HTTP
Internet-Draft
Intended status: Standards Track
Expires: 9 March 2026

Y. Rosomakho
Zscaler

5 September 2025

Detecting Outdated Proxy Configuration
draft-rosomakho-httpbis-outdated-proxy-config-01

Abstract

This document defines a lightweight mechanism that allows explicit HTTP proxies to inform clients when their proxy configuration, such as a Proxy Auto-Configuration (PAC) file or Provisioning Domain (PvD) proxy configuration, is outdated. Clients signal to the proxy the configuration URL and the timestamp of when it was last fetched. In response, the proxy may indicate that a newer version of the configuration is available. This enables clients to refresh their configuration without relying on frequent polling or short expiration intervals. The mechanism is designed to be compatible with existing proxy deployment models and supports both PAC-based and PvD-based configurations.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://yaroslavros.github.io/httpbis-outdated-proxy-config/draft-rosomakho-httpbis-outdated-proxy-config.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-rosomakho-httpbis-outdated-proxy-config/.

Discussion of this document takes place on the HTTP Working Group mailing list (mailto:ietf-http-wg@w3.org), which is archived at https://lists.w3.org/Archives/Public/ietf-http-wg/.

Source for this draft and an issue tracker can be found at https://github.com/yaroslavros/httpbis-outdated-proxy-config.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

This Internet-Draft will expire on 9 March 2026.

Copyright Notice

Table of Contents

1. Introduction

Explicit HTTP proxies are widely deployed in enterprise and managed network environments to enforce routing policies, enable traffic inspection, or implement access control. Clients relying on such proxies typically obtain configuration through a Proxy Auto-Configuration (PAC) file [PAC-FILE] or a Provisioning Domain (PvD) [PROXY-PVD]. In many deployments, it is necessary to update these configurations in response to operational changes such as service onboarding, emergency routing adjustments, or policy corrections.

Currently, clients have limited mechanisms to detect whether the proxy configuration they are using is stale. As a result, PAC files are often polled at short intervals, and PvD expiration times are configured with low values to accelerate refreshes. These approaches are inefficient and may introduce unnecessary network traffic or delay the application of important updates.

This document defines a simple mechanism that enables a proxy to inform a client that its current configuration is outdated. The client includes in its request a structured field indicating the URL of the PAC file or PvD and the timestamp of when it last fetched the configuration. If the proxy recognizes the configuration and determines that a newer version is available, it may respond with a boolean indicator prompting the client to refresh the configuration.

This mechanism applies to forms of explicit proxying over HTTP where there is a clear separation between client headers intended for the proxy and headers intended for the origin server. This includes:

* HTTP CONNECT as defined Section 9.3.6 of [HTTP]

* [CONNECT-UDP]

* [CONNECT-IP]

* Templated [CONNECT-TCP]

This mechanism is not applicable to HTTP/1.1 proxying modes that use the "absolute-form" request URI defined in Section 3.2.2 of [HTTP/1.1] with HTTP methods other than CONNECT.

The mechanism is optional, compatible with existing protocols, and requires no persistent state. It allows clients to discover configuration updates proactively while preserving the existing operational model.

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.

3. Protocol Overview

To enable detection of stale proxy configuration, clients may include a Proxy-Config HTTP header field in requests sent to an explicit proxy. This header communicates the URL of the proxy configuration (such as a PAC file or PvD) and the timestamp of when the client last fetched it.

Upon receiving a request containing the Proxy-Config header, a proxy that supports this mechanism compares the provided timestamp with the most recent update time of the corresponding configuration. If the proxy determines that the client configuration is outdated, it can signal this condition using the Proxy-Config-Stale response header.

This mechanism is optional and advisory. Proxies are not required to track or respond to client configuration metadata, and clients are not required to act immediately upon receiving a staleness indication. The mechanism is designed to supplement, not replace, existing configuration refresh behaviors.

4. Proxy-Config Header

The Proxy-Config request header field is used by a client to inform an explicit proxy about the proxy configuration it is currently using. The field conveys a dictionary structured field as defined in Section 3.2 of [STRUCTURED-FIELD] with the following keys:

* url (optional): A string identifying the URL from which the client fetched the configuration. This may point to a PAC file or a PvD configuration. It MAY be omitted in the following cases:

  - The client is using the default PvD URI based on proxy hostname and ".well-known/pvd" path as defined in Section 4.1 of [PVDDATA].

  - The configuration was provisioned through a mechanism that is not associated with a specific URL, such as enterprise device management or a local policy engine

* fetched (required): A date value indicating when the client last fetched the configuration. The value MUST use the Date format defined in Section 3.3.7 of [STRUCTURED-FIELD].

4.1. Handling Unknown or Sensitive URLs

Clients MUST NOT include URLs that expose local system information (e.g., file:// URLs). Clients SHOULD limit use of the Proxy-Config header to contexts where it does not introduce privacy or security risks, such as trusted or encrypted proxy connections.

4.2. Examples

A client using a PAC file retrieved from https://config.example.net/proxy.pac at 2025-06-01T00:00:00Z MAY include the following header:

```
Proxy-Config: url="https://config.example.net/proxy.pac", fetched=@1748736000
```

Figure 1: Example of Proxy-Config header with url field

If the client is using the default PvD URI associated with the proxy hostname, it may omit the url key:

```
Proxy-Config: fetched=@1748736000
```

Figure 2: Example of Proxy-Config header without url field

5. Proxy-Config-Stale Header

The Proxy-Config-Stale response header field is used by a proxy to inform the client whether its current proxy configuration is considered outdated. This allows the client to make informed decisions about whether to refresh the configuration.

The field is a boolean as defined in Section 3.3.6 of [STRUCTURED-FIELD], where:

* ?1 indicates that the configuration used by the client is stale and a newer version is available.

* ?0 indicates that the configuration used by the client is current.

The proxy MUST only include this header if it has received a valid Proxy-Config header from the client and is able to recognize and evaluate the configuration URL. If the proxy does not recognize the provided configuration URL, does not track updates for it, or does not support this mechanism, it MUST omit the Proxy-Config-Stale header.

The Proxy-Config-Stale header MAY appear in both successful and error responses, except for responses related to client authentication (e.g., 407 Proxy Authentication Required). Including the header in such authentication-related responses could result in unintended exposure of configuration metadata and may interfere with authentication workflows.

The Proxy-Config-Stale field is advisory. Its presence does not affect the processing of the current request or response. Clients MAY choose how and when to act upon the information, including deferring configuration refresh until convenient.

6. Security Considerations

Clients using the Proxy-Config header field must take care to avoid disclosing sensitive information in the URL or metadata they send to the proxy.

6.1. Avoiding Sensitive URLs

Clients MUST NOT include configuration URLs that reveal local system details, such as file:// URIs or paths containing user-specific or internal directory structures. To reduce exposure, clients SHOULD only use this mechanism when proxy configuration is relevant to the proxy (e.g., hosted on the proxy or a part of the same enterprise domain).

6.2. Trusted Communication Channels

The Proxy-Config header is intended for use over secure and trusted communication channels. Clients SHOULD send this header only when using TLS or when otherwise confident that the request is not observable or modifiable by unauthorized intermediaries.

6.3. Authentication-related Responses

Proxies MUST NOT include the Proxy-Config-Stale header in responses related to client authentication (e.g., 407 Proxy Authentication Required). This avoids potential leakage of client configuration metadata during authentication flows that may be visible to other components or exposed through logging or error handling.

6.4. Denial-of-Service Considerations

A misconfigured or malicious proxy could include Proxy-Config-Stale: ?1 in every response, causing the client to repeatedly refetch proxy configuration. This behavior can lead to excessive network traffic, CPU usage, or degraded performance on the client, particularly in environments where configuration retrieval is resource-intensive.

To mitigate this risk, clients MUST implement appropriate rate limiting or throttling mechanisms when acting on stale configuration indications. For example, a client may choose to ignore repeated ?1 responses within a minimum refresh interval or apply exponential backoff when encountering multiple stale signals in quick succession.

Clients SHOULD validate the authenticity and integrity of any fetched configuration before applying it, and ensure that configuration refreshes do not interfere with ongoing connection or session state.

6.5. Inapplicability to Non-CONNECT Proxying Modes

This mechanism is not intended for use with HTTP/1.1 proxying models that rely on the "absolute-form" request URI defined in Section 3.2.2 of [HTTP/1.1] with methods other than CONNECT. In such configurations, all client headers may be forwarded by the proxy to the destination server. This can result in unintended disclosure of internal configuration metadata.

Clients MUST ensure that the Proxy-Config header is only sent when the proxying mode provides a clear separation between hop-by-hop headers (intended for the proxy) and end-to-end headers (intended for the destination server). This includes CONNECT-based methods such as CONNECT (Section 9.3.6 of [HTTP]), [CONNECT-UDP], [CONNECT-IP] and templated [CONNECT-TCP]. These methods establish a tunnel or encapsulation that ensures Proxy-Config header is visible only to the proxy and is not forwarded to the destination server even if the proxy does not recognize it.

7. IANA Considerations

This document registers the following HTTP header fields in the "Hypertext Transfer Protocol (HTTP) Field Name Registry":

Proxy-Config

* Field Name: Proxy-Config

* Status: permanent

* Reference: this document

Proxy-Config-Stale

* Field Name: Proxy-Config-Stale

* Status: permanent

* Reference: this document

8. References

8.1. Normative References

[HTTP] 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/rfc/rfc9110>.

[HTTP/1.1] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP/1.1", STD 99, RFC 9112, DOI 10.17487/RFC9112, June 2022, <https://www.rfc-editor.org/rfc/rfc9112>.

[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/rfc/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/rfc/rfc8174>.

[STRUCTURED-FIELD] Nottingham, M. and P. Kamp, "Structured Field Values for HTTP", RFC 9651, DOI 10.17487/RFC9651, September 2024, <https://www.rfc-editor.org/rfc/rfc9651>.

8.2. Informative References

[CONNECT-IP] Pauly, T., Ed., Schinazi, D., Chernyakhovsky, A., Khlewind, M., and M. Westerlund, "Proxying IP in HTTP", RFC 9484, DOI 10.17487/RFC9484, October 2023, <https://www.rfc-editor.org/rfc/rfc9484>.

[CONNECT-TCP] Schwartz, B. M., "Template-Driven HTTP CONNECT Proxying for TCP", Work in Progress, Internet-Draft, draft-ietf-

httpbis-connect-tcp-09, 30 June 2025,
<https://datatracker.ietf.org/doc/html/draft-ietf-httpbis-connect-tcp-09>.

[CONNECT-UDP]
Schinazi, D., "Proxying UDP in HTTP", RFC 9298, DOI 10.17487/RFC9298, August 2022, <https://www.rfc-editor.org/rfc/rfc9298>.

[PAC-FILE] Mozilla, "Proxy Auto-Configuration (PAC) file", May 2025, <https://developer.mozilla.org/en-US/docs/Web/HTTP/Guides/Proxy_servers_and_tunneling/Proxy_Auto-Configuration_PAC_file>.

[PROXY-PVD]
Pauly, T., Damjanovic, D., and Y. Rosomakho, "Communicating Proxy Configurations in Provisioning Domains", Work in Progress, Internet-Draft, draft-ietf-intarea-proxy-config-07, 26 July 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-intarea-proxy-config-07>.

[PVDDATA] Pfister, P., Vyncke, ., Pauly, T., Schinazi, D., and W. Shao, "Discovering Provisioning Domain Names and Data", RFC 8801, DOI 10.17487/RFC8801, July 2020, <https://www.rfc-editor.org/rfc/rfc8801>.

Acknowledgments

Thanks to Tommy Pauly and Dragana Damjanovic for reviewing the concept and providing initial feedback.

Author’s Address

Yaroslav Rosomakho
Zscaler
Email: yrosomakho@zscaler.com