.

Allow for any protocol name.

Rename kv-v to forwarded-element and kv to forwarded-value.

Add informative reference RFC6269.

A.3. Since draft-petersson-forwarded-for-02

Name change to draft-ietf-appsawg-http-forwarded-00.

Updated proto in list under section 5 Parameters.

Remove "hidden" but mention _hidden as an example in 6.3 Obfuscated identifier.

Clarify that IPv6-addresses must be enclosed by square brackets.

Restrict ext-value: do not allow "," or ";".

A.4. Since draft-ietf-appsawg-http-forwarded-00

Write IP address instead of IP number.

Remove BNF for IP addresses.

A.5. Since draft-ietf-appsawg-http-forwarded-01

Refer to httpbis instead of RFC2616. Thereby also change to RFC5234 ABNF.

Split up ABNF to be more general on top level.

Add some comments on draft-ietf-httpbis-p2-semantics-19#section-3.1 to "Implementation Considerations"

Removal of ABNF appendix.

Merging of the sections "Private extensions" and "Future extensions".

A.6. Since draft-ietf-appsawg-http-forwarded-02

Require obfport to start with an underscore.

Include "._-" as valid characters in obfnode.

Remove MAY-references from section 5.

Add a section about the relation to the via-header field.

Add some privacy considerations.

Encourage proxies to convert X-Forwarded-* to this format, when possible.

Mention and demonstrate that IPv6-addresses must be quoted.

Add motivation for the obfnode.

Add some notes on when this header field should be preserved or not.

Fix some typos and make some clarifications.

A.7. Since draft-ietf-appsawg-http-forwarded-03

Require that each parameter only occur once per instance.

Request for a new registry at IANA.

Authors' Addresses

Andreas Petersson
Opera Software
S:t Larsgatan 12
Linkoping SE-582 24

Email: pettson@opera.com

Martin Nilsson
Opera Software
S:t Larsgatan 12
Linkoping SE-582 24

Email: nilsson@opera.com