Workgroup: ohai
Internet-Draft:
draft-rdb-ohai-feedback-to-proxy-04
Published: 10 July 2022
Intended Status: Standards Track
Expires: 11 January 2023
Authors: T. Reddy D. Wing M. Boucadair
Akamai Citrix Orange
R. Polli
Team Digitale, Italian Government

Oblivious Relay Feedback

Abstract

To provide equitable service to clients, servers often rate-limit incoming requests, for example, based upon the source IP address. However, oblivious HTTP removes the ability for the server to distinguish amongst clients so the server can only rate-limit traffic from the oblivious relay. This harms all clients behind that oblivious relay.

This specification enables a server to convey rate-limit information to an oblivious relay, which can use it to apply rate-limit policies on clients. Cooperating oblivious relays can thus provide more equitable service to their distinguishable clients without impacting on all clients behind that oblivious relay.

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 11 January 2023.

Copyright Notice

Table of Contents

1. Introduction

Oblivious HTTP [OHTTP] requires three parties to exchange HTTP messages: the client, the relay, and the target (formally, the Oblivious Gateway Resource and Oblivious Target Resource). Oblivious HTTP enables a client to send requests to a target in such a way that the target cannot tell whether two requests came from the same client, and the relay cannot see the contents of the requests.

Since clients are located behind a relay, a target cannot distinguish between well-behaving and malicious clients: an unexpected behavior from one or more clients can then impact on all the intermediated clients, as described in Section 8.2.1 of [OHTTP]. This can be problematic when the target implements rate limiting

policies based on an information masked by the intermediary, such as the source IP address.

This document defines a mechanism that allows Oblivious gateway and target resource to provide rate-limit information to an Oblivious relay via the RateLimit fields defined in [RATELIMIT]. This is useful when such servers identify traffic anomalies or unexpected request volumes. The Oblivious relay can then use this information to apply rate-limit policies on clients.

While [RATELIMIT] provides enough information to generic clients to shape their request policy and avoid being throttled out, this specification allows an Oblivious gateway and target resource to indicate their RateLimit information is intended for the Oblivious relay (rather than to the client).

How an Oblivious relay can use this information to avoid being throttled out or shape its request policy is outside the scope of this specification.

2. Terminology

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.

The terms "content", "receiver", "request", and "response" are to be interpreted as described in [HTTP].

The terms "Encapsulated request", "Encapsulated response", "Oblivious relay resource", "Oblivious gateway resource", "Oblivious target resource", and "Client" are to be interpreted as described in [OHTTP].

The collective term "Oblivious resource" indicates either an "Oblivious gateway resource" or an "Oblivious target resource".

The terms "quota policy", "service limit", "expiring limit", and "RateLimit fields" are to be interpreted as described in [RATELIMIT].

This document uses the Integer type from [STRUCTURED-FIELDS].

3. Providing RateLimit Information to an Oblivious Proxy

An Oblivious resource that uses RateLimit fields [RATELIMIT] to return service limit information MAY add the "ohttp-target" quota policy parameter defined in Section 4 to signal to the receiver that

the associated quota policy is intended for an Oblivious relay. For example, when an Oblivious target identifies a high frequency or high volume anomalies in the HTTP requests it would include the "ohttp-target" parameter.

The term "Oblivious Relay Feedback" denotes both the mechanism described in this specification and the complete set of RateLimit fields together with the "ohttp-target" parameter.

To know whether the RateLimit fields provides Oblivious Relay Feedback (see Section 3.1), an Oblivious relay MUST:

1. Identify the quota policy associated to the expiring limit.

2. Check whether the "ohttp-target" parameter is present and its syntax is correct.

In the example shown in Figure 1, the expiring limit value is "100", so the associated quota policy is the second one. This quota policy includes the "ohttp-target" parameter: this indicates that the RateLimit fields are intended for an Oblivious relay.

```
RateLimit-Limit: 100
RateLimit-Policy: 10;w=1, 100;w=60;ohttp-target=1
RateLimit-Remaining: 8
RateLimit-Reset: 15
```

Figure 1: An Example of Oblivious Proxy Feedback.

4. The ohttp-target Quota Policy Parameter

4.1. ohttp-target Parameter

The following quota policy parameter is defined for the RateLimit-Policy field [RATELIMIT]:

ohttp-target: Indicates that the associated quota policy provides Oblivious Relay Feedback. This parameter is OPTIONAL.

The "ohttp-target" parameter has the following syntax:

```
ohttp-target = sf-integer
```

Its value MUST be an Integer (Section 3.3.1 of [STRUCTURED-FIELDS]) and indicates whether the quota policy is applicable to all the clients that are serviced by the Oblivious relay or applicable only to a specific client. The "ohttp-target" parameter MUST have one of the following values:

1:
Indicates that RateLimit fields are applicable to all the clients that are serviced by the same Oblivious relay.

2: Indicates that RateLimit fields are applicable only to the offending client. For example, this value is used if the client is attacking the server (e.g., the client is using an abnormal header that matches an attack pattern).

The Oblivious relay does not immediately act to rate-limit the traffic from the client but starts maintaining a count of responses to the client with "ohttp-target" parameter value set to "2" marked as "potential malicious requests" and responses without the parameter marked as "legitimate requests".

The Oblivious relay can rate-limit requests from the offending client for a certain duration only when the client has a high ratio of "potential malicious requests" to "legitimate requests". In other words, the Oblivious relay will rate-limit requests from a client if the target has seen an attack pattern in multiple requests from that same client. A malicious client sends malformed HTTP requests, whereas a benign client sends valid HTTP requests. The malformed HTTP requests are linkable whereas the valid HTTP requests are not linkable. Most importantly, the target will not be able to partition the anonymity set of legitimate clients.

Other values MUST cause the parameter to be ignored.

The "ohttp-target" parameter MUST NOT appear more than once in a quota policy. If the parameter is malformed or its value is invalid, it MUST be ignored, and the receiving Oblivious relay MUST NOT attempt to fix neither the parameter nor its value. That is, the RateLimit fields must not be considered as providing Oblivious Relay Feedback.

4.2. Processing the ohttp-target Parameter

An Oblivious relay receiving RateLimit fields providing Oblivious Relay Feedback will do the following:

1. It MUST remove the RateLimit fields from the response, since they are not intended to be forwarded to clients.

2. It MAY add a new set of RateLimit fields that are intended to be forwarded to a client.

An Oblivious gateway resource receiving RateLimit fields providing Oblivious Relay Feedback MUST proceed as follows:

1. Remove the RateLimit fields from the HTTP response, since they are not intended to be forwarded to the client. It, then, encapsulates the HTTP response.

2. Add the above RateLimit fields to the response containing the encapsulated response sent to the Oblivious relay, so that the Oblivious relay can access them.

If the RateLimit fields along with the "ohttp-target" parameter are generated by the Oblivious gateway resource before removing the protection (including being unable to remove the encapsulation for any reason)(Section 6.2 of [OHTTP]), it will result in the RateLimit fields added in the response being sent without protection in response to a POST request from a client.

While this specification does not mandate specific traffic shaping actions for Oblivious proxies in addition to the ones indicated in [RATELIMIT], an Oblivious relay failing to reshape traffic from a specific client or from all the clients according to the received Oblivious Relay Feedback can experience different levels of service denial by the Oblivious gateway and target resources. There is no explicit mechanism for an Oblivious relay to indicate to the server that the rate-limit information was processed or was ignored.

5. The attack-severity Quota Policy Parameter

The following quota policy parameter is defined for the RateLimit-Policy field defined in [RATELIMIT]:

attack-severity: Is used by the Oblivious resource to convey the likeliness that an HTTP request is malicious. This parameter is OPTIONAL.

```
attack-severity = sf-string
```

Note that sf-string is defined in Section 3.3.3 of [STRUCTURED-FIELDS].

The value of the "attack-severity" parameter is a String (Section 3.3.3 of [RFC8941]) that takes one of the values defined in [SEVERITY]. This parameter MUST NOT appear more than once in a quota policy. If the parameter is malformed or its value is invalid, the parameter MUST be ignored, and the relays MUST NOT attempt to fix neither the parameter nor the value.

6. Use of The ohttp-target Quota Policy Parameters: An Example

The example depicted in Figure 2 illustrates the use of the "ohttp-target" parameter. An oblivious target resource receives a malformed request and uses the source IP address to identify that it was an encapsulated request decapsulated by an oblivious gateway resource. The Oblivious target resource generates a 400 response and adds the RateLimit fields along with the "ohttp-target" quota policy parameter. The oblivious gateway resource proceeds as follows:

1. Copy the RateLimit fields from the original response.

2. Remove them from the original response before encapsulating it.

3. Generate a single 200 response containing the encapsulated response for the oblivious relay resource along with the copied RateLimit fields.

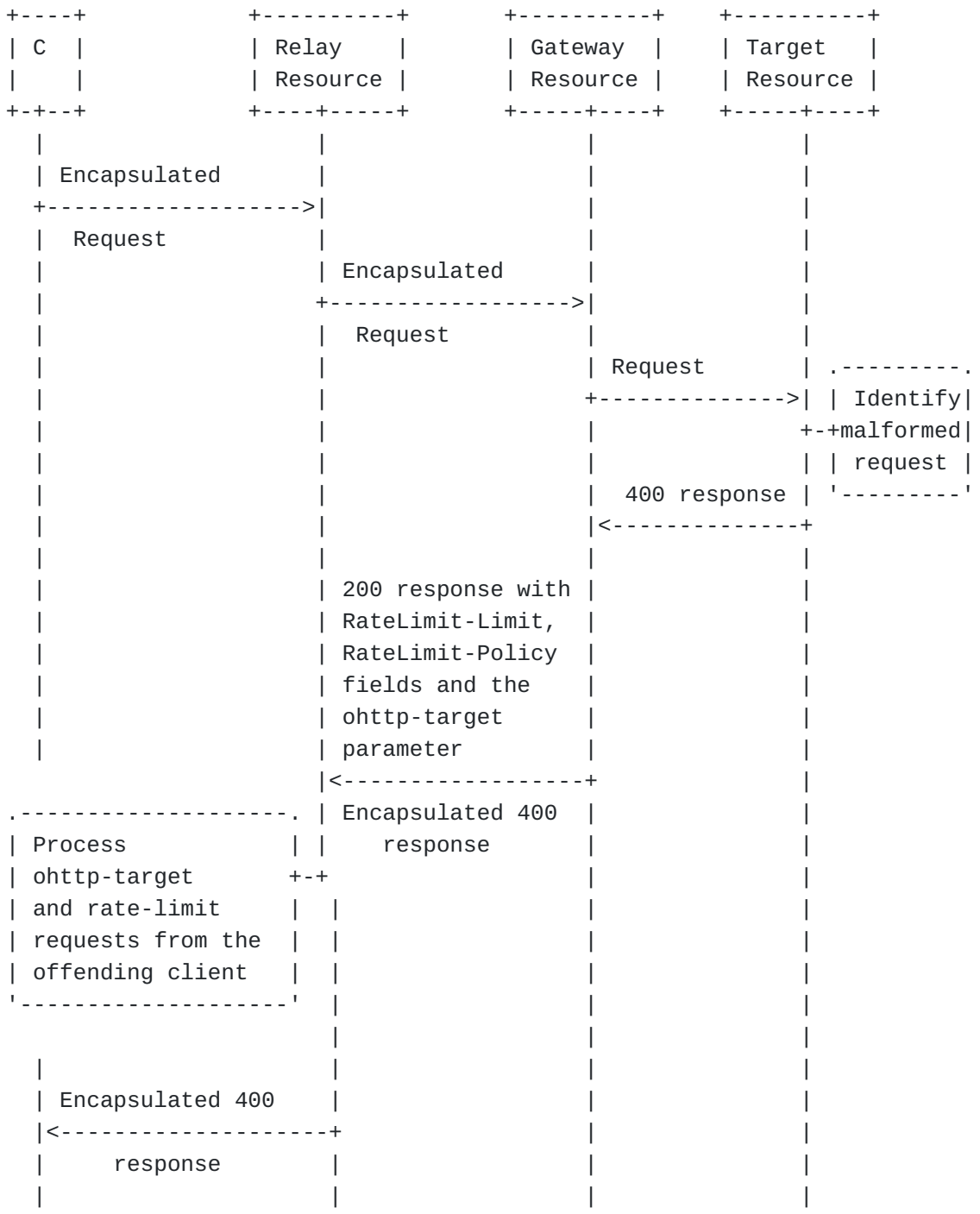

Figure 2: An Example of Ratelimit Feedback to Proxy

The response that is generated by the Oblivious gateway resource is depicted in Figure 3. This response includes an unregistered, informative "comment" quota policy parameter providing the rationale for the "attack- severity".

```
=============== NOTE: '\' line wrapping per RFC 8792 ================

  HTTP/1.1 200 OK
  Date: Wed, 27 March 2022 04:45:07 GMT
  Cache-Control: private, no-store
  RateLimit-Limit: 10
  RateLimit-Policy: 10;ohttp-target=2;attack-severity="high";\
comment="abnormal header matching a WAF rule"
  Content-Type: message/ohttp-res
  Content-Length: 38 <content is the encapsulated 400 response>
  ...encrypted content...
```

Figure 3: Example of a Response

7. Ohttp-Outside-Encap Header

The "Ohttp-Outside-Encap" header is defined in this specification (Section 9.2.1). Its purpose is to signal which HTTP headers will be removed by the Oblivious gateway.

When an Oblivious gateway resource sends an HTTP request to an Oblivious taget, it adds the "Ohttp-Outside-Encap" header to indicate which headers will be removed from the response.

On receipt of an HTTP response from the Oblivious target resource, the Oblivious gateway resource copies the header fields signaled in the associated request and removes those headers from the HTTP response. The Oblivious gateway then encapsulates the HTTP response. The Oblivious gateway resource adds the copied header fields and values to the response containing the encapsulated response, so that the Oblivious relay can access and act on them.

The "Ohttp-Outside-Encap" header is useful in deployments where the Oblivious gateway resource and Oblivious target resource are managed by separate entities.

Figure 4 describes the syntax using Augmented Backus-Naur Form (ABNF) of the header field, using the grammar defined in [RFC5234] and the rules defined in Section 5 of [RFC9110]. The field values of the header field conform to the same rules.

```
  Ohai-Outside-Encap =  header-field *( OWS "|" OWS header-field)
  header-field = token
```

Figure 4: Ohttp-Outside-Encap Header Syntax

Optional white space (OWS) is used as defined in Section 5.6.3 of [RFC9110].

An example is illustrated below:

```
Ohttp-Outside-Encap: RateLimit-Limit|RateLimit-Remaining|RateLimit-Reset
```

8. Security Considerations

The security considerations for the Oblivious HTTP protocol (Section 8 of [OHTTP]) as well as the ones for RateLimit fields (Section 6 of [RATELIMIT]) apply. The following sub-sections discuss security considerations specific to this specification.

8.1. Client and Oblivous Proxy Collusion

While Oblivious HTTP relies upon an Oblivious relay to prevent leaking the client identity to the Oblivious resources, it might be the case that the Oblivious relay colludes with clients in attacking Oblivious resources. RateLimit fields might disclose operational capacity information useful to design denial of service attacks or to circumvent defensive measures put in place by the Oblivious resources (Section 6.2 of [RATELIMIT]). The Oblivious target and gateway resources SHOULD convey Oblivious Relay Feedback only to trusted Oblivious proxies.

8.2. Attack Categories

Attacks against the Oblivious Gateway and Target Resources can be classified into three primary categories:

1. A client deliberately sends a malformed encapsulated request causing decryption failure or decryption overload failure on the oblivious gateway resource. This causes the oblivious gateway resource to send an error status code back to the oblivious relay.

2. A client deliberately sends an HTTP request that causes an HTTP error on the oblivious target resource. This might be a malformed HTTP request, or request for a missing resource.

3. A botnet performing an application layer denial of service attack (e.g. HTTP flood) against an Oblivious resource. Because each bot in a botnet makes seemingly legitimate network requests the traffic may appear "normal" in origin, nonetheless as a whole it not only can saturate the Oblivious resources, but also makes appear the Oblivious relay as an attacker. This might be too many requests from a single client, too many requests from the clients behind the same oblivious relay or too many requests from all clients on the Internet.

9. IANA Considerations

9.1. RateLimit Parameter Value Registrations

This specification requests IANA to add the following parameters to the "Hypertext Transfer Protocol (HTTP) RateLimit Parameters" registry defined in [RATELIMIT].

Field Name	Parameter Name	Description	Specification
RateLimit-Policy	ohttp-target	ohttp ratelimit	Section 3 of this document
RateLimit-Policy	attack-severity	ohttp ratelimit	Section 5 of this document

9.2. Registration of new HTTP Header Field

9.2.1. Ohttp-Outside-Encap Header

This section describes a header field for registration in the Permanent Message Header Field Registry [RFC3864].

Header field name
Ohttp-Outside-Encap

Applicable protocol
http

Status
Standard

Author/Change controller
IETF

Specification document(s)
RFC XXXX

Related information
This header field is only used for Oblivious HTTP.

10. Acknowledgements

Thanks to Lucas Pardue, Rich Salz, Martin Thomson, Christopher A. Wood and Brandon Williams for the discussion and comments.

11. References

11.1. Normative References

[HTTP] Fielding, R. T., Nottingham, M., and J. Reschke, "HTTP Semantics", Work in Progress, Internet-Draft, draft-ietf-httpbis-semantics-19, 12 September 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-httpbis-semantics-19>.

[OHTTP] Thomson, M. and C. A. Wood, "Oblivious HTTP", Work in Progress, Internet-Draft, draft-ietf-ohai-ohttp-01, 15 February 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-ohai-ohttp-01>.

[RATELIMIT] Polli, R. and A. M. Ruiz, "RateLimit Fields for HTTP", Work in Progress, Internet-Draft, draft-ietf-httpapi-ratelimit-headers-05, 6 July 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-httpapi-ratelimit-headers-05>.

[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>.

[RFC3864] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", BCP 90, RFC 3864, DOI 10.17487/RFC3864, September 2004, <https://www.rfc-editor.org/info/rfc3864>.

[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>.

[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>.

[RFC8941] Nottingham, M. and P-H. Kamp, "Structured Field Values for HTTP", RFC 8941, DOI 10.17487/RFC8941, February 2021, <https://www.rfc-editor.org/info/rfc8941>.

[RFC9110] 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/info/rfc9110>.

[STRUCTURED-FIELDS] Nottingham, M. and P-H. Kamp, "Structured Field Values for HTTP", RFC 8941, DOI 10.17487/RFC8941, February 2021, <https://www.rfc-editor.org/rfc/rfc8941>.

11.2. Informative References

[SEVERITY] IANA, "Incident Object Description Exchange Format v2 (IODEF)", <https://www.iana.org/assignments/iodef2/iodef2.xhtml#businessimpact-severity>.

Authors' Addresses

Tirumaleswar Reddy
Akamai
Embassy Golf Link Business Park
Bangalore 560071
Karnataka
India

Email: kondtir@gmail.com

Dan Wing
Citrix Systems, Inc.
4988 Great America Pkwy
Santa Clara, CA 95054
United States of America

Email: danwing@gmail.com

Mohamed Boucadair
Orange
35000 Rennes
France

Email: mohamed.boucadair@orange.com

Roberto Polli
Team Digitale, Italian Government

Email: robipolli@gmail.com