opsawg
Internet-Draft
Intended status: Standards Track
Expires: 27 August 2026

X. Gao, Ed.
S. Zhang
China Unicom
C. Lin
New H3C Technologies

23 February 2026

Requirements and Information Elements for Application Layer Information Export in IP Flow Information Export (IPFIX)
draft-gao-opsawg-ipfix-term-and-app-01

Abstract

This document explains the requirements for exporting application layer information and specifies the information elements used in IPFIX (IP Flow Information Export).

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 27 August 2026.

Copyright Notice

Table of Contents

1. Introduction

In network operation management, relying solely on traffic information from the third layer (IP layer) and fourth layer (transport layer) of the network is no longer sufficient to meet the needs of refined operation and intelligent decision-making. These underlying information can only reflect the routing, ports, and transmission status of data packets, but cannot reveal the business attributes, user behavior, and application intentions behind the traffic. The application layer information can accurately identify the business types and user behavior preferences corresponding to traffic, providing key basis for network planning, resource scheduling, and experience optimization.

This document solves the limitation of traditional IPFIX in only collecting network/transport layer data by defining the export of IPFIX application layer information. By standardizing the collection of core features such as HTTP/HTTPS at the application layer, deep mining of user behavior can be achieved, while meeting the requirements of network fine-grained resource scheduling and intelligent monitoring and management of traffic.

The information elements specified in this document can directly export corresponding data, which can directly or indirectly support operation and maintenance personnel to obtain various key operation and maintenance information; It should be noted that HTTP adopts a plaintext transmission mechanism, and relevant fields can be directly extracted and exported; HTTPS first completes encryption negotiation through TLS handshake, and then transmits HTTP messages. The encrypted content of HTTP messages may not have access to their related fields.

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. Sample Use Cases

3.1. CDN content introduction and traffic scheduling optimization

There is a large amount of HTTP/HTTPS traffic in the backbone network or metropolitan area network of the operator. Traditional IPFIX can only see the IP, port, and traffic size, and cannot identify the business type and content, making it difficult to accurately introduce content and schedule traffic. By exporting application layer information elements, business domain names, resource types, and access quality can be identified without decrypting HTTPS, achieving intelligent traffic scheduling and local content introduction, reducing cross network bandwidth costs. Network devices export relevant information based on IPfix, and network management platforms analyze and identify high traffic resources such as videos, images, and downloads. For high traffic and high access proportion businesses, they are automatically scheduled to local CDN nodes, reducing cross provincial/cross network traffic and improving user experience.

3.2. End to end monitoring and analysis of IPv6 networks

[I-D.pang-v6ops-ipv6-menitoring-deployment] proposes that in order to accurately identify bottlenecks and bottlenecks in IPv6 traffic, improve end-to-end connectivity and service quality of IPv6 networks, network management systems need to grasp the end-to-end IPv6 capability support, including not only network side information but also application layer information such as terminal types and applications. This document defines the export of relevant information elements.

4. New Information Elements

4.1. httpUserAgent

ElementID: 468

Name: httpUserAgent

Abstract Data Type: string

Data Type Semantics: identifier

Description：

The "User-Agent" header field contains information about the user agent originating the request, which is often used by servers to help identify the scope of reported interoperability problems, to work around or tailor responses to avoid particular user agent limitations, and for analytics regarding browser or operating system use. A user agent SHOULD send a User-Agent field in each request unless specifically configured not to do so.

Reference：See RFC 7231 section 5.5.3 for the specification of User-Agent

4.2. httpRequestHost

ElementID: 460

Name: httpRequestHost

Abstract Data Type: string

Data Type Semantics: identifier

Description：

The "Host" header field in a request provides the host and port information from the target URI, enabling the origin server to distinguish among resources while servicing requests for multiple host names on a single IP address.

Reference : See RFC 7230 section 5.4 for the specification of Host

4.3. httpRequestReferer

ElementID: TBD

Name: httpRequestReferer

Abstract Data Type: string

Data Type Semantics: identifier

Description :

The "Referer" [sic] header field allows the user agent to specify a URI reference for the resource from which the target URI was obtained.

Reference : See RFC 7231 section 5.5.2 for the specification of Referer

4.4. httpRequestAccept

ElementID: TBD

Name: httpRequestAccept

Abstract Data Type: string

Data Type Semantics: identifier

Description :

The "Accept" header field can be used by user agents to specify response media types that are acceptable. Accept header fields can be used to indicate that the request is specifically limited to a small set of desired types, as in the case of a request for an in-line image.

Reference : See RFC 7231 section 5.3.2 for the specification of Accept

4.5. httpStatusCode

ElementID: 457

Name: httpStatusCode

Abstract Data Type: unsigned16

Data Type Semantics: identifier

Description: The status-code element is a three-digit integer result code of the attempt to understand and satisfy the request. Status codes indicate whether a specific HTTP request has been successfully completed, requires further action, or has failed.

Reference: See RFC 7231 section 6 for the specification of status codes.

4.6. httpRequestMethod

ElementID: 459

Name: httpRequestMethod

Abstract Data Type: string

Data Type Semantics:identifier

Description: The request method indicates the desired action to be performed on the target resource. Request methods define the semantics of the request and the expected behavior of servers when handling the request.

Reference: See RFC 7231 section 4 for the specification of request methods.

4.7. httpRequestTarget

ElementID: 461

Name: httpRequestTarget

Abstract Data Type: string

Data Type Semantics:identifier

Description: The request-target identifies the target resource upon which to apply the request. It is derived from the request URI and determines which resource is being requested. Reference: See RFC 7230 section 5.3 for the specification of the request-target.

4.8. httpMessageVersion

ElementID: 462

Name: httpMessageVersion

Abstract Data Type: string

Data Type Semantics:identifier

Description: The HTTP-version indicates the version of the HTTP protocol used to send the message. It allows the sender and recipient to determine the protocol semantics to be applied.

Reference: See RFC 7230 section 2.6 for the specification of HTTP versioning.

4.9. httpContentType

ElementID: 469

Name: httpContentType

Abstract Data Type: string

Data Type Semantics: default

Description: The "Content-Type" header field indicates the media type of the associated representation. It allows the recipient to understand how to process the enclosed message body.

Reference: See RFC 7231 section 3.1.1.5 for the specification of Content-Type.

4.10. httpReasonPhrase

ElementID: 470

Name: httpReasonPhrase

Abstract Data Type: string

Data Type Semantics: default

Description:

The reason-phrase is intended to give a short textual description of the status code. It is meant for human consumption and does not affect the interpretation of the status code.

Reference: See RFC 7230 section 6.1 for the specification of the reason phrase.

4.11. httpResponseCacheControl

ElementID: TBD

Name: httpResponseCacheControl

Abstract Data Type: string

Data Type Semantics: identifier

Description： The "Cache-Control" header field is used to list directives for caches along the request/response chain. Cache directives are unidirectional, in that the presence of a directive in a request does not imply that the same directive is present or copied in the response.

Reference：See RFC 9111 section 5.2 for the specification of Cache-Control

4.12. httpResponseETag

ElementID: TBD

Name: httpResponseETag

Abstract Data Type: string

Data Type Semantics: identifier

Description：

The "ETag" header field in a response provides the current entity-tag for the selected representation, as determined at the conclusion of handling the request. An entity-tag is an opaque validator for differentiating between multiple representations of the same resource, regardless of whether those multiple representations are due to resource state changes over time, content negotiation resulting in multiple representations being valid at the same time,or both. An entity-tag consists of an opaque quoted string, possibly prefixed by a weakness indicator.

Reference：See RFC 7232 section2.3 for the specification of Etag

4.13. httpResponseServer

ElementID: TBD

Name: httpResponseServer

Abstract Data Type: string

Data Type Semantics: identifier

Description：

The "Server" header field contains information about the software used by the origin server to handle the request, which is often used by clients to help identify the scope of reported interoperability problems, to work around or tailor requests to avoid particular server limitations, and for analytics regarding server or operating system use. An origin server MAY generate a Server field in its responses.

Reference：See RFC 7231 section7.4.2 for the specification of Server.

4.14. tlsSNI

ElementID: TBD

Name: tlsSNI

Abstract Data Type: string

Data Type Semantics: identifier

Description：

The value of the Server Name Indication (SNI) extension in the TLS ClientHello message, which specifies the target server hostname (encoded in UTF - 8) as defined in Section 3 of RFC 6066 and used for certificate selection in multi - domain hosting scenarios.The Server Name Indication (SNI) extension enables a TLS client to indicate the name of the server it is attempting to connect to.This mechanism facilitates secure connections to servers hosting multiple virtual servers on a single network address.

Reference : See RFC 6066 section 3 for the specification of Server Name Indication.

4.15. tlsVersion

ElementID: TBD

Name: tlsVersion

Abstract Data Type: unsigned16

Data Type Semantics: identifier

Description :

The TLS protocol version number negotiated during the handshake and used in the TLS record layer header, encoded as a 2-byte unsigned integer (e.g., 0x0304 for TLS 1.3) as defined in RFC 8446 and RFC 5246.

Reference : RFC 8446,RFC 5246

4.16. tlsClientHelloCipherSuite

ElementID: TBD

Name: tlsClientHelloCipherSuite

Abstract Data Type: unsigned16

Data Type Semantics: identifier

Description :

A list of cipher suite values (each 2-byte unsigned integer) advertised by the client in the TLS ClientHello message.

Reference : RFC 8446

4.17. tlsClientHelloSSLExtension

ElementID: TBD

Name: tlsClientHelloSSLExtension

Abstract Data Type: unsigned16

Data Type Semantics: identifier

Description： A list of TLS extension type codes (each 2-byte unsigned integer) from the ClientHello extensions field.

Reference：RFC 8446

4.18. tlsClientHelloEllipticCurve

ElementID: TBD

Name:tlsClientHelloEllipticCurve

Abstract Data Type: unsigned16

Data Type Semantics: identifier

Description： A list of elliptic curve identifiers (each 2-byte unsigned integer) from the "elliptic_curves" extension in ClientHello.

Reference：RFC 8446

4.19. tlsClientHelloEllipticCurvePointFormat

ElementID: TBD

Name: tlsClientHelloEllipticCurvePointFormat

Abstract Data Type: unsigned16

Data Type Semantics: identifier

Description： A list of elliptic curve point format identifiers (each 1-byte unsigned integer) from the "elliptic_curves_point_formats" extension in ClientHello.

Reference：RFC 8446

5. Security Considerations

TBD

6. IANA Considerations

The document makes a request to IANA to register the Information Elements defined in section 4.

7. Informative References

[I-D.pang-v6ops-ipv6-monitoring-deployment]
Pang, R., Zhao, J., Jin, M., and S. Zhang, "IPv6 Network Deployment Monitoring and Analysis", Work in Progress, Internet-Draft, draft-pang-v6ops-ipv6-monitoring-deployment-04, 18 November 2025, <https://datatracker.ietf.org/doc/html/draft-pang-v6ops-ipv6-monitoring-deployment-04>.

Authors' Addresses

Xing Gao (editor)
China Unicom
Beijing
China
Email: gaox60@chinaunicom.cn

Shuai Zhang
China Unicom
Beijing
China
Email: zhangs366@chinaunicom.cn

Changwang Lin
New H3C Technologies
Beijing
China
Email: linchangwang.04414@h3c.com