Network Working Group
Internet-Draft
Intended status: Informational
Expires: 23 October 2026

G. Illyes
Independent

21 April 2026

A JSON-Based Format for Publishing IP Ranges of Automated HTTP Clients
draft-illyes-webbotauth-jafar-00

Abstract

This document defines a standardized JSON format for operators of automated HTTP clients (e.g., web crawlers, AI bots) to publicly disclose their IP address ranges. A consistent, machine-readable format for IP range publication simplifies the task of identifying and verifying legitimate automated traffic, thereby decreasing maintenance load on website operators while reducing the risk of inadvertently blocking beneficial clients. This specification codifies and extends common existing practices to provide a simple yet extensible format that accommodates a variety of use cases.

Discussion Venues

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

Discussion of this document takes place on the AI Preferences Working Group mailing list (ai-control@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/ai-control/.

Source for this draft and an issue tracker can be found at https://github.com/garyillyes/jafar.

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 23 October 2026.

Table of Contents

1. Introduction

This document specifies a data format using JavaScript Object Notation ([JSON]). It is intended for the publication of IP address ranges associated with automated HTTP clients. The scope of this specification is limited to the syntax and semantics of the JSON file itself. It does not specify the transport mechanism for retrieving the file. It also does not prescribe specific policies for how consumers should use the data (e.g., allowlisting, rate-limiting, or monitoring). This format is intended to complement, not replace, other established methods for crawler verification. Techniques such as forward-confirmed reverse DNS (FCrDNS) lookups remain a vital part of a comprehensive verification strategy. This specification provides a scalable, machine-readable component that can be

integrated into a multi-layered verification process.

2. Format

2.1. Top-Level Object Definition

An IP range publication file MUST be a single JSON object. The text encoding of the file MUST be UTF-8. This top-level object serves as the root container for metadata relevant to the file and the list of IP prefixes.

The top-level object contains the fields defined in Table 1.

Field Name	Type	Requirement	Description
synctoken	String	OPTIONAL	An opaque synchronization token that changes whenever there is a change to any metadata associated with one or more prefixes.
creationTime	String	MUST	An ISO 8601 timestamp in the "Z" timezone (UTC) indicating when the file was generated (e.g., "2025-08-15T14:30:00Z").
notes	String	OPTIONAL	A human-readable string containing any relevant notes, disclaimers, or comments from the publisher. This can be used to provide context that is not captured by the structured data.
prefixes	Array	MUST	An array of Prefix Objects, as defined in Section 2.3. Each object in the array describes an IPv4 or IPv6 address range. This array MAY be empty if the publisher currently has no active IP ranges to declare.

Table 1

2.2. The prefixes Array

The prefixes member of the top-level object MUST contain an array of JSON objects. Each of these objects is a Prefix Object, as defined in Section 2.3.

To simplify implementation for consumers, there MUST be a single, unified array for both IPv4 and IPv6 prefixes.

2.3. The Prefix Object

Each object within the prefixes array represents a single IP address range and its associated metadata.

A Prefix Object MUST contain exactly one of either the ipv4Prefix or ipv6Prefix field. An object containing both or neither of these fields is invalid and MUST be ignored by consumers. The fields are defined in Table 2.

Field Name	Type	Requirement	Description
ipv4Prefix	String	CONDITIONAL	The IPv4 address range in Classless Inter-Domain Routing ([CIDR]) notation (e.g., "66.249.64.0/20"). This field MUST be present if the ipv6Prefix field is absent.
ipv6Prefix	String	CONDITIONAL	The IPv6 address range in CIDR notation (e.g., "2001:4860:4000::/36"). This field MUST be present if the ipv4Prefix field is absent.
services	Array	OPTIONAL	An array of publisher-defined, case-sensitive strings identifying the services associated with this prefix. For example ["Examplebot", "AdsBot-Example"]. This allows consumers to apply more granular policies.

Table 2

2.4. Extensibility

The JAFAR format is extensible. Both the top-level object and Prefix Objects MAY include additional fields not defined in this specification. Consumers MUST ignore any fields they do not recognize.

3. Processing and Consumption Rules

3.1. Fetching and Caching

Consumers SHOULD fetch the machine-readable IP range publication file from a stable URL provided by the publisher. The file location MUST be disclosed by the publisher of the file on the page that describes the crawler as specified by [CBCP].

Publishers SHOULD update the file when there is any change to the prefixes, or at least every 24 hours, even if the only update is to creationTime. Consumers SHOULD implement a polling mechanism to check for updates at a reasonable interval, such as once every 24 hours. Consumers MUST NOT poll more frequently than once per hour unless explicitly permitted by the publisher’s documentation or HTTP caching headers.

To minimize server load for the publisher and reduce unnecessary bandwidth usage for the consumer, consumers MUST respect standard HTTP caching headers specified in [HTTP-CACHING] that may be present in the response, such as Cache-Control, ETag, and Last-Modified. Publishers SHOULD provide these headers to facilitate efficient caching.

3.2. Handling Format Versioning

To ensure long-term stability and allow for future evolution of this specification, consumers MUST inspect the version optional parameter of the application/jafar+json media type, as specified in the Content-Type HTTP header. The version is expressed as "major.minor". If the version parameter is absent, consumers SHOULD assume the latest stable version of this specification they are programmed to handle.

* Major Version Changes: A change in the major version number (e.g., from "1.0" to "2.0") indicates a non-backward-compatible change to the specification. If a consumer encounters a file advertised with a major version number greater than the major version it is programmed to handle, the consumer MUST NOT attempt to parse the file. It SHOULD treat this situation as an error and MAY continue to use its last known valid list until it can be updated to support the new version. This prevents the misinterpretation of data from a significantly altered schema.

* Minor Version Changes: A change in the minor version number (e.g., from "1.0" to "1.1") indicates the addition of new features or fields that are backward-compatible. For example, a new OPTIONAL field might be added to the Prefix Object. A consumer programmed

to handle version "1.0" MUST be able to correctly parse a file with version "1.1". The minor version number increases numerically independently of the major version number, for instance: 1.9 -> 1.10 -> 1.11. The consumer MUST ignore any unrecognized fields or properties within the JSON objects.

3.3. Prefix Aggregation and Specificity

A publication file MAY contain overlapping IP address ranges. For instance, a publisher might list a broad range like 198.51.100.0/22 with a generic service tag, and also list a more specific range within it, such as 198.51.100.0/24, with a more specific service tag.

When a consumer evaluates a specific IP address against the list, it MAY match multiple Prefix Objects. In such cases, the consumer’s logic SHOULD use the data from the Prefix Object with the most specific CIDR range (i.e., the one with the largest prefix length) for that IP address.

4. Use Cases and Examples

4.1. Example 1: Basic Publication

This example demonstrates a minimal, valid file for a publisher with a single type of automated client. It uses only the required fields, making it simple to generate and consume.

```
{
  "creationTime": "2025-08-15T14:30:00Z",
  "prefixes": [
    {
      "ipv4Prefix": "66.249.64.0/20"
    },
    {
      "ipv4Prefix": "34.64.0.0/12"
    },
    {
      "ipv6Prefix": "2001:4860:4000::/36"
    }
  ]
}
```

4.2. Example 2: Advanced Publication with Multiple Services

This example illustrates a more complex file from a large cloud provider that uses optional fields to provide richer context. It differentiates between IP ranges used for a general web crawler, a specialized ads crawler, and a user-triggered fetching service. This allows consumers to implement more granular access policies.

```
{
  "creationTime": "2025-08-15T14:30:00Z",
  "prefixes": [
    {
      "ipv4Prefix": "66.249.64.0/24",
      "services": ["ExampleCloud-Crawler", "ExampleCloud-Ads"]
    },
    {
      "ipv6Prefix": "2001:4860:4860::/48",
      "services": ["ExampleCloud-Fetcher"]
    }
  ]
}
```

4.3. Example 3: Aggregated Publication by a Third Party

This example shows how a third-party security provider or an open-source project could aggregate IP lists from multiple publishers into a single file. The service field is used to attribute each prefix to its original source, allowing consumers to apply policies based on the original publisher. This reflects the use case of services that provide curated lists of known bots.

```
{
  "synctoken": "20260410223000",
  "creationTime": "2026-04-10T22:30:00Z",
  "notes": "Aggregated feed of verified automated clients. Attribution is maintained via
the services array. Data refreshed every 24 hours.",
  "prefixes": [
    {
      "ipv4Prefix": "192.0.2.0/24",
      "services": [
        "SearchEngine-A-Crawler",
        "SearchEngine-A-ImageBot"
      ]
    },
    {
      "ipv4Prefix": "198.51.100.0/24",
      "services": [
        "SocialMedia-B-Preview"
      ]
    },
    {
      "ipv6Prefix": "2001:db8:abc::/48",
      "services": [
        "TechCo-C-HealthCheck",
        "TechCo-C-Ads"
      ]
    }
  ]
}
```


5. 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.

6. Security Considerations

* Spoofing: Malicious actors claiming IP ranges they don’t own. Mitigate with FCrDNS

* DoS: Clients being tricked into polling the JSON file too frequently. Mitigate by setting a reasonable polling limit, rely on caching (add 9110)

* Data Integrity: The risk of the file being intercepted and modified. Mitigated by HTTPS, which should be mandated for the transport.

7. IANA Considerations

IANA is requested to register the following media type in the "Media Types" registry:

* Type name: application

* Subtype name: jafar+json

* Required parameters: N/A

* Optional parameters:

  - version: The version of the JAFAR specification the file conforms to (e.g., "1.0"). If absent, it defaults to the latest stable version supported by the consumer.

* Encoding considerations: binary

* Security considerations: See Section 6 (Section 6)

* Interoperability considerations: N/A

* Published specification: This Document

8. Normative References

[CBCP] Illyes, G., K端hlewind, M., and K. Aj, "Crawler best practices", Work in Progress, Internet-Draft, draft-illyes-aipref-cbcp-04, 9 April 2026, <https://datatracker.ietf.org/doc/html/draft-illyes-aipref-cbcp-04>.

[CIDR] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, DOI 10.17487/RFC4632, August 2006, <https://www.rfc-editor.org/rfc/rfc4632>.

[HTTP-CACHING] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Caching", STD 98, RFC 9111, DOI 10.17487/RFC9111, June 2022, <https://www.rfc-editor.org/rfc/rfc9111>.

[IPV6] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/rfc/rfc4291>.

[JSON] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/rfc/rfc8259>.

[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>.

Acknowledgments

TODO acknowledge.

Author’s Address

Gary Illyes
Independent
Email: synack@garyillyes.com