Équipe d'ingénierie de l'Internet (IETF)
Request for Comments : 7011
STD : 77
RFC rendue obsolète : 5101
Catégorie : Sur la voie de la normalisation
ISSN: 2070-1721

B. Claise, éd., Cisco Systems, Inc.
B. Trammell, éd., ETH Zurich
P. Aitken, Cisco Systems, Inc.
septembre 2013

Traduction Claude Brière de L'Isle

Spécification du protocole d'exportation d'informations de flux IP (IPFIX) pour l'échange d'informations de flux

Résumé
Le présent document spécifie le protocole d'exportation d'informations de flux IP (IPFIX, *IP Flow Information Export*) qui sert de moyen pour transmettre les informations de flux de trafic sur le réseau. Afin de transmettre les informations de flux de trafic d'un processus d'exportation à un processus de collecte, une représentation commune des flux de données et un moyen standard de les communiquer sont nécessaires. Le présent document décrit comment les données IPFIX et les enregistrements de gabarit sont portés sur un certain nombre de protocoles de transport d'un processus d'exportation IPFIX à un processus de collecte IPFIX. Le présent document rend obsolète la RFC 5101.

Statut de ce mémoire
Ceci est un document de l'Internet sur la voie de la normalisation.

Le présent document a été produit par l'équipe d'ingénierie de l'Internet (IETF). Il représente le consensus de la communauté de l'IETF. Il a subi une révision publique et sa publication a été approuvée par le groupe de pilotage de l'ingénierie de l'Internet (IESG). Tous les documents approuvés par l'IESG ne sont pas candidats à devenir une norme de l'Internet ; voir la Section 2 de la RFC5741.

Les informations sur le statut actuel du présent document, tout errata, et comment fournir des réactions sur lui peuvent être obtenues à http://www.rfc-editor.org/info/rfc7011

Table des matières

1. Introduction

Le trafic sur un réseau de données peut être vu comme consistant en des flux passant à travers des éléments de réseau. Pour des besoins administratifs ou autres, il est souvent intéressant, utile, ou même nécessaire d'avoir accès aux informations sur ces flux qui passent à travers les éléments de réseau. Un processus de collecte devrait être capable de recevoir les informations de flux passant à travers plusieurs éléments de réseau au sein du réseau de données. Cela exige l'uniformité de la méthode de représentation des informations de flux et des moyens de communiquer les flux des éléments de réseau au point de collecte. Le présent document spécifie un protocole pour satifaire ces exigences. Le présent document spécifie en détails la représentation des différents flux, ainsi que des données supplémentaires requises pour l'interprétation des flux, du format de paquet, des mécanismes de transport utilisés, des problèmes de sécurité, etc.

1.1 Changements par rapport à la RFC 5101

Le présent document rend obsolète la révision de la proposition de norme de la spécification de protocole IPFIX [RFC5101]. Le protocole spécifié par le présent document est interopérable avec le protocole spécifié dans la [RFC5101]. Les changements suivants ont été faits par ce document par rapport au précédent document :

- Tous les errata techniques et rédactionnels en cours sur la [RFC5101] ont été incorporés.

- Comme le registre [IANA-IPFIX] est maintenant la référence normative pour toutes les définitions d'élément d'information (voir la [RFC7012]) toutes les définitions des éléments d'information de la Section 4 ont été remplacés par des références à ce registre.

- Le codage des types de données dateTimeSeconds, dateTimeMilliseconds, dateTimeMicroseconds, et dateTimeNanoseconds, et le codage qui s'y rapporte du champ Heure d'exportation de l'en-tête de message IPFIX, a été précisé, en particulier par rapport à l'époque sur laquelle les types de données d'horodatage se fondent.

- Un nouveau paragraphe 5.2 a été ajouté pour traiter le retour à zéro de ces types de données d'horodatage après qu'elles débordent les années 2032-2038.

- Des précisions sur le codage, en particulier à la Section 6, ont été faites à toutes les valeurs IPFIX qui sont codées dans l'ordre des octets du réseau.

- La gestion de gabarit, décrite à la Section 8, a été simplifiée et précisée : la spécification a été assouplie par rapport à la [RFC5101], en particulier concernant les défaillances potentielles dans la réutilisation d'identifiant de gabarit. Des cas particuliers supplémentaires de gestion de gabarit ont été traités. Le nouveau langage de gestion de gabarit est interopérable avec celui de la [RFC5101] dans la mesure où le comportement était défini dans la spécification précédente.

- Le paragraphe 11.3 (Authentification mutuelle) a été amélioré pour se référer aux pratiques courantes dans l'authentification mutuelle de la sécurité de la couche transport (TLS, *Transport Layer Security*) ; les références à Punycode ont été supprimées, car elles sont couvertes dans la [RFC6125].

- Des amélioration rédactionnelles, incluant des changements structurels aux Sections 8, 9, et 10 pour améliorer la lisibilité, ont été appliquées. Le comportement commun à tous les protocoles de transport a été séparé, à l'exception de transports spécifiquement notés. Tout le langage de gestion de gabarit (sur les deux processus d'exportation et de collecte) a été unifié à la Section 8.

- Une nouvelle Section 12 sur les considérations de gestion a été ajoutée.

1.2 Vue d'ensemble des documens IPFIX

Le protocole IPFIX fournit aux administrateurs de réseau l'accès aux informations de flux IP. L'architecture pour l'exportation des informations de flux IP mesurées à partir d'un processus d'exportation IPFIX à un processus de collecte est définie dans la [RFC5470], selon les exigences définies dans la [RFC3917]. Le présent document spécifie comment les enregistrements de données et les gabarits IPFIX sont portés, via un certain nombre de protocoles de transport, du processus d'exportation IPFIX au processus de collecte IPFIX.

Quatre optimisations/extensions IPFIX sont actuellement spécifiées : une méthode pour économiser la bande passante pour le protocole IPFIX [RFC5473], une méthode efficace pour exporter les flux bidirectionnels [RFC5103], une méthode pour la définition et l'exportation de structures de données complexes [RFC6313], et la spécification du protocole sur les médiateurs IPFIX [RFC7119] fondée sur le cadre de médiation IPFIX [RFC6183].

Un "transport fondé sur le fichier" pour IPFIX, qui définit comment les messages IPFIX peuvent être mémorisés dans des fichiers pour les flux de travail fondés sur le document et pour les besoins d'archivage, est discuté dans la [RFC5655].

IPFIX a une description formelle des éléments d'information IPFIX -- leur nom, types de données, et informations de sémantique supplémentaires -- comme spécifié dans la [RFC7012]. Le registre est tenu par l'IANA [IANA-IPFIX]. L'exportation en ligne des informations de type d'élément d'information est spécifiée dans la [RFC5610].

Le cadre du choix de paquet et de rapport [RFC5474] permet aux éléments de réseau de choisir des sous ensembles de paquets par des méthodes statistiques et autres, et d'exporter un flux de rapports sur les paquets choisis à un collecteur. L'ensemble de techniques de choix de paquet (échantillonnage, filtrage, et hachage) normalisées par le protocole d'échantillonnage de paquet (PSAMP, *Packet Sampling Protocol*) est décrit dans la [RFC5475]. Le protocole PSAMP [RFC5476], qui utilise IPFIX comme protocole d'exportation, spécifie l'exportation des informations de paquets d'un processus d'exportation PSAMP à un collecteur PSAMP. Au lieu d'exporter des rapports de paquets PSAMP, le flux de paquets choisi peut aussi servir d'entrée à la génération d'enregistrements de flux IPFIX. Comme IPFIX, PSAMP a une description formelle de ses éléments d'information : leur noms, type, et des informations de sémantique supplémentaires. Le modèle d'information PSAMP est défini dans la [RFC5477].

La [RFC6615] spécifie un module de MIB pour la surveillance, et la [RFC6728] spécifie un modèle de données pour configurer et surveiller les appareils conformes à IPFIX et PSAMP en utilisant le protocole de configuration de réseau (NETCONF, *Network Configuration Protocol*). La [RFC6727] spécifie le module de MIB PSAMP comme une extension du module de MIB IPFIX SELECTOR défini dans la [RFC6615].

En termes de développement, la [RFC5153] fournit des lignes directrices pour la mise en œuvre et l'utilisation du protocole IPFIX, tandis que la [RFC5471] fournit des lignes directrices pour les essais. Finalement, la [RFC5472] décrit quels types d'applications peuvent utiliser le protocole IPFIX et comment elles peuvent utiliser les informations fournies. Elle montre de plus comment le cadre IPFIX se rapporte aux autres architectures et cadres.

2. Terminologie

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

Les définitions des termes de base comme flux de trafic, processus d'exportation, processus de collecte, points d'observation, etc. sont sémantiquement identiques à celles qui se trouvent dans le document sur les exigences de IPFIX [RFC3917]. Certains des termes ont été étendus pour plus de clarté dans la définition du protocole. Des termes supplémentaires nécessaires pour le protocole ont aussi été définis. Les définitions de ce document et de la [RFC5470] sont équivalents ; les définitions qui ne sont pertinentes que pour le protocole IPFIX apparaissent seulement ici.

Le Tableau de résumé de la terminologie du paragraphe 2.1 donne un rapide survol des relations entre certains des différents termes définis.

Point d'observation : un point d'observation est une localisation dans le réseau où les paquets peuvent être observés. Des exemples incluent une ligne à laquelle une sonde est rattachée, un support partagé, comme un LAN fondé sur Ethernet, un seul accès d'un routeur, ou un ensemble d'interfaces (physiques ou logiques) d'un routeur.

Noter que tout point d'observation est associé à un domaine d'observation (défini ci-dessous) et que un point d'observation peut être un sur-ensemble de plusieurs autres points d'observation. Par exemple, un point d'observation peut être une carte de ligne entière. Cela serait le sur-ensemble des points d'observation individuels aux interfaces de la carte de ligne.

Domaine d'observation : un domaine d'observation est le plus grand ensemble de points d'observation pour lequel des flux d'informations peuvent être agrégés par un processus de mesures. Par exemple, une carte de ligne de routeur peut être un domaine d'observation si elle est composée de plusieurs interfaces, dont chacune est un point d'observation. Dans le message IPFIX qu'il génère, le domaine d'observation inclut son identifiant de domaine d'observation, qui est unique par processus d'exportation. De cette façon, le processus de collecte peut identifier le domaine d'observation spécifique à partir de l'exportateur qui envoie les messages IPFIX. Chaque point d'observation est associé à un domaine d'observation. Il est RECOMMANDÉ que les identifiants de domaine d'observation soient aussi être uniques par appareil IPFIX.

Traitement de paquet : "traitement de paquet" se réfère à la ou aux actions effectuées sur un paquet par un appareil émetteur ou autre boîtier de médiation, incluant la transmission, l'élimination, le retard pour les besoins du formatage du trafic, etc.

Flux de trafic ou flux : plusieurs définitions du terme de "flux" sont utilisées dans la communauté de l'Internet. Dans le contexte de IPFIX, on utilise la définition suivante : un flux est défini comme un ensemble de paquets ou trames passant à un point d'observation dans le réseau durant un certain intervalle de temps. Tous les paquets qui appartiennent à un flux particulier ont un ensemble commun de propriétés. Chaque propriété est définie comme le résultat de l'application d'une fonction aux valeurs de :

1. un ou plusieurs champs d'en-tête de paquet (par exemple, adresse de destination IP) d'en-tête de transport (par exemple, numéro d'accès de destination) ou d'en-tête d'application (par exemple, champs d'en-tête RTP [RFC3550]) ;
2. une ou plusieurs caractéristiques du paquet lui-même (par exemple, nombre d'étiquettes MPLS, etc.) ;
3. un ou plusieurs des champs déduits du traitement de paquet (par exemple, adresse du prochain bond IP, interface de sortie, etc.).

Un paquet est défini comme appartenant à un flux si il satisfait complètement à toutes les propriétés définies du flux.

Noter que l'ensemble de paquets représenté par un flux peut être vide ; c'est-à-dire, un flux peut représenter zéro, un ou plusieurs paquets. Comme l'échantillonnage est un traitement de paquet, cette définition inclut les paquets choisis par un mécanisme d'échantillonnage.

Clé de flux : chacun des champs qui :

1. appartient à l'en-tête de paquet (par exemple, adresse de destination IP) ou
2. est une propriété du paquet lui-même (par exemple, longueur de paquet) ou
3. est déduit du traitement de paquet (par exemple, numéro de système autonome (AS))

et qui est utilisé pour définir un flux (c'est-à-dire, est une propriété commune à tous les paquets du flux) est appelé une clé de flux. Par exemple, la clé de flux traditionnelle "quintuplet" d'adresses IP de source et destination, accès de transport de source et destination, et protocole de transport, groupe tous les paquets qui appartiennent à une seule direction de communication sur une seule prise.

Enregistrement de flux : un enregistrement de flux contient des informations sur un flux spécifique qui a été observé à un point d'observation. Un enregistrement de flux contient les propriétés mesurées du flux (par exemple, le nombre total d'octets pour tous les paquets du flux) et contient généralement les propriétés caractéristiques du flux (par exemple, l'adresse IP de source).

Processus de mesures : le processus de mesures génère des enregistrements de flux. Les entrées au processus sont les en-têtes de paquet, les caractéristiques, et le traitement de paquet observé à un ou plusieurs points d'observation. Le processus de mesures consiste en un ensemble de fonctions qui incluent la capture d'en-tête de paquet, l'horodatage, l'échantillonnage, la classification, et le maintien des enregistrements de flux. La maintenance des enregistrements de flux peut inclure de créer de nouveaux enregistrements, de mettre à jour des enregistrements existants, de calculer des statistiques de flux, de déduire d'autres propriétés de flux, de détecter l'expiration du flux, de passer les enregistrements de flux au processus d'exportation, et de supprimer les enregistrements de flux.

Processus d'exportation : le processus d'exportation envoie les messages IPFIX à un ou plusieurs processus de collecte. Les enregistrements de flux dans les messages sont générés par un ou plusieurs processus de mesure.

Exportateur : l'appareil qui héberge un ou plusieurs processus d'exportation est appelé un exportateur.

Appareil IPFIX : un appareil IPFIX héberge au moins un processus d'exportation. Il peut héberger d'autres processus d'exportation ainsi qu'un nombre arbitraire de points d'observation et processus de mesure.

Processus de collecte : un processus de collecte reçoit des messages IPFIX provenant d'un ou plusieurs processus d'exportation. Le processus de collecte peut traiter ou mémoriser les enregistrements de flux reçus dans ces messages, mais ces actions sortent du domaine d'application du présent document.

Collecteur : l'appareil qui héberge un ou plusieurs processus de collecte est appelé un collecteur.

Gabarit *(template)* : un gabarit est une séquence ordonnée de paires <type, longueur> utilisée pour spécifier complètement la structure et la sémantique d'un ensemble particulier d'informations qui doivent être communiquées d'un appareil IPFIX à un collecteur. Chaque gabarit est identifiable au moyen d'un identifiant de gabarit.

Message IPFIX : c'est un message qui a son origine au processus d'exportation et porte les enregistrements IPFIX d'un processus d'exportation, et dont la destination est un processus de collecte. Un message IPFIX est encapsulé à la couche transport.

En-tête de message : l'en-tête de message est la première partie d'un message IPFIX ; l'en-tête de message donne les informations de base sur le message, comme la version IPFIX, la longueur du message, le numéro de séquence du message, etc.

Enregistrement de gabarit : il définit la structure et l'interprétation des champs dans un enregistrement de données.

Enregistrement de données : c'est un enregistrement qui contient les valeurs des paramètres correspondants à un enregistrement de gabarit.

Enregistrement de gabarit d'options : c'est un enregistrement de gabarit qui définit la structure et l'interprétation des champs dans un enregistrement de données, incluant de définir la portée de l'enregistrement de données.

Ensemble : un ensemble est une collection d'enregistrements qui ont une structure similaire, précédée d'un en-tête. Dans un message IPFIX, zéro, un ou plusieurs ensembles suivent l'en-tête de message. Il y a trois différents types d'ensembles : ensembles de gabarits, ensembles de gabarits d'options, et ensembles de données.

Ensemble de gabarits : c'est une collection d'un ou plusieurs enregistrements de gabarit qui ont été groupés dans un message IPFIX.

Ensemble de gabarits d'options : c'est une collection d'un ou plusieurs enregistrements de gabarit d'options qui ont été groupés dans un message IPFIX.

Ensemble de données : c'est un ou plusieurs enregistrements de données, de même type, qui sont groupés dans un message IPFIX. Chaque enregistrement de données est précédemment défini par un enregistrement de gabarit ou un enregistrement de gabarit d'options.

Élément d'information : un élément d'information est une description indépendante du protocole et du codage d'un attribut qui peut apparaître dans un enregistrement IPFIX. Les éléments d'information sont définis dans le registre de l'IANA "Éléments d'information IPFIX" [IANA-IPFIX]. Le type associé à un élément d'information indique les contraintes sur ce qu'il peut contenir et aussi détermine les mécanismes de codage valides pour l'utilisation dans IPFIX.

Session de transport : dans le protocole de transmission de commandes de flux (SCTP, *Stream Control Transmission Protocol*) la session de transport est appelée l'association SCTP, qui est identifiée de façon univoque par les points d'extrémité SCTP [RFC4960] ; dans TCP, la session de transport est appelée la connexion TCP, qui est identifiée de façon univoque par la combinaison des adresses IP et des accès TCP utilisés. Dans UDP, la session de transport est appelée la session UDP, qui est identifiée de façon univoque par la combinaison des adresses IP et des accès UDP utilisés.

2.1 Tableau résumé de la terminologie

La Figure A montre un résumé de la terminologie IPFIX.

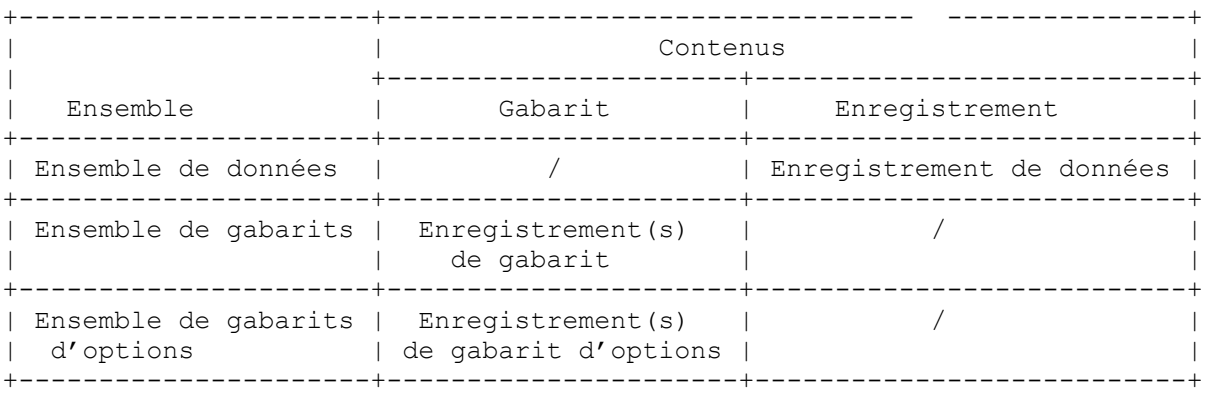


Figure A : Tableau résumé de la terminologie

Un ensemble de données est composé d'enregistrements de données. Aucun enregistrement de gabarit n'est inclus. Un enregistrement de gabarit ou un enregistrement de gabarit d'options définit l'enregistrement de données.

Un ensemble de gabarits contient seulement un ou des enregistrements de gabarit.

Un ensemble de gabarits d'options contient seulement un ou des enregistrements de gabarit d'options.

3. Format de message IPFIX

Un message IPFIX consiste en un en-tête de message, suivi par zéro, un ou plusieurs ensembles. Les ensembles peuvent être d'un des trois types possibles : ensemble de données, ensemble de gabarits, ou ensemble de gabarits d'options.

Le format du message IPFIX est montré à la Figure B.

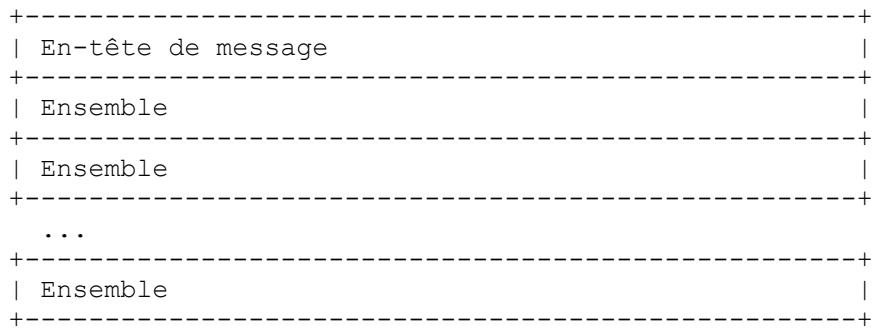


Figure B : Format de message IPFIX

Voici quelques exemples de messages IPFIX :

1. Un message IPFIX consistant en un entrelacement d'ensembles de gabarits, de données, et de gabarits d'options, comme le montre la Figure C. Ici, les ensembles de gabarit et de gabarits d'options sont transmis "à la demande", avant le premier ensemble de données dont ils définissent la structure.

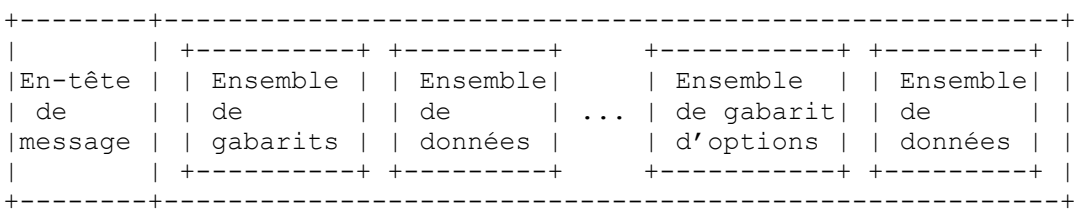


Figure C : Message IPFIX, exemple 1

2. Un message IPFIX consistant entièrement en ensembles de données, envoyé après que les enregistrements de gabarit appropriés ont été définis et transmis au processus de collecte, comme le montre la Figure D.

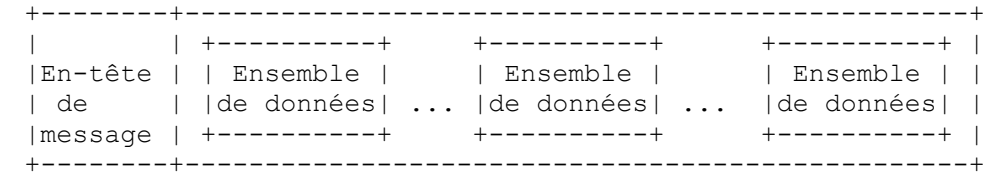


Figure D : Message IPFIX, exemple 2

3. Un message IPFIX consistant entièrement en ensembles de gabarits et de gabarits d'options, comme le montre la Figure E. Un tel message peut être utilisé pour définir ou redéfinir en bloc des gabarits et gabarits d'options.

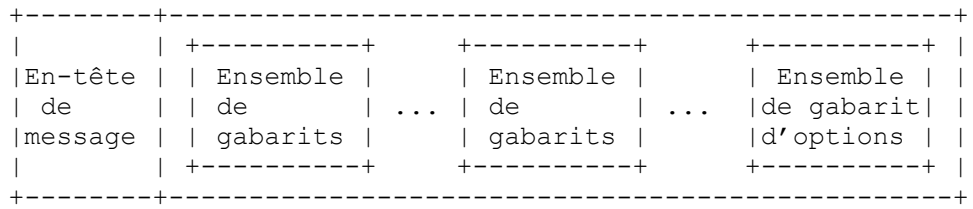


Figure E : Message IPFIX, exemple 3

3.1 Format d'en-tête de message

Le format de l'en-tête de message IPFIX est montré à la Figure F.

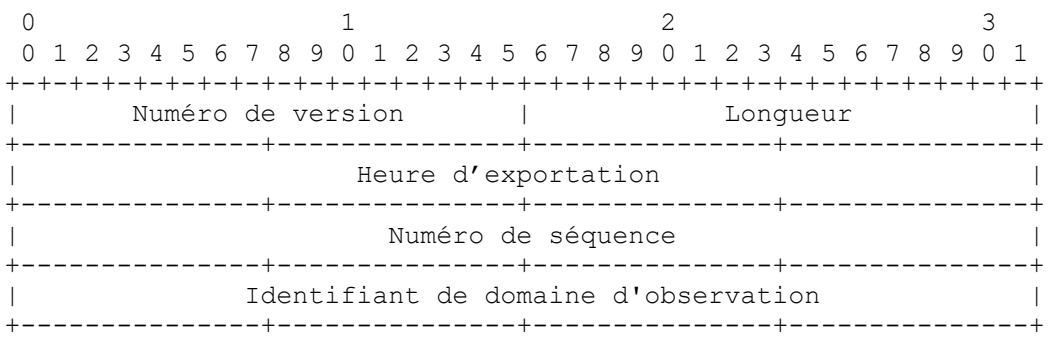


Figure F : Format d'en-tête de message IPFIX

Chaque champ d'en-tête de message est exporté dans l'ordre des octets du réseau. Les champs sont définis comme suit :

Version : version de IPFIX à laquelle se conforme ce message. La valeur de ce champ est 0x000a pour la version actuelle, incrémentant de un la version utilisée dans la version 9 des services export NetFlow [RFC3954].

Longueur : longueur totale du message IPFIX, mesurée en octets, incluant l'en-tête de message et le ou les ensembles.

Heure d'exportation : heure à laquelle l'en-tête de message IPFIX quitte l'exportateur, exprimée en secondes depuis l'époque UNIX du 1er janvier 1970 à 00:00 UTC, codée comme un entier non signé de 32 bits.

Numéro de séquence : compteur incrémentaire de séquence modulo 2^32 de tous les enregistrements de données IPFIX envoyés dans le flux en cours depuis le domaine d'observation actuel par le processus d'exportation. Chaque flux SCTP compte séparément les numéros de séquence, tandis que tous les messages dans une connexion TCP ou session UDP sont considérés faire partie du même flux. Cette valeur peut être utilisée par le processus de collecte pour identifier si des enregistrements de données IPFIX ont été manqués. Les gabarits et enregistrements de gabarit d'options n'augmentent pas le numéro de séquence.

Identifiant de domaine d'observation : identifiant de 32 bits du domaine d'observation, localement unique pour le processus d'exportation. Le processus d'exportation utilise l'identifiant de domaine d'observation pour identifier de façon univoque le processus de collecte au domaine d'observation qui a mesuré les flux. Il est RECOMMANDÉ que cet identifiant soit aussi unique par appareil IPFIX. Les processus de collecte DEVRAIENT utiliser le champ Identifiant de session de transport et Identifiant de domaine d'observation pour séparer les différents flux exportés originaires du même exportateur. L'identifiant de domaine d'observation DEVRAIT être 0 quand aucun identifiant spécifique de domaine d'observation n'est pertinent pour le message IPFIX entier, par exemple, quand on exporte les statistiques du processus d'exportation, ou dans le cas d'une hiérarchie de collecteurs quand des enregistrements de données agrégés sont exportés.

3.2 Format de spécificateur de champ

Les fabricants doivent être capables de définir des éléments d'information propriétaires, parce que, par exemple, ils livrent un produit non standardisé, ou que l'élément d'information est d'une certaine façon commercialement sensible. Ce paragraphe décrit le format Spécificateur de champ pour les éléments d'information enregistrés par l'IANA [IANA-IPFIX] et les éléments d'information spécifiques de l'entreprise.

Les éléments d'information sont identifiés par l'identifiant d'élément d'information. Quand le bit Entreprise est réglé à 0, l'élément d'information correspondant apparaît dans [IANA-IPFIX], et le numéro d'entreprise NE DOIT PAS être présent. Quand le bit Entreprise est établi à 1, l'identifiant d'élément d'information correspondant identifié dans l'élément d'information spécifique de l'entreprise, le numéro d'entreprise DOIT être présent. Un exemple est montré à l'Appendice A.2.2.

Le format Spécificateur de champ est montré à la Figure G.

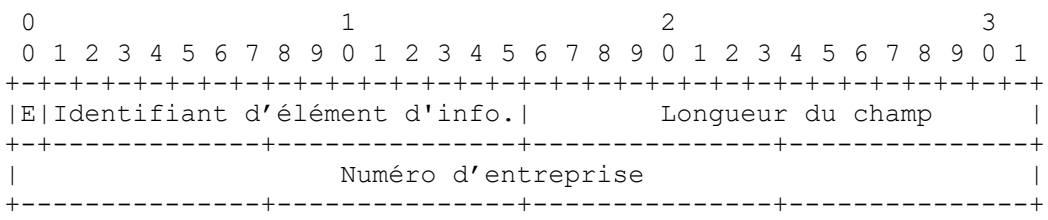


Figure G : Format de spécificateur de champ

Où :

E (bit Entreprise) : c'est le premier bit du spécificateur de champ. Si ce bit est zéro, l'identifiant d'élément d'information identifie un élément d'information dans [IANA-IPFIX], et le champ de quatre octets Numéro d'entreprise NE DOIT PAS être présent. Si ce bit est un, l'identifiant d'élément d'information identifie un élément d'information spécifique de l'entreprise, et le champ Numéro d'entreprise DOIT être présent.

Identifiant d'élément d'information : valeur numérique qui représente l'élément d'information. Voir [IANA-IPFIX].

Longueur de champ : longueur de l'élément d'information correspondant codé, en octets. Voir [IANA-IPFIX]. La longueur de champ peut être plus petite que celle donnée dans [IANA-IPFIX] si le codage de taille réduite est utilisé (voir le paragraphe 6.2). La valeur 65535 est réservée pour les éléments d'information de taille variable (voir la Section 7).

Numéro d'entreprise : numéro d'entreprise IANA [IANA-PEN] de l'autorité qui définit l'identifiant d'élément d'information dans cet enregistrement de gabarit.

3.3 Format Ensemble et en-tête d'ensemble

Un ensemble est un terme générique pour une collection d'enregistrements qui ont une structure similaire. Il y a trois différent types d'ensembles : ensembles de gabarits, ensembles de gabarits d'options, et ensembles de données. Chacun de ces ensembles consiste en un en-tête d'ensemble et en un ou plusieurs enregistrements. Le format d'ensemble et le format d'en-tête d'ensemble sont définis dans les paragraphes qui suivent.

3.3.1 Format d'ensemble

Un ensemble a le format montré à la Figure H. Les types d'enregistrements peuvent être des enregistrements de gabarit, des enregistrements de gabarit d'options, ou des enregistrements de données. Les types d'enregistrements NE DOIVENT PAS être mélés dans un ensemble.

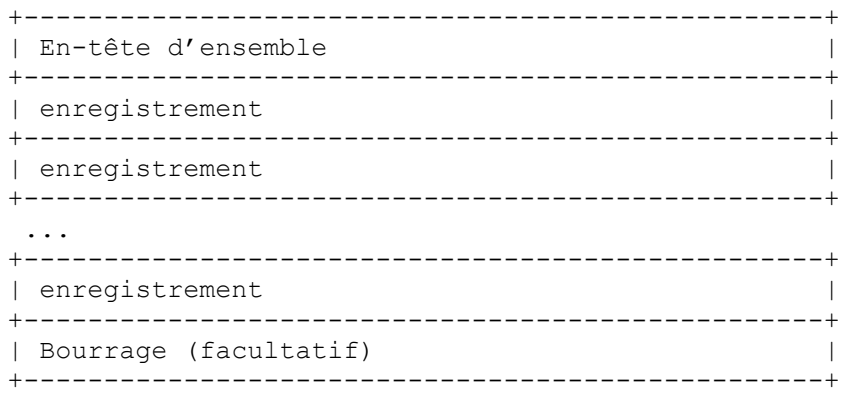


Figure H : Format d'ensemble

En-tête d'ensemble : le format d'en-tête d'ensemble est défini au paragraphe 3.3.2.

Enregistrement : un des formats d'enregistrement : enregistrement de gabarit, enregistrement de gabarit d'options, ou enregistrement de données.

Bourrage : le processus d'exportation PEUT insérer des octets de bourrage, afin que l'ensemble suivant commence à une limite alignée. Pour des raisons de sécurité, les octets de bourrage DOIVENT être composés d'octets de valeur zéro (0). La longueur de bourrage DOIT être plus courte que celle de tout enregistrement admissible dans cet ensemble. Si le bourrage du message IPFIX est désiré en combinaison avec de très courts enregistrements, alors l'élément d'information de bourrage "paddingOctets" peut être utilisé pour des enregistrements de bourrage tels que leur longueur soit augmentée à plusieurs de 4 ou 8 octets. Parce que les ensembles de gabarit sont toujours alignés sur 4 octets par définition, le bourrage n'est nécessaire que dans le cas d'autres alignements, par exemple, sur des limites de 8 octets.

3.3.2 Format d'en-tête d'ensemble

Chaque ensemble contient un en-tête commun. Cet en-tête est défini à la Figure I.

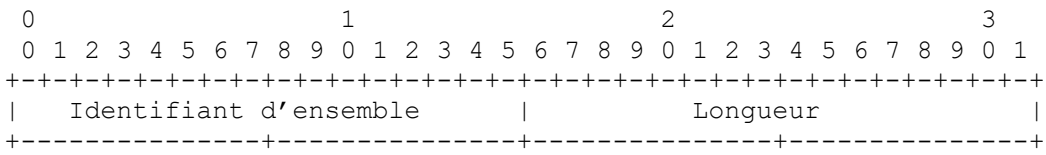


Figure I : Format d'en-tête d'ensemble

Chaque champ d'en-tête d'ensemble est exporté "dans le format du réseau". Les champs sont définis comme suit :

Identifiant d'ensemble : identifie l'ensemble. Une valeur de 2 est réservée pour les ensembles de gabarits. Une valeur de 3 est réservée pour les ensembles de gabarits d'options. Les valeurs de 4 à 255 sont réservées pour une utilisation future. Les valeurs 256 et au-dessus sont utilisées pour les ensembles de données. Les valeurs d'identifiant d'ensemble de 0 et 1 ne sont pas utilisées, pour des raisons historiques [RFC3954].

Longueur : longueur totale de l'ensemble, en octets, incluant l'en-tête d'ensemble, tous les enregistrements, et le bourrage facultatif. Parce que un ensemble individuel PEUT contenir plusieurs enregistrements, la valeur de Longueur DOIT être utilisée pour déterminer la position du prochain ensemble.

3.4 Format Enregistrement

IPFIX définit trois formats d'enregistrement, comme définis dans les paragraphes qui suivent : le format d'enregistrement de gabarit, le format d'enregistrement de gabarit d'options et le format d'enregistrement de données.

3.4.1 Format d'enregistrement de gabarit

Un des éléments essentiels du format d'enregistrement IPFIX est l'enregistrement de gabarit. Les gabarits améliorent beaucoup la souplesse du format d'enregistrement parce qu'ils permettent au processus de collecte de traiter les messages IPFIX sans nécessairement connaître l'interprétation de tous les enregistrements de données. Un enregistrement de gabarit contient toutes les combinaisons d'identifiants d'élément d'information alloués par l'IANA et/ou spécifiques d'entreprise.

Le format de l'enregistrement de gabarit est montré à la Figure J. Il consiste en un en-tête d'enregistrement de gabarit et un ou plusieurs spécificateurs de champs. Les spécificateurs de champs sont définis à la Figure G ci-dessus.

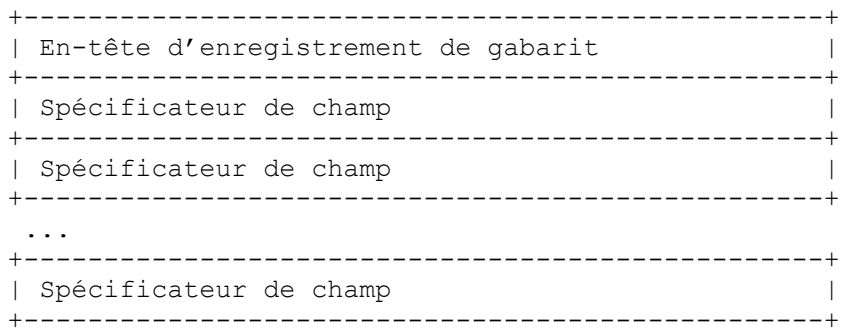


Figure J : Format d'enregistrement de gabarit

Le format de l'en-tête d'enregistrement de gabarit est montré à la Figure K.

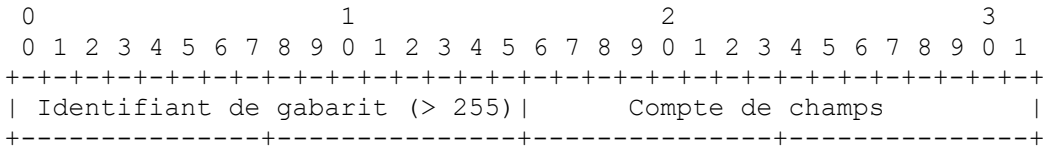


Figure K : Format d'en-tête d'enregistrement de gabarit

Les définitions du champ En-tête d'enregistrement de gabarit sont comme suit :

Identifiant de gabarit : chaque enregistrement de gabarit reçoit un identifiant de gabarit unique dans la gamme 256 à 65535. Cette unicité est locale pour la session de transport et le domaine d'observation qui génèrent l'identifiant de gabarit. Comme les identifiants de gabarit sont utilisés comme identifiant d'ensemble dans les ensembles qu'ils décrivent (voir le paragraphe 3.4.3) les valeurs 0 à 255 sont réservées pour des types d'ensembles particuliers (par exemple, les ensembles de gabarits eux-mêmes) et les gabarits et gabarits d'options (voir le paragraphe 3.4.2) ne peuvent pas partager des identifiants de gabarit au sein d'une session de transport et d'un domaine d'observation. Il n'y a pas de contrainte sur l'ordre d'allocation des identifiants de gabarit. Comme les processus d'exportation sont libres d'allouer les identifiants de gabarit comme ils le jugent bon, les processus de collecte NE DOIVENT PAS supposer des identifiants de gabarit incrémentaires, ou quelque chose sur le contenu d'un gabarit sur la base de son seul identifiant de gabarit.

Compte de champs : nombre de champs dans cet enregistrement de gabarit.

L'exemple de la Figure L montre un ensemble de gabarits avec un mélange d'éléments d'information alloués par l'IANA et spécifiques d'entreprise. Il consiste en un en-tête d'ensemble, un en-tête de gabarit, et plusieurs spécificateurs de champs.

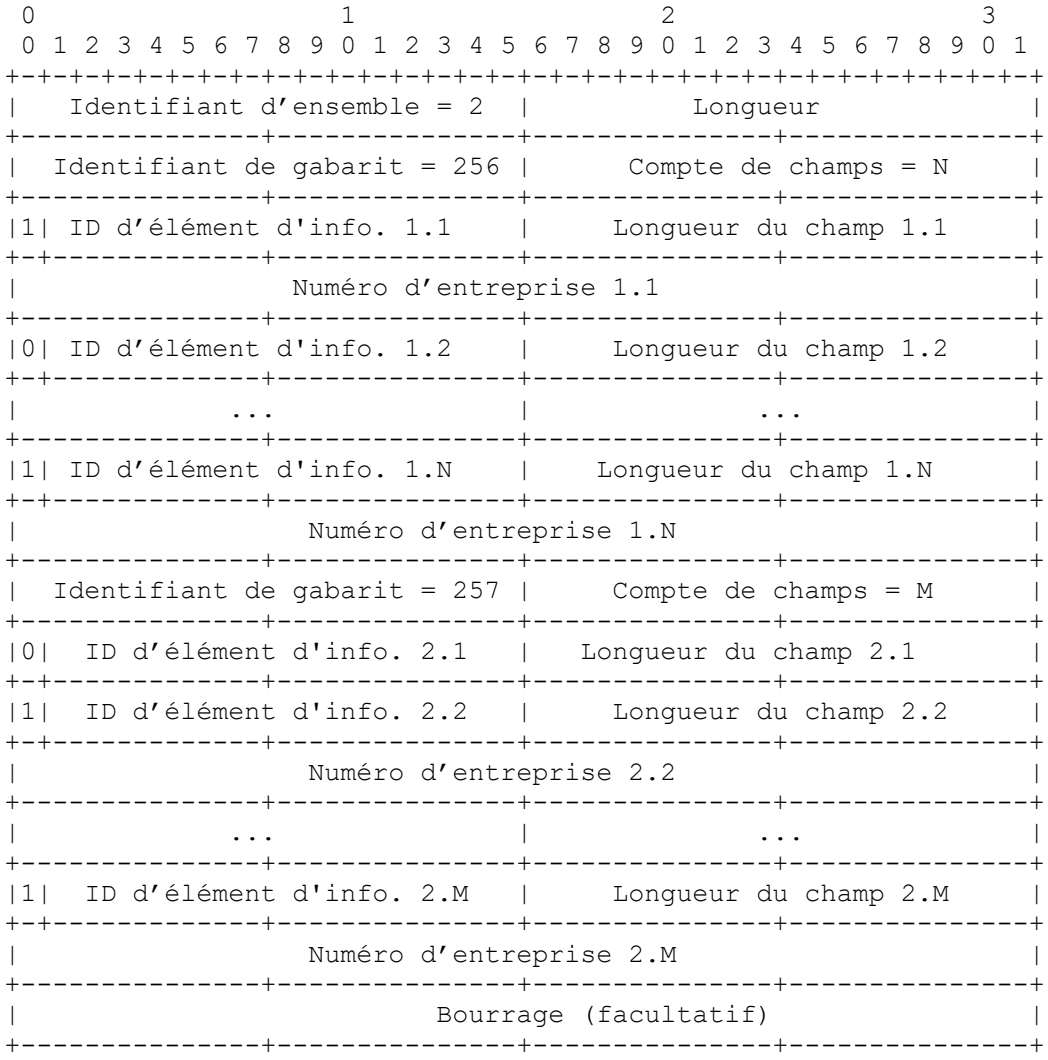


Figure L : Exemple d'ensemble de gabarits

Les identifiants d'élément d'information 1.2 et 2.1 apparaissent dans [IANA-IPFIX] (bit Entreprise = 0) et donc n'ont pas besoin d'un numéro d'entreprise pour les identifier.

3.4.2 Format d'enregistrement de gabarit d'options

Grâce à la notion de portée, l'enregistrement de gabarit d'options donne à l'exportateur la capacité de fournir des informations supplémentaires au collecteur, qui ne seraient pas possibles avec les seuls enregistrements de flux.

Voir à la Section 4 les gabarits d'options spécifiques utilisés pour rapporter des métadonnées sur les processus d'exportation et de mesure IPFIX.

3.4.2.1 Portée

La portée, qui n'est disponible que dans l'ensemble de gabarits d'options, donne le contexte des éléments d'information rapportés dans les enregistrements de données.

La portée est un ou plusieurs éléments d'information, spécifiés dans l'enregistrement de gabarit d'options. Au minimum, le processus de collecte DEVRAIT prendre en charge comme portée les éléments d'information observationDomainId, exportingProcessId, meteringProcessId, templateId, lineCardId, exporterIPv4Address, exporterIPv6Address, et ingressInterface. Le protocole IPFIX n'empêche l'utilisation d'aucun élément d'information pour la portée. Cependant, certains types d'élément d'information n'ont pas de sens spécifiés comme portée (par exemple, les éléments d'information de compteur).

L'en-tête de message IPFIX contient déjà l'identifiant de domaine d'observation. Si ce n'est pas zéro, cet identifiant de domaine d'observation peut être considéré comme une portée implicite pour les enregistrements de données dans le message IPFIX.

Plusieurs champs Portée PEUVENT être présents dans l'enregistrement de gabarit d'options, et dans ce cas la portée composite est la combinaison des portées. Par exemple, si les deux portées sont meteringProcessId et templateId, la portée combinée est ce gabarit pour ce processus de mesures. Si il y a un ordre différent des champs Portée, il en résulterait une signification différente de l'enregistrement, donc l'ordre des champs Portée DOIT être préservé par le processus d'exportation. Par exemple, dans le contexte de PSAMP [RFC5476], si la première portée définit la fonction de filtrage, alors que la seconde portée définit la fonction d'échantillonnage, l'ordre des portées est important. Appliquer d'abord la fonction d'échantillonnage, suivie par la fonction de filtrage, conduirait potentiellement à des enregistrements de données différents de si on applique d'abord la fonction de filtrage, suivie par la fonction d'échantillonnage.

3.4.2.2 Format enregistrement de gabarit d'options

Un enregistrement de gabarit d'options contient toute combinaison d'identifiants d'élément d'information alloués par l'IANA et/ou spécifiques d'enterprise.

Le format de l'enregistrement de gabarit d'options est montré à la Figure M. Il consiste en un en-tête d'enregistrement de gabarit d'options et en un ou plusieurs spécificateurs de champ. Les spécificateurs de champ sont définis à la Figure G ci-dessus.

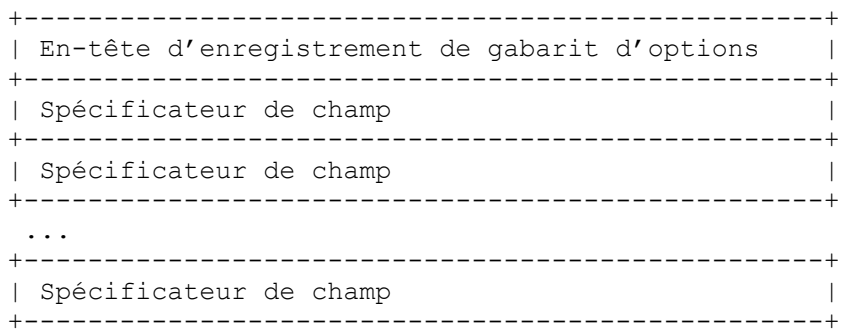


Figure M : Format d'enregistrement de gabarit d'options

Le format de l'en-tête d'enregistrement de gabarit d'options est montré à la Figure N.

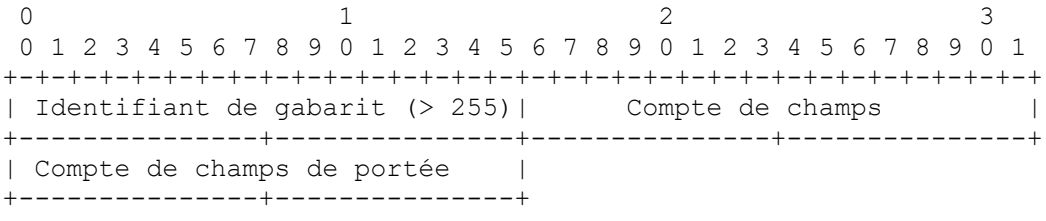


Figure N : Format d'en-tête d'enregistrement de gabarit d'options

Les définitions de champ d'en-tête d'enregistrement de gabarit d'options sont comme suit :

Identifiant de gabarit : chaque enregistrement de gabarit d'options reçoit un identifiant de gabarit unique dans la gamme 256 à 65 535. Cette unicité est locale pour la session de transport et le domaine d'observation qui ont généré l'identifiant de gabarit. Comme les identifiants de gabarit sont utilisés comme identifiants d'ensemble dans les ensembles qu'ils décrivent (voir le paragraphe 3.4.3) les valeurs 0 à 255 sont réservées pour des types d'ensembles particuliers (par exemple, les ensembles de gabarit eux-mêmes) et les gabarits et gabarits d'options ne peuvent pas partager des identifiants de gabarit au sein d'une session de transport et d'un domaine d'observation. Il n'y a pas de contrainte sur l'ordre d'allocation de l'identifiant de gabarit. Comme les processus d'exportation sont libres d'allouer les identifiants de gabarit comme bon leur semble, les processus de collecte NE DOIVENT PAS supposer des identifiants de gabarit incrémentaires, ou quelque chose sur le contenu d'un gabarit d'options fondé sur le seul identifiant de gabarit.

Compte de champs : nombre de tous les champs de cet enregistrement de gabarit d'options, incluant les champs de portée.

Compte de champs de portée : nombre des champs de portée dans cet enregistrement de gabarit d'options. Les champs de portée sont des champs normaux, sauf qu'ils sont interprétés comme de portée au collecteur. Un compte de champs de portée de N spécifie que les N premiers spécificateurs de champ dans l'enregistrement de gabarit sont des champs de portée. Le compte de champs de portée NE DOIT PAS être zéro.

L'exemple de la Figure O montre un ensemble de gabarits d'options avec un mélange d'éléments d'information alloués par l'IANA et spécifiques d'entreprise. Il consiste en un en-tête d'ensemble, un en-tête de gabarit d'options, et plusieurs spécificateurs de champ.

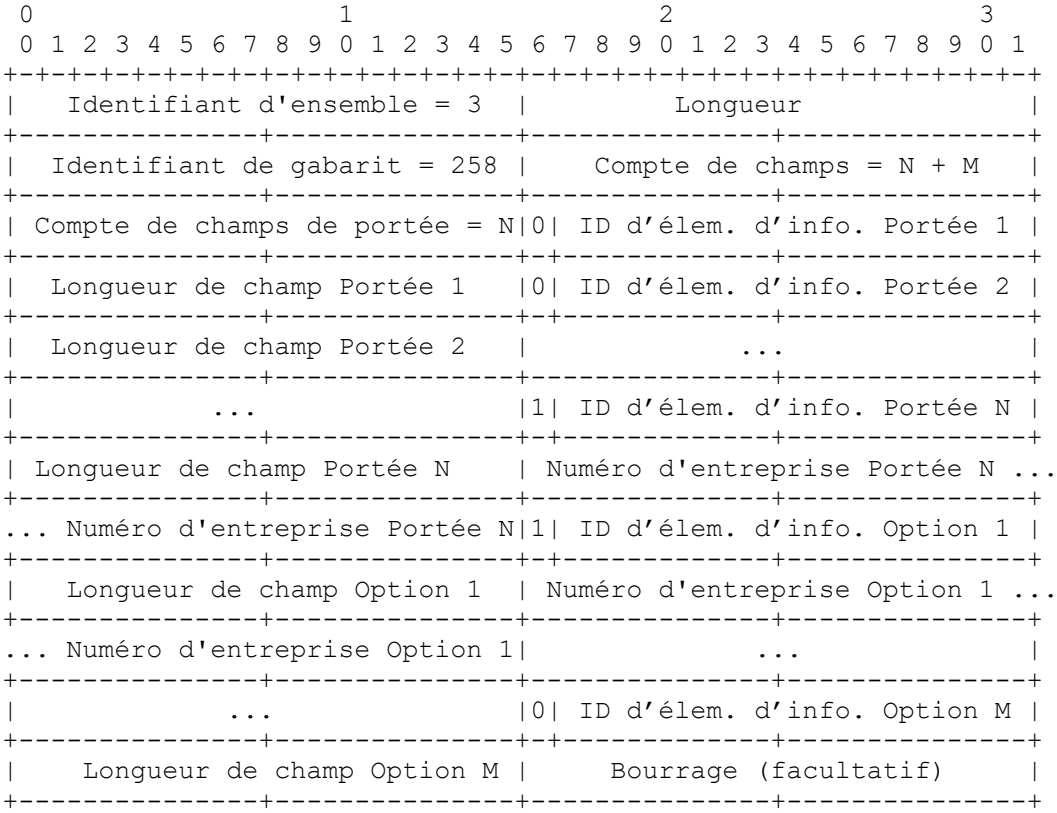


Figure O : Exemple d'ensemble de gabarits d'options

3.4.3 Format enregistrement de données

Les enregistrements de données sont envoyés dans des ensembles de données. Le format de l'enregistrement de données est montré à la Figure P. Il consiste seulement en une ou plusieurs valeurs de champs. L'identifiant de gabarit auquel les valeurs de champs appartiennent est codé dans le champ d'en-tête d'ensemble "Identifiant d'ensemble", c'est-à-dire, "Identifiant d'ensemble" = "Identifiant de gabarit".

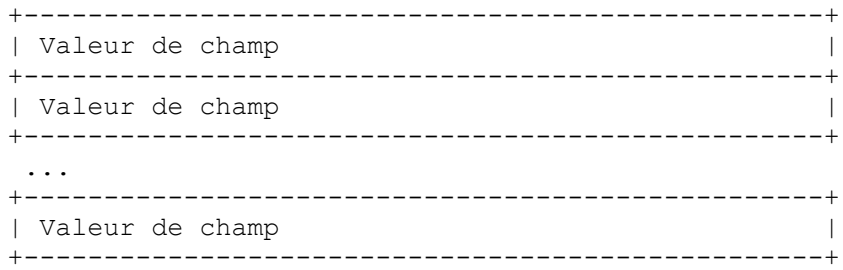


Figure P : Format d'enregistrement de données

Noter que les valeurs de champ n'ont pas nécessairement une longueur de 16 bits. Les valeurs de champs sont codées selon leur type de données comme spécifié dans la [RFC7012].

L'interprétation du format d'enregistrement de données ne peut être fait que si l'enregistrement de gabarit correspondant à l'identifiant de gabarit est disponible au processus de collecte.

L'exemple de la Figure Q montre un ensemble de données. Il consiste en un en-tête d'ensemble et en plusieurs valeurs de champs.

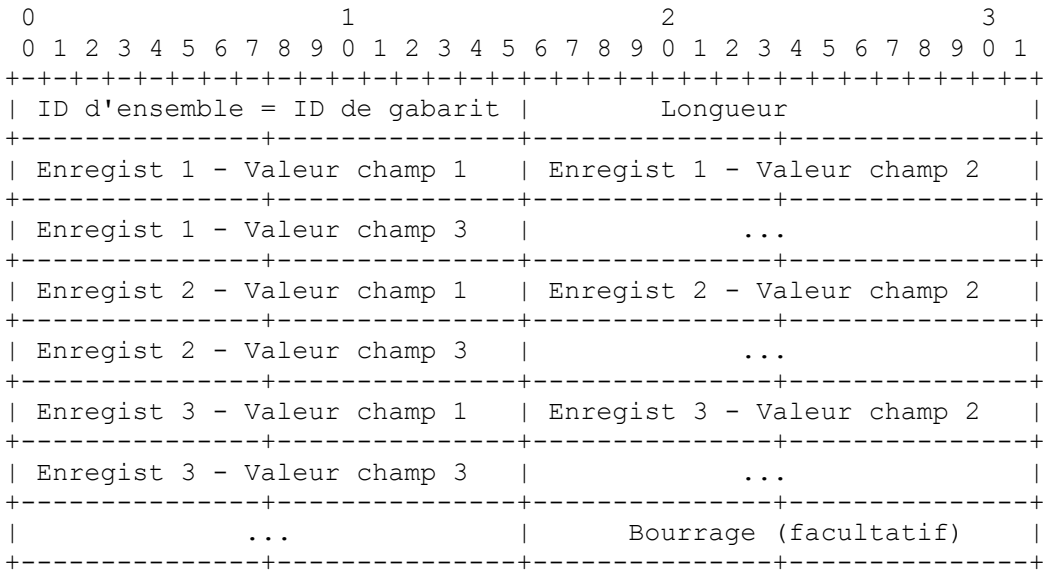


Figure Q : Ensemble de données, contenant des enregistrements de données

4. Exigences spécifique de rapport

Certains gabarits d'options et enregistrements de gabarit d'options spécifiques sont nécessaires pour fournir des informations supplémentaires sur les enregistrements de flux et sur les processus de mesures.

Le gabarit d'options et les enregistrements de gabarit d'options définis dans ces paragraphes, qui imposent des contraintes aux mises en œuvre de processus de mesures et de processus d'exportation, PEUVENT être mis en œuvre. Si ils sont mis en œuvre, les gabarits d'options spécifique DEVRAIENT être mis en œuvre comme spécifié dans ces paragraphes.

L'ensemble minimum d'éléments d'information est toujours spécifié dans ces gabarits d'options IPFIX spécifiques. Néanmoins, des éléments d'information supplémentaires peuvent être utilisés dans ces gabarits d'options spécifiques.

Le processus de collecte DOIT vérifier les combinaisons possibles d'éléments d'information au sein des enregistrements de gabarit d'options pour interpréter correctement les gabarits d'options suivants.

4.1 Processus de mesures de gabarit d'options de statistiques

Le processus de mesures de gabarit d'options de statistiques spécifie la structure d'un enregistrement de données pour rapporter les statistiques du processus de mesures. Il DEVRAIT contenir les éléments d'information suivants, comme défini dans [IANA-IPFIX] :

(portée) observationDomainId : cet élément d'information DOIT être défini comme champ de portée et DOIT être présent, sauf si l'identifiant de domaine d'observation du message qui l'inclut est non zéro.

(portée) meteringProcessId : si il est présent, cet élément d'information DOIT être défini comme champ de portée.

exportedMessageTotalCount

exportedFlowRecordTotalCount

exportedOctetTotalCount

Le processus d'exportation DEVRAIT exporter l'enregistrement de données spécifié par le processus de mesures de gabarit d'options de statistiques de façon régulière ou sur la base d'une politique d'export. Cette périodicité ou cette politique d'export DEVRAIT être configurable.

Noter que si plusieurs processus de mesure sont disponibles sur le domaine d'observation exportateur, l'élément d'information meteringProcessId DOIT être spécifié comme champ de portée supplémentaire.

4.2 Processus de mesures de gabarit d'options de statistiques de fiabilité

Le processus de mesures de gabarit d'options de statistiques de fiabilité spécifie la structure d'un enregistrement de données pour rapporter le manque de fiabilité du processus de mesures. Il DEVRAIT contenir les éléments d'information suivants, comme défini dans [IANA-IPFIX]:

(portée) observationDomainId : cet élément d'information DOIT être défini comme champ de portée et DOIT être présent, sauf si l'identifiant de domaine d'observation du message qui l'inclut est non zéro.

(portée) meteringProcessId : si il est présent, cet élément d'information DOIT être défini comme champ de portée.

IgnoredPacketTotalCount *(compte total de paquets ignorés)*

ignoredOctetTotalCount *(compte total d'octets ignorés)*

heure du premier paquet ignoré : horodatage du premier paquet qui a été ignoré par le processus de mesures. Pour cet horodatage, tout élément d'information d'horodatage suivant peut être utilisé :
- observationTimeSeconds,
- observationTimeMilliseconds,
- observationTimeMicroseconds, ou
- observationTimeNanoseconds.

heure du dernier paquet ignoré : horodatage du dernier paquet qui a été ignoré par le processus de mesures. Pour cet horodatage, tout élément d'information d'horodatage suivant peut être utilisé :
- observationTimeSeconds,
- observationTimeMilliseconds,
- observationTimeMicroseconds, ou
- observationTimeNanoseconds.

Le processus d'exportation DEVRAIT exporter l'enregistrement de données spécifié par le processus de mesures de gabarit d'options de statistiques de fiabilité de façon régulière ou sur la base d'une politique d'export. Cette périodicité ou cette politique d'export DEVRAIT être configurable.

Noter que si plusieurs processus de mesure sont disponibles sur le domaine d'observation exportateur, l'élément d'information meteringProcessId DOIT être spécifié comme champ de portée supplémentaire.

Comme le processus de mesures de gabarit d'options de statistiques de fiabilité contient deux horodatages d'éléments d'information identiques, et comme l'ordre des éléments d'information dans les enregistrements de gabarit n'est pas garanti, le processus de collecte interprête l'intervalle de temps des paquets ignorés comme la gamme entre les deux valeurs ; voir au paragraphe 5.2 les considérations de retour à zéro.

4.3 Processus d'exportation de gabarit d'options de statistiques de fiabilité

Le processus d'exportation de gabarit d'options de statistiques de fiabilité spécifie la structure d'un enregistrement de données pour rapporter le manque de fiabilité du processus d'exportation. Il DEVRAIT contenir les éléments d'information suivants, comme définis dans [IANA-IPFIX] :

(portée) identifiant de processus d'exportation : l'identifiant du processus d'exportation pour lequel la fiabilité est rapportée. Tout élément d'information exporterIPv4Address, exporterIPv6Address, ou exportingProcessId peut être utilisé pour ce champ. Cet élément d'information DOIT être défini comme champ de portée.

NotSentFlowTotalCount *(compte total de flux non envoyés)*

notSentPacketTotalCount *(compte total de paquets non envoyés)*

notSentOctetTotalCount *(compte total d'octets non envoyés)*

heure du premier flux abandonné : heure à laquelle le premier enregistrement de flux a été abandonné par le processus d'exportation. Pour cet horodatage, tout élément d'information d'horodatage suivant peut être utilisé :
observationTimeSeconds,
observationTimeMilliseconds,
observationTimeMicroseconds, ou
observationTimeNanoseconds.

heure du dernier flux abandonné : heure à laquelle le dernier enregistrement de flux a été abandonné par le processus d'exportation. Pour cet horodatage, tout élément d'information d'horodatage suivant peut être utilisé :
observationTimeSeconds,
observationTimeMilliseconds,
observationTimeMicroseconds, ou
observationTimeNanoseconds.

Le processus d'exportation DEVRAIT exporter l'enregistrement de données spécifié par le processus d'exportation de gabarit d'options de statistiques de fiabilité de façon régulière ou sur la base d'une politique d'export. Cette périodicité ou politique d'export DEVRAIT être configurable.

Comme le processus d'exportation de gabarit d'options de statistiques de fiabilité contient deux horodatages d'éléments d'information identiques, et comme l'ordre des éléments d'information dans les enregistrements de gabarit n'est pas garanti, le processus de collecte interprête l'intervalle de temps des paquets ignorés comme la gamme entre les deux valeurs ; voir au paragraphe 5.2 les considérations de retour à zéro.

4.4 Gabarit d'options Clés de flux

Le gabarit d'options Clés de flux spécifie la structure d'un enregistrement de données pour rapporter les clés de flux des flux rapportés. Un enregistrement de données de clés de flux étend un enregistrement de gabarit particulier qui est référencé par son identifiant de gabarit. L'enregistrement de gabarit est étendu en spécifiant quels éléments d'information contenus dans les enregistrements de données correspondants décrivent les propriétés de flux qui servent de clés de flux du flux rapporté.

Le gabarit d'options Clés de flux DEVRAIT contenir les éléments d'information suivants, comme défini dans [IANA-IPFIX] :

(portée) templateId : cet élément d'information DOIT être défini comme champ de portée.

FlowKeyIndicator *(indicateur de clé de flux)*

5. Considérations de temps

5.1 En-tête de message IPFIX Heure d'exportation et Heure d'enregistrement de flux

L'en-tête de message IPFIX Heure d'exportation est l'heure à laquelle l'en-tête de message IPFIX quitte l'exportateur, en utilisant le même codage que le type de données abstraites dateTimeSeconds [RFC7012], c'est-à-dire, exprimé en secondes depuis l'époque UNIX, 1er janvier 1970 à 00:00 UTC, codé comme un entier non signé de 32 bits.

Certains éléments d'information relatifs au temps peuvent être exprimés comme un décalage par rapport à cette heure d'exportation. Par exemple, des enregistrements de données qui exigent une précision d'une microseconde peuvent exporter les heures de début et de fin du flux avec les éléments d'information flowStartMicroseconds et flowEndMicroseconds, qui codent le temps absolu en microsecondes en termes d'époque NTP, le 1er janvier 1900 à 00:00 UTC, dans un champ de 64 bits. Une autre solution est d'exporter les éléments d'information flowStartDeltaMicroseconds et flowEndDeltaMicroseconds dans l'enregistrement de données, qui rapportent respectivement l'heure de début et de fin du flux comme des décalages négatifs par rapport à l'heure d'exportation, comme un entier non signé de 32 bits. Cette dernière solution diminue les exigences de bande passante de l'exportation, économisant quatre octets par horodatage, tout en augmentant la charge pour l'exportateur, car le processus d'exportation doit calculer le flowStartDeltaMicroseconds et flowEndDeltaMicroseconds de chaque enregistrement de données avant d'exporter le message IPFIX.

On doit noter que les horodatages fondés sur l'heure d'exportation imposent des contraintes de temps aux enregistrements de données contenus dans le message IPFIX. Dans l'exemple des éléments d'information flowStartDeltaMicroseconds et flowEndDeltaMicroseconds, l'enregistrement de données peut seulement contenir des enregistrements avec des horodatages dans les 71 minutes de l'heure d'exportation. Autrement, le compteur de 32 bits ne serait pas suffisant pour contenir le décalage de l'heure de début du flux.

5.2 Prise en charge du retour à zéro de l'horodatage

Le type de données abstraites dateTimeSeconds [RFC7012] et le champ d'en-tête de message Heure d'exportation (paragraphe 3.1) sont codés comme des entiers non signés de 32 bits, exprimés en secondes depuis l'époque UNIX, le 1er janvier 1970 à 00:00 UTC, comme défini dans [POSIX.1]. Ces valeurs vont revenir à zéro le 7 février 2106 à 06:28:16 UTC.

Afin de prendre en charge l'utilisation continue du protocole IPFIX au delà de cette date, le processus d'exportation DEVRAIT exporter les valeurs de dateTimeSeconds et le champ d'en-tête de message Heure d'exportation comme le nombre de secondes depuis l'époque UNIX, le 1er janvier 1970 à 00:00 UTC, modulo 2^32. Le processus de collecte DEVRAIT utiliser la date en cours, ou une autre information contextuelle, pour interpréter de façon appropriée les valeurs de dateTimeSeconds et le champ d'en-tête de message Heure d'exportation.

Il y a des considérations similaires pour les types de données abstraits dateTimeMicroseconds et dateTimeNanoseconds fondés sur NTP [RFC7012]. Le processus d'exportation DEVRAIT exporter les valeurs de dateTimeMicroseconds et dateTimeNanoseconds comme si l'ère NTP [RFC5905] était implicite ; le processus de collecte DEVRAIT utiliser la date actuelle, ou autres informations contextuelles, pour déterminert l'ère NTP afin de d'interpréter correctement les valeurs de dateTimeMicroseconds et dateTimeNanoseconds dans les enregistrements de données reçus.

Le type de données abstrait dateTimeMilliseconds va revenir à zéro dans approximativement 500 millions d'années ; la spécification du comportement de ce type de données abstrait après cette date est laissé comme sujet d'une future révision de la présente spécification.

La mémorisation à long terme des fichiers [RFC5655] aux fins d'archivage est affectée par le retour à zéro de l'horodatage, car l'utilisation de la date actuelle pour interpréter les valeurs d'horodatage dans des fichiers mémorisés sur plusieurs décades dans le passé peut conduire à des valeurs incorrectes ; donc, il est RECOMMANDÉ que de tels fichiers soient mémorisés avec des informations de contexte pour aider à l'interprétation de ces horodatages.

6. Lien avec le modèle d'information

Comme avec les valeurs d'en-tête de message IPFIX et d'en-tête d'ensemble, les valeurs de tous les éléments d'information [RFC7012], sauf celles des types de données "string" et "octetArray", sont codées en format canonique dans l'ordre des octets du réseau (aussi appelé gros boutien).

6.1 Codage des types de données IPFIX

Le paragraphes qui suivent définissent le codage des types de données spécifiés dans la [RFC7012].

6.1.1 Types de données Integral

Les types de données Integral -- unsigned8, unsigned16, unsigned32, unsigned64, signed8, signed16, signed32, et signed64 -- DOIVENT être codés en utilisant le format canonique par défaut dans l'ordre des octets du réseau. Les types de données Signed integral sont représentés en notation de complément à deux.

6.1.2 Types Adresse

Les types Address -- macAddress, ipv4Address, et ipv6Address -- DOIVENT être codés de la même façon que les types de données Integral, comme, respectivement six, quatre, et seize octets dans l'ordre des octets du réseau.

6.1.3 float32

Le type de données float32 DOIT être codé comme un type IEEE binary32 à virgule flottante comme spécifié dans [IEEE.754.2008], dans l'ordre des octets du réseau comme spécifié au paragraphe 3.6 de la [RFC1014]. Noter que sur les machines petit boutiennes, l'échange d'octets de la représentation native est nécessaire avant l'exportation. Noter que la méthode pour le faire peut dépendre de la plate-forme de mise en œuvre.

6.1.4 float64

Le type de données float64 DOIT être codé comme un type IEEE binary64 en virgule flottante comme spécifié dans [IEEE.754.2008], dans l'ordre des octets du réseau comme spécifié au paragraphe 3.7 de la [RFC1014]. Noter que sur les machines petit boutiennes, l'échange d'octets de la représentation native est nécessaire avant l'exportation. Noter que la méthode pour le faire peut dépendre de la plate-forme de mise en œuvre.

6.1.5 booléen

Le type de données booléen est spécifié conformément à la TruthValue de la [RFC2579]. Il est codé comme un entier d'un seul octet selon le paragraphe 6.1.1, avec la valeur 1 pour vrai et 2 pour faux. Toute autre valeur est indéfinie.

6.1.6 string et octetArray

Le type de données "string" représente une chaîne de longueur finie de caractères valides du jeu de caractères Unicode. Le type de données string DOIT être codé en format UTF-8 [RFC3629]. La chaîne est envoyée comme un dispositif de zéro, un ou plusieurs octets en utilisant un élément d'information de longueur fixe ou variable. Le processus d'exportation IPFIX NE DOIT PAS envoyer de messages IPFIX contenant des valeurs de chaîne UTF-8 mal formées pour les éléments d'information du type de données "string" ; le processus de collecte DEVRAIT détecter et ignorer de telles valeurs. Voir [UTF8-EXPLOIT] pour les fondements de cette question.

Le type de données octetArray n'a pas de règle de codage ; il représente un dispositif brut de zéro, un ou plusieurs octets, avec l'interprétation des octets définie dans la définition de l'élément d'information.

6.1.7 dateTimeSeconds

Le type de données dateTimeSeconds est un entier non signé de 32 bits dans l'ordre des octets du réseau, contenant le nombre de secondes depuis l'époque UNIX, le 1er janvier 1970 à 00:00 UTC, comme défini dans [POSIX.1]. dateTimeSeconds est codé de façon identique au champ Heure d'exportation d'en-tête de message IPFIX. Il peut représenter des dates entre le 1er janvier 1970 et le 7 février 2106 sans retour à zéro ; voir au paragraphe 5.2 les considérations de retour à zéro.

6.1.8 dateTimeMilliseconds

Le type de données dateTimeMilliseconds est un entier non signé de 64 bits dans l'ordre des octets du réseau, contenant le nombre de millisecondes depuis l'époque UNIX, le 1er janvier 1970 à 00:00 UTC, comme défini dans [POSIX.1]. Il peut représenter des dates depuis le 1er janvier 1970 et pour approximativement les 500 millions d'années à venir sans retour à zéro.

6.1.9 dateTimeMicroseconds

Le type de données dateTimeMicroseconds est un champ de 64 bits codé conformément au format d'horodatage NTP défini à la Section 6 de la [RFC5905]. Ce champ est constitué de deux entiers non signés de 32 bits dans l'ordre des octets du réseau : Secondes et Fraction. Le champ Secondes est le nombre de secondes depuis l'époque NTP, le 1er janvier 1900 à 00:00 UTC.

Le champ Fraction est le nombre de fractions de secondes en unités de 1/(2^32) seconde (approximativement 233 pico secondes). Il peut représenter des dates entre le 1er janvier 1900 et le 8 fécrier 2036 dans l'ère NTP actuelle ; voir au paragraphe 5.2 les considérations sur le retour à zéro.

Noter que dateTimeMicroseconds et dateTimeNanoseconds partagent un codage identique. Le type de données dateTimeMicroseconds est seulement destiné à représenter des horodatages à la précision de la microseconde. Donc, les 11 bits du fond du champ Fraction DEVRAIENT être zéro et DOIVENT être ignorés pour tous les éléments d'information de ce type de données (car 2^11 x 233 picosecondes = 0,477 microseconde).

6.1.10 dateTimeNanoseconds

Le type de données dateTimeNanoseconds est un champ de 64 bits codé selon le format d'horodatage NTP, comme défini à la Section 6 de la [RFC5905]. Ce champ est constitué de deux entiers non signés de 32 bits dans l'ordre des octets du réseau : Secondes et Fraction. Le champ Secondes est le nombre de secondes depuis l'époque NTP, le 1er janvier 1900 à 00:00 UTC. Le champ Fraction est le nombre de fractions de seconde en unités de 1/(2^32) seconde (approximativement 233 picosecondes). Il peut représenter des dates entre le 1er janvier 1900 et le 8 février 2036 dans l'ère NTP actuelle ; voir au paragraphe 5.2 les considérations de retour à zéro.

Noter que dateTimeMicroseconds et dateTimeNanoseconds partagent un codage identique. Il n'y a pas de restriction sur l'interprétation du champ Fraction pour le type de connées dateTimeNanoseconds.

6.2 Codage de taille réduite

Les éléments d'information codés comme types de données signed, unsigned, ou float PEUVENT être codés en utilisant moins d'octets qu'impliqué par leur type dans la définition du modèle d'information, sur la base de l'hypothèse que la plus petite taille est suffisante pour porter toute valeur que l'exportateur peut avoir besoin de livrer. Cela réduit l'exigence de bande passante du réseau entre l'exportateur et le collecteur. Noter que les définitions d'élément d'information [IANA-IPFIX] définissent toujours la taille de codage maximum.

Par exemple, le modèle d'information définit octetDeltaCount comme un type unsigned64, qui exigerait 64 bits. Cependant, si l'exportateur ne va jamais rencontrer localement le besoin d'envoyer une valeur supérieure à 4 294 967 295, il peut choisir d'envoyer plutôt la valeur comme unsigned32.

Ce comportement est indiqué par l'exportateur en spécifiant une taille dans le gabarit avec une longueur plus petite que celle associée au type alloué à l'élément d'information. Dans l'exemple ci-dessus, l'exportateur placerait une longueur de 4 au lieu de 8, dans le gabarit.

Le codage de taille réduite PEUT être appliqué aux types d'entiers suivants : unsigned64, signed64, unsigned32, signed32, unsigned16, et signed16. La propriété signée opposée à non signée de la valeur rapportée DOIT être préservée. La réduction de taille peut être à tout nombre d'octets plus petit que le type original si la valeur des données tient, c'est-à-dire, afin que seuls les zéros en tête soient éliminés. Par exemple, un unsigned64 peut être réduit à 7, 6, 5, 4, 3, 2, ou 1 octets.

Un codage de taille réduite PEUT être utilisé pour réduire float64 en float32. Le float32 a non seulement une gamme de nombres réduite mais, due à la plus petite mantisse, est aussi moins précise. Dans ce cas, le float64 va être réduit à 4 octets.

Le codage de taille réduite NE DOIT PAS être appliqué aux autres types définis dans la [RFC7012] qui impliquent une longueur fixe, car ces types ont une structure interne (comme ipv4Address ou dateTimeMicroseconds) ou des gammes restreintes qui ne conviennent pas pour le codage de taille réduite (comme dateTimeMilliseconds).

Les éléments d'information de type octetArray et string peuvent être exportés en utilisant une longueur quelconque, sous réserve des restrictions de longueur spécifiques de chaque élément d'information, comme noté dans la description de cet élément d'information.

7. Élément d'information de longueur variable

Le mécanisme de gabarit IPFIX est optimisé pour des éléments d'information de longueur fixe [RFC7012]. Lorsque un élément d'information a une longueur variable, le mécanisme suivant DOIT être utilisé pour porter les informations de longueur des éléments d'information alloués par l'IANA et spécifiques d'entreprise.

Dans l'ensemble de gabarit, l'élément d'information Longueur de champ est enregistré comme 65 535. Cette valeur réservée de longueur notifie au processus de collecte que la valeur de longueur de l'élément d'information va être portée dans le contenu de l'élément d'information lui-même.

Dans la plupart des cas, la longueur de l'élément d'information va être de moins de 255 octets. Le mécanisme de codage de la longueur suivant optimise les frais généraux du transport de la longueur de l'élément d'information dans le cas le plus courant. La longueur est portée dans l'octet avant l'élément d'information, comme le montre la Figure R.

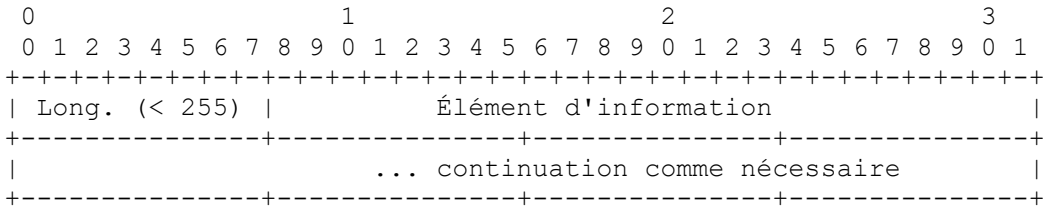


Figure R : Élément d'information (IE) de longueur variable (longueur < 255 octets)

La longueur peut aussi être codée sur 3 octets avant l'élément d'information, permettant que la longueur de l'élément d'information soit supérieure ou égale à 255 octets. Dans ce cas, le premier octet du champ Longueur DOIT être 255, et la longueur est portée dans le second et le troisième octets, comme le montre la Figure S.

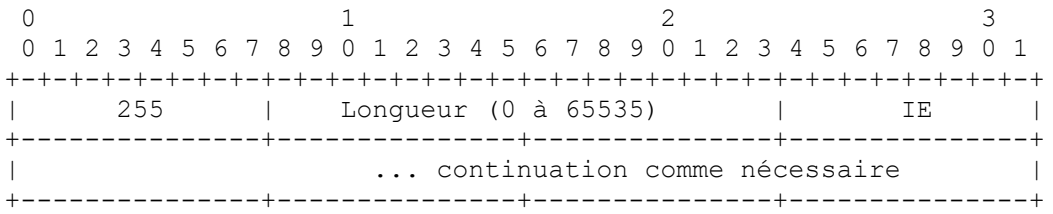


Figure S : Élément d'information (IE) de longueur variable (longueur de 0 à 65 535 octets)

Les octets portant la longueur (soit le premier, soit les trois premiers octets) NE DOIVENT PAS être inclus dans la longueur de l'élément d'information.

8. Gestion de gabarit

Cette Section décrit la gestion des gabarits et gabarits d'option aux processus d'exportation et de collecte. Le but de la gestion de gabarit est de s'assurer, dans la mesure du possible, que le processus d'exportation et le processus de collecte ont une vue cohérente des gabarits et gabarits d'options utilisés pour coder et décoder les enregistrements envoyés du processus d'exportation au processus de collecte.

Atteindre ce but est un peu compliqué par deux facteurs : 1) le besoin de prendre en charge la réutilisation des identifiants de gabarit au sein d'une session de transport et 2) le besoin de prendre en charge la transmission fiable des gabarits quand UDP est utilisé comme protocole de transport pour les messages IPFIX.

Les mécanismes de gestion de gabarit définis dans cette section s'appliquent à l'exportation des messages IPFIX sur SCTP, TCP, ou UDP. Des considérations supplémentaires spécifiques du transport SCTP et UDP sont données aux paragraphes 8.3 et 8.4, respectivement.

Le processus d'exportation alloue et maintient les identifiants de gabarit par session de transport et domaine d'observation. Un nouvel enregistrement de gabarit créé reçoit du processus d'exportation un identifiant de gabarit inutilisé. Le processus de collecte DOIT mémoriser toutes les informations d'enregistrement de gabarit reçues pour la durée de chaque session de transport jusqu'à leur réutilisation ou suppression comme décrit au paragraphe 8.1, ou leur expiration sur UDP comme décrit au paragraphe 8.4, afin qu'il puisse interpréter les enregistrements de données correspondants.

Le processus de collecte NE DOIT PAS supposer que les identifiants de gabarit provenant d'un processus d'exportation donné se réfèrent aux mêmes gabarits auxquels ils se référaient dans les précédentes sessions de transport du même processus d'exportation ; un processus de collecte NE DOIT PAS utiliser des gabarits provenant d'une session de transport pour décoder des ensembles de données dans une session de transport suivante.

Si un élément d'information spécifique est exigé par un gabarit mais n'est pas présent dans les paquets observés, le processus d'exportation PEUT choisir d'exporter les enregistrements de flux sans cet élément d'information dans un enregistrement de données décrit par un nouveau gabarit.

Si un élément d'information est exigé plus d'une fois dans un gabarit, les différentes occurrences de cet élément d'information DEVRAIENT suivre l'ordre logique de leur traitement par le processus de mesures. Par exemple, si un paquet sélectionné passe par deux fonctions de hachage, et si les deux valeurs de hachage sont envoyées dans un seul gabarit, la première occurrence de la valeur de hachage devrait appartenir à la première fonction de hachage dans le processus de mesures. Par exemple, lors de l'exportation de deux adresses de source IP d'un paquet IPv4 dans IPv4, la première occurrence de l'élément d'information sourceIPv4Address devrait être l'adresse IPv4 de l'en-tête externe, tandis que la seconde occurrence devrait être l'adresse de l'en-tête interne. Le processus de collecte DOIT traiter correctement les gabarits avec plusieurs éléments d'information identiques.

Le processus d'exportation DEVRAIT transmettre l'ensemble de gabarits et l'ensemble de gabarits d'options avant tout ensemble de données qui utilise cet identifiant de gabarit (d'options) pour s'assurer que le collecteur a l'enregistrement de gabarit avant de recevoir le premier enregistrement de données. Les enregistrements de données qui correspondent à un enregistrement de gabarit PEUVENT apparaître dans le même message IPFIX et/ou les suivants. Cependant, un processus de collecte NE DOIT PAS supposer que l'ensemble de données et l'ensemble de gabarits (ou ensemble de gabarits d'options) associé sont exportés dans le même message IPFIX.

Bien qu'un processus de collecte reçoive normalement des enregistrements de gabarit provenant du processus d'exportation avant de recevoir des enregistrements de données, ce n'est pas toujours le cas, par exemple, en cas de réarrangement ou de redémarrage du processus de collecte sur UDP. Dans ces cas, le processus de collecte PEUT mettre en mémoire tampon les enregistrements de données pour lesquels il n'a pas de gabarit, en attente des enregistrements de gabarit qui les décrivent ; cependant, on notera que en présence d'une suppression et redéfinition de gabarit (paragraphe 8.1) cela peut conduire à une interprétation incorrecte des enregistrements de données.

Différents domaines d'observation au sein d'une session de transport PEUVENT utiliser la même valeur d'identifiant de gabarit pour se référer à des gabarits différents ; les processus de collecte DOIVENT traiter correctement ce cas.

Les gabarits d'options et les gabarits qui sont en relation ou sont interdépendants (par exemple, en partageant des propriétés communes, comme décrit dans la [RFC5473]) DEVRAIENT être envoyés ensemble dans le même message IPFIX.

8.1 Retrait et redéfinition de gabarit

Les gabarits qui ne vont plus être utilisés par un processus d'exportation PEUVENT être supprimés par l'envoi d'un Retrait de gabarit. Après la réception d'un Retrait de gabarit, un processus de collecte DOIT arrêter d'utiliser le gabarit pour interpréter les ensembles de données exportés ultérieurement. Noter que ce mécanisme ne s'applique pas quand UDP est utilisé pour transporter les messages IPFIX ; pour ce cas, voir le paragraphe 8.4.

Un Retrait de gabarit consiste en un enregistrement de gabarit pour l'identifiant de gabarit qui doit être supprimé, avec un Compte de champs de 0. Le format d'un Retrait de gabarit est montré à la Figure T.

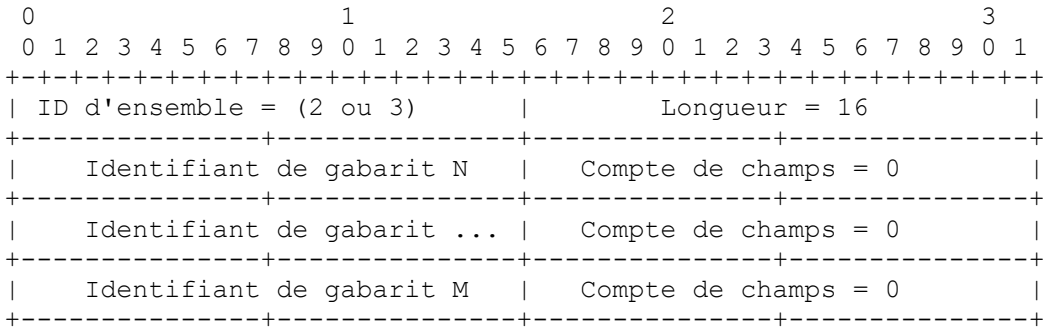


Figure T : Format de retrait de gabarit

Le champ Identifiant d'ensemble DOIT contenir la valeur 2 pour un retrait d'ensemble de gabarits ou la valeur 3 pour un retrait d'ensemble de gabarits d'options. Plusieurs identifiants de gabarit PEUVENT être retirés avec un seul Retrait de gabarit ; dans ce cas, un bourrage PEUT être utilisé.

Les retraits de gabarit PEUVENT apparaître entrelacés avec des ensembles de gabarits, des ensembles de gabarits d'options, et des ensembles de données au sein d'un message IPFIX. Dans ce cas, les gabarits et retraits de gabarits devront être interprétés comme prenant effet dans l'ordre de leur apparition dans le message IPFIX. Un processus d'exportation NE DEVRAIT PAS envoyer de retrait de gabarit avant qu'un délai suffisant se soit écoulé pour permettre la réception et le traitement de tous les enregistrements de données décrits par les retraits de gabarits ; voir au paragraphe 8.2 les détails du séquençage des actions de gestion de gabarits.

La fin d'une session de transport supprime implicitement tous les gabarits utilisés dans la session de transport, et les gabarits doit être renvoyés durant les sessions de transport suivantes entre un processus d'exportation et un processus de collecte. Cela s'applique seulement à SCTP et TCP ; voir les paragraphes 8.4 et 10.3.4 pour la discussion de la session de transport et la durée de vie de gabarit sur UDP.

Tous les gabarits pour un domaine d'observation PEUVENT aussi être supprimés en utilisant un Retrait de tous les gabarits, comme le montre la Figure U. Tous les gabarits d'options pour un domaine d'observation donné PEUVENT de même être supprimés en utilisant un Retrait de tous les gabarits d'options, comme le montre la Figure V.

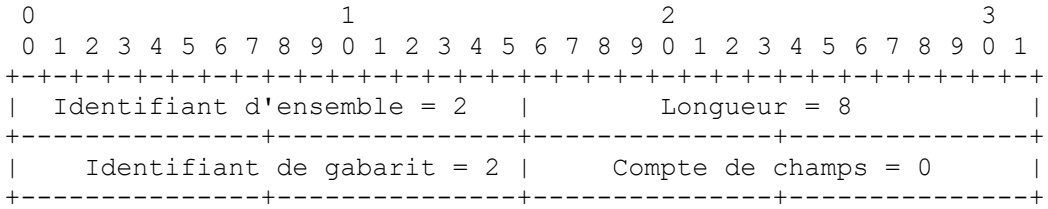


Figure U : Format de l'ensemble Retrait de tous les gabarits

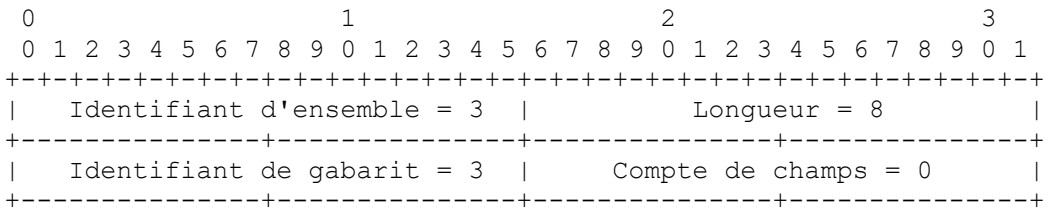


Figure V : Format de l'ensemble Retrait de tous les gabarits d'options

Les identifiants de gabarit PEUVENT être réutilisés pour de nouveaux gabarits en envoyant un nouvel enregistrement de gabarit ou enregistrement de gabarit d'options pour un identifiant de gabarit après la suppression du gabarit.

Si un processus de collecte reçoit un Retrait de gabarit pour un gabarit ou un gabarit d'options qu'il n'a pas mémorisé actuellement, cela indique un disfonctionnement ou un processus d'exportation mis en œuvre de façon inappropriée. La poursuite de la réception et de l'interprétation des enregistrements de données est encore possible, mais le processus de collecte DOIT ignorer le Retrait de gabarit et DEVRAIT enregistrer l'erreur.

Si un processus de collecte reçoit un nouvel enregistrement de gabarit ou de gabarit d'options pour un identifiant de gabarit déjà alloué, et si ce gabarit ou gabarit d'options est identique au gabarit ou gabarit d'options déjà reçu, il DEVRAIT enregistrer la retransmission ; cependant, ceci n'est pas une condition d'erreur, car cela n'affecte pas l'interprétation des enregistrements de données.

Si un processus de collecte reçoit un nouvel enregistrement de gabarit ou enregistrement de gabarit d'options pour un identifiant de gabarit déjà alloué, et si ce gabarit ou gabarit d'options est différent du gabarit ou gabarit d'option déjà reçu, cela indique un disfonctionnement ou une mise en œuvre de processus d'exportation inappropriée. La poursuite de la réception et l'interprétation sans ambiguité des enregistrements de données pour cet identifiant de gabarit ne sont plus possibles, et le processus de collecte DEVRAIT enregistrer l'erreur. Les autres actions de processus de collecte sortent du domaine d'application de cette spécification.

8.2 Actions de gestion de séquençage de gabarit

Comme il n'y a pas de garantie de l'ordre des messages IPFIX exportés à travers les flux SCTP ou sur UDP, un processus d'exportation DOIT séquencer toutes les actions de gestion de gabarit (c'est-à-dire, les enregistrements de gabarit qui définissent de nouveaux gabarits et les Retraits de gabarit qui les suppriment) en utilisant le champ Heure d'exportation dans l'en-tête de message IPFIX.

Un processus d'exportation NE DOIT PAS exporter un ensemble de données décrit par un nouveau gabarit dans un message IPFIX avec une heure d'exportation avant l'heure d'exportarion du message IPFIX qui contient ce gabarit. Si un nouveau gabarit et un ensemble de données qui le décrit apparaissent dans le même message IPFIX, l'ensemble de gabarits contenant le gabarit DOIT apparaître avant l'ensemble de données dans le message.

Un processus d'exportation NE DOIT PAS exporter d'ensembles de données décrits par un Retrait de gabarit dans des messages IPFIX avec une heure d'exportation après l'heure d'exportation du message IPFIX contenant le Retrait de gabarit qui supprime ce gabarit.

Dit autrement, un gabarit décrit les enregistrements de données contenus dans des messages IPFIX quand l'heure d'exportation de ces messages est entre un instant de début et de fin spécifique, inclus. L'instant de début est l'heure d'exportation du message IPFIX contenant l'enregistrement de gabarit. L'instant de fin est un des deux suivants : si le gabarit est supprimé pendant la session, il est alors l'heure d'exportation du message IPFIX contenant le Retrait de gabarit pour le gabarit ; autrement, c'est la fin de la session de transport.

Même si ils sont envoyés dans l'ordre, les messages IPFIX contenant des actions de gestion de gabarit pourraient arriver déclassés au processus de collecte, c'est-à-dire, si ils sont envoyés via UDP ou via des flux SCTP différents. Cela étant, des retraits de gabarit et une réutilisation ultérieure des identifiants de gabarit peuvent compliquer de façon significative le problème de la détermination des durées de vie des gabarits chez le processus de collecte. Un processus de collecte PEUT mettre en œuvre une mémoire tampon et utiliser les informations d'heure d'exportation pour préciser l'ordre des actions de gestion de gabarit. Cette mémoire tampon, si elle est mise en œuvre, DEVRAIT être configurable pour imposer un délai à l'ordre du délai maximum de réarrangement subi par le processus de collecte. Noter que dans ce cas, l'horloge du processus de collecte n'est pas pertinente : il faut seulement comparer les heures d'exportation des messages.

8.3 Considérations supplémentaires pour la gestion de gabarits sur SCTP

Les spécifications de ce paragraphe s'appliquent seulement à SCTP ; en cas de contradiction avec les spécifications de la Section 8 ou du paragraphe 8.1, ce paragraphe a la préséance.

Les ensembles de gabarits et les ensembles de gabarits d'options PEUVENT être envoyés sur tout flux SCTP. Les ensembles de données envoyés sur un flux SCTP PEUVENT être représentés par des enregistrements de gabarit exportés sur tout flux SCTP.

Les ensembles de gabarits et les ensembles de gabarits d'options DOIVENT être envoyés de façon fiable, en utilisant une livraison SCTP ordonnée.

Les retraits de gabarits PEUVENT être envoyés sur tout flux SCTP. Les retraits de gabarits DOIVENT être envoyés de façon fiable, en utilisant une livraison SCTP ordonnée. Les identifiants de gabarit PEUVENT être réutilisés par l'envoi d'un Retrait de gabarit et/ou un nouvel enregistrement de gabarit sur un flux SCTP différent du flux sur lequel le gabarit original a été envoyé.

Des considérations de gestion de gabarit supplémentaires sont fournies dans la [RFC6526], qui spécifie une extension pour relier explicitement les gabarits avec les flux SCTP. En échange de règles plus restrictives sur l'allocation des enregistrements de gabarit aux flux SCTP, cette extension permet une réutilisation rapide et fiable des identifiants de gabarit et une estimation des pertes d'enregistrement de données par gabarit.

8.4 Considérations supplémentaires pour la gestion de gabarits sur UDP

Les spécifications de ce paragraphe s'appliquent seulement à UDP ; en cas de contradiction avec les spécifications de la Section 8 ou du paragraphe 8.1, ce paragraphe a la préséance.

Comme UDP ne fournit pas de méthode pour une transmission fiable des gabarits, les processus d'exportation qui utilisent UDP comme protocole de transport DOIVENT périodiquement retransmettre chaque gabarit actif à des intervalles réguliers. L'intervalle de retransmission de gabarit DOIT être configurable via, par exemple, les paramètres templateRefreshTimeout et optionsTemplateRefreshTimeout comme définis dans la [RFC6728]. Les réglages par défaut de ces valeurs sont spécifiques du déploiement et de l'application.

Avant d'exporter tout enregistrement de données décrit par un enregistrement de gabarit ou enregistrement de gabarit d'options, en particulier dans le cas de réutilisation d'identifiant de gabarit comme décrit au paragraphe 8.1, le processus d'exportation DEVRAIT envoyer plusieurs copies de l'enregistrement de gabarit dans un message IPFIX séparé, afin de s'assurer que le processus de collecte l'a reçu.

Afin de minimiser les exigences de ressource pour les gabarits qui ne sont plus utilisés par le processus d'exportation, le processus de collecte PEUT associer une durée de vie à chaque gabarit reçu dans une session de transport. Les gabarits non rafraîchis par le processus d'exportation pendant la durée de vie peuvent alors être éliminés par le processus de collecte. La durée de vie de gabarit au processus de collecte PEUT être exposée par un paramètre de configuration ou PEUT être déduite de l'observation de l'intervalle des retransmissions périodiques de gabarit par le processus d'exportation. Dans ce dernier cas, la durée de vie de gabarit DEVRAIT par défaut être d'au moins trois fois le taux de retransmission observé.

Les retraits de gabarit (paragraphe 8.1) NE DOIVENT PAS être envoyés par un processus d'exportation qui exporte via UDP et DOIVENT être ignorés par les processus de collecte qui collectent via UDP. Les identifiants de gabarit PEUVENT être réutilisés par les processus d'exportation en exportant un nouveau gabarit pour l'identifiant de gabarit après une attente d'au moins trois fois le délai de retransmission. Noter que la réutilisation d'un identifiant de gabarit peut conduire à une interprétation incorrecte des enregistrements de données si la retransmission et la durée de vie ne sont pas configurées correctement.

Quand un processus de collecte reçoit un nouvel enregistrement de gabarit ou enregistrement de gabarit d'options via UDP pour un identifiant de gabarit déjà alloué, et que le gabarit ou gabarit d'options est identique au gabarit ou gabarit d'options déjà reçu, il NE DEVRAIT PAS enregistrer la retransmission, car c'est le fontionnement normal du rafraîchissement de gabarit sur UDP.

Quand un processus de collecte reçoit un nouvel enregistrement de gabarit ou enregistrement de gabarit d'options pour un identifiant de gabarit déjà alloué, et que ce gabarit ou gabarit d'options est différent du gabarit ou gabarit d'options déjà reçu, le processus de collecte DOIT remplacer le gabarit ou gabarit d'options pour cet identifiant de gabarit par le nouveau gabarit ou gabarit d'options reçu. C'est le fonctionnement normal de la réutilisation d'identifiant de gabarit sur UDP.

Comme les identifiants de gabarit sont uniques par session UDP et par domaine d'observation, à tout moment, le processus de collecte DEVRAIT conserver ce qui suit pour tous les enregistrements de gabarit et enregistrements de gabarit d'options en cours : <Appareils IPFIX, Accès UDP de source exportateur, Adresse IP du collecteur, Accès UDP de destination du collecteur, Identifiant du domaine d'observation, Identifiant de gabarit, Définition de gabarit, Dernier reçu>.

9. Processus de collecte

Cette section décrit le traitement du protocole IPFIX au processus de collecte commun à tous les protocoles de transport. Des considérations supplémentaires pour SCTP et UDP sont fournies respectivement aux paragaphes 9.2 et 9.3. La gestion de gabarit au processus de collecte est traitée à la Section 8.

Le processus de collecte DOIT écouter les demandes/connexions d'association pour débuter de nouvelles sessions de transport de la part du processus d'exportation.

Le processus de collecte DOIT noter l'identifiant d'élément d'information de tout élément d'information qu'il ne comprend pas et PEUT éliminer cet élément d'information des enregistrements de données reçus.

Le processus de collecte DOIT accepter le bourrage dans les enregistrements de données et les enregistrements de gabarit. La taille du bourrage est la longueur de l'ensemble moins la taille de l'en-tête de l'ensemble (4 octets pour l'identifiant d'ensemble et la longueur de l'ensemble) modulo la taille minimum d'enregistrement déduite de l'enregistrement de gabarit.

Le protocole IPFIX a un champ Numéro de séquence dans l'en-tête Export qui augmente avec le nombre d'enregistrements de données IPFIX dans le message IPFIX. Un collecteur peut supprimer les messages IPFIX hors séquence, abandonnés, ou dupliqués en suivant le numéro de séquence. Un collecteur DEVRAIT fournir un mécanisme d'enregistrement pour suivre les messages IPFIX hors séquence. De tels messages IPFIX hors séquence peuvent être dus à l'épuisement des ressources de l'exportateur lorsque il ne peut pas transmettre les messages à leur rythme de création, lors d'une réinitialisation du processus d'exportation, à cause d'encombrement sur la liaison réseau entre l'exportateur et le collecteur, de l'épuisement des ressources du collecteur où il ne peut pas traiter les messages IPFIX à leur rythme d'arrivée, de la réception de paquets déclassés, de la réception de paquets dupliqués, ou d'un attaquant qui injecte de faux messages.

9.1 Traitement par le processus de collecte des messages IPFIX mal formés

Si le processus de collecte reçoit un message IPFIX mal formé, il DOIT éliminer le message IPFIX et DEVRAIT enregistrer l'erreur. Un message IPFIX mal formé est un message qui ne peut pas être interprété à cause de valeurs de longueur absurdes (par exemple, un élément d'information de longueur variable plus long que l'ensemble qui le contient, un ensemble plus long que le message IPFIX qui le contient, ou un message IPFIX plus court qu'un en-tête de message IPFIX) ou une valeur de version réservée (qui peut indiquer qu'une future version de IPFIX est utilisée pour l'export mais en pratique se produit le plus souvent quand des données non IPFIX sont envoyées à un processus de collecte IPFIX). Noter que le bourrage d'un ensemble non zéro ne constitue pas un message IPFIX mal formé.

Comme la cause la plus probable de messages IPFIX mal formés est une mise en œuvre de processus d'exportation déficiente, ou l'envoi de données non IPFIX à un processus de collecte IPFIX, une intervention humaine est probablement nécessaire pour corriger le problème. Pendant ce temps, le processus de collecte PEUT tenter de rectifier la situation de toutes les façons qu'il juge appropriée, incluant :
- de terminer la connexion TCP ou SCTP,
- d'utiliser la fenêtre de réception pour réduire la charge du réseau pour le processus d'exportation qui fontionne mal,
- de mettre en mémoire tampon et sauvegarder le ou les messages IPFIX mal formés pour aider au diagnostic,
- de tenter de resynchroniser le flux, par exemple, comme décrit au paragraphe 10.3 de la [RFC5655].

La resynchronisation ne devrait être tentée que si le processus de collecte a des raisons de croire que l'erreur est transitoire. Par ailleurs, le processus de collecte DEVRAIT arrêter de traiter les messages IPFIX provenant d'un processus d'exportation clairement défaillant (par exemple, celui dont les derniers messages IPFIX étaient mal formés).

9.2 Considérations supplémentaires pour les processus de collecte SCTP

Comme un processus d'exportation peut demander et prendre en charge plus d'un flux par association SCTP, le processus de collecte DOIT prendre en charge l'ouverture de plusieurs flux SCTP.

9.3 Considérations supplémentaires pour les processus de collecte UDP

Une session de transport pour les messages IPFIX transportés sur UDP est définie du point de vue du processus d'exportation et correspond en gros à la durée pendant laquelle un processus d'exportation envoie des messages IPFIX sur UDP à un processus de collecte. Comme ceci est difficile à détecter au processus de collecte, le processus de collecte PEUT éliminer tout l'état de session de transport après qu'aucun message IPFIX n'est reçu d'un certain processus d'exportation dans une certaine session de transport durant une temporisation d'inactivité configurable.

Le processus de collecte DEVRAIT accepter des enregistrements de données sans l'enregistrement de gabarit associé (ou d'autres définitions comme des propriétés communes) exigé pour décoder l'enregistrement de données. Si les enregistrements de gabarit ou autres définitions n'ont pas été reçus au moment où les enregistrements de données sont reçus, le processus de collecte PEUT mémoriser les enregistrements de données pendant un court délai et les décoder après que les enregistrements de gabarit ou autres définitions sont reçus, en comparant les heures d'exportation des messages IPFIX contenant les enregistrements de gabarit avec ceux contenant les enregistrements de données, comme discuté au paragraphe 8.2. Noter que ce mécanisme peut conduire à une interprétation incorrecte des enregistrements en présence de réutilisation d'identifiant de gabarit ou autres identifiants avec des durées de vie limitée.

10. Protocole de transport

La spécification du protocole IPFIX a été conçue pour être indépendante du protocole de transport. Noter que l'exportateur peut exporter plusieurs processus de collecte en utilisant des protocoles de transport indépendants.

Le champ Longueur de l'en-tête de message IPFIX de 16 bits limite la longueur d'un message IPFIX à 65 535 octets, incluant l'en-tête. Un processus de collecte DOIT être capable de traiter des longueurs de message IPFIX jusqu'à 65 535 octets.

Bien qu'un processus d'exportation ou un processus de collecte puisse prendre en charge plusieurs protocoles de transport, les sessions de transport sont liées à un protocole de transport. L'état de la session de transport NE DOIT PAS être changé par un processus d'exportation ou processus de collecte parmi les sessions de transport en utilisant différents protocoles de transport entre la même paire de processus d'exportation et de collecte. En d'autres termes, un processus d'exportation prenant en charge plusieurs protocoles de transport est conceptuellement plusieurs processus d'exportation, un par protocole de transport pris en charge. De même, un processus de collecte prenant en charge plusieurs protocoles de transport est conceptuellement plusieurs processus de collecte, un par protocole de transport pris en charge.

10.1 Conformité au transport et usage du transport

SCTP [RFC4960] DOIT être mis en œuvre en utilisant l'extension SCTP partiellement fiable (PR-SCTP, *Partially Reliable SCTP*) spécifiée dans la [RFC3758] par toutes les mises en œuvre conformes. UDP [RFC0768] PEUT aussi être mis en œuvre par les applications conformes. TCP [RFC0793] PEUT aussi être mis en œuvre par les applications conformes.

SCTP devrait être utilisé dans les déploiements où les exportateurs et les collecteurs communiquent sur des liaisons qui sont susceptibles d'encombrement. SCTP est capable de fournir tous les degrés de fiabilité requis quand utilisé avec l'extension PR-SCTP.

TCP peut être utilisé dans des déploiements où les exportateurs et les collecteurs communiquent sur des liaisons qui sont susceptibles d'encombrement, mais SCTP est préféré du fait de sa capacité de limiter la pression sur les exportateurs et de son orientation message plutôt que flux.

UDP peut être utilisé, bien que ce ne soit pas un protocole adapté à l'encombrement. Cependant, dans ce cas le trafic IPFIX entre l'exportateur et le collecteur doit être contenu séparément ou provisionné à minimiser le risque de pertes liées à l'encombrement.

Par défaut, le processus de collecte écoute les connexions sur l'accès SCTP, TCP, et/ou UDP 4739. Par défaut, le processus de collecte écoute pour des connexions sûres sur l'accès SCTP, TCP, et/ou UDP 4740 (voir la section des considérations sur la sécurité). Par défaut, le processus d'exportation tente de se connecter à un de ces accès. Il DOIT être possible de configurer les processus d'exportation et de collecte à utiliser des accès différents de celui par défaut.

10.2 SCTP

Ce paragraphe décrit comment IPFIX est transporté sur SCTP [RFC4960] en utilisant l'extension PR-SCTP [RFC3758].

10.2.1 Évitement d'encombrement

SCTP fournit par sa conception le niveau requis d'évitement d'encombrement.

SCTP détecte l'encombrement dans le chemin de bout en bout entre le processus d'exportation IPFIX et le processus de collecte IPFIX, et limite le taux de transfert en conséquence. Quand un processus d'exportation IPFIX a des enregistrements à exporter mais détecte que la transmission par SCTP est temporairement impossible, il peut soit attendre que l'envoi soit à nouveau possible, soit décider d'abandonner l'enregistrement. Dans ce dernier cas, les données d'exportation abandonnées DEVRAIENT être prises en compte, afin que la quantité de données d'exportation abandonnées puisse être rapportée en utilisant le mécanisme décrit au paragraphe 4.3.

10.2.2 Fiabilité

Le protocole de transport SCTP est fiable par défaut mais a la capacité de livrer les messages avec une fiabilité partielle [RFC3758].

L'utilisation de messages SCTP fiables pour l'exportation IPFIX n'est pas par elle-même une garantie que tous les enregistrements de données vont être livrés. Si il y a de l'encombrement sur la liaison du processus d'exportation au processus de collecte, ou si un nombre significatif de retransmissions sont requises, les files d'attente d'envoi sur le processus d'exportation peuvent être saturées ; le processus d'exportation PEUT suspendre, exporter, ou éliminer les messages IPFIX. Si les enregistrements de données sont éliminés, les numéros de séquence IPFIX utilisés pour l'exportation DOIVENT refléter la perte des données.

10.2.3 MTU

SCTP fournit le service requis de fragmentation de message IPFIX fondé sur la découverte de la MTU du chemin (PMTU, *Path MTU*).

10.2.4 Établissement et fermeture d'association

Le processus d'exportation IPFIX initie une association SCTP avec le processus de collecte IPFIX. Le processus d'exportation PEUT établir plus d'une association ("faisceau" de connexions dans la terminologie SCTP) avec le processus de collecte.

Un processus d'exportation PEUT prendre en charge plus d'une association active aux différents processus de collecte (incluant le cas de différents processus de collecte sur le même hôte).

Quand un processus d'exportation est fermé, il DEVRAIT fermer l'association SCTP.

Quand un processus de collecte ne veut plus recevoir de messages IPFIX, il DEVRAIT fermer son extrémité de l'association. Le processus de collecte DEVRAIT continuer de recevoir et traiter les messages IPFIX jusqu'à ce que le processus d'exportation ferme son extrémité de l'association.

Quand un processus de collecte détecte que l'association SCTP a été terminée de façon anormale, il DOIT continuer d'écouter pour l'établissement d'une nouvelle association.

Quand un processus d'exportation détecte que l'association SCTP au processus de collecte est terminée de façon anormale, il DEVRAIT essayer de rétablir l'association.

Les temporisations d'association DEVRAIENT être configurables.

10.2.5 Reprise sur défaillance

Si le processus de collecte n'accuse pas réception d'une tentative du processus d'exportation d'établir une association, SCTP va automatiquement réessayer l'établissement d'association en utilisant un retard exponentiel. L'exportateur PEUT enregistrer une alarme si l'établissement d'association SCTP sous-jacente arrive en fin de temporisation ; cette temporisation devrait être configurable chez l'exportateur.

Le processus d'exportation PEUT ouvrir à l'avance une association SCTP de sauvegarde avec le processus de collecte, si il prend en charge la reprise sur défaillance de processus de collecte.

10.2.6 Flux

Un processus d'exportation PEUT demander plus d'un flux SCTP par association. Chacun de ces flux peut être utilisé pour la transmission de messages IPFIX contenant des ensembles de données, des ensembles de gabarits, et/ou des ensembles de gabarits d'options.

Selon les exigences de l'application, le processus d'exportation peut envoyer des ensembles de données avec fiabilité pleine ou partielle, en utilisant la livraison ordonnée ou sans ordre, sur tout flux SCTP établi durant l'établissement de l'association SCTP.

Un processus d'exportation IPFIX PEUT utiliser toute définition de service PR-SCTP conformément à la Section 4 de la spécification de PR-SCTP [RFC3758] quand il utilise la fiabilité partielle pour transmettre des messages IPFIX contenant seulement des ensembles de données.

Cependant, le processus d'exportation DEVRAIT marquer de tels messages IPFIX pour retransmission pour autant que les ressources ou autres contraintes le permettent.

10.3 UDP

Ce paragraphe décrit comment IPFIX est transporté sur UDP [RFC0768].

10.3.1 Évitement d'encombrement

UDP n'a pas de mécanisme intégré d'évitement d'encombrement. Son utilisation sur des chemins de réseau sensibles à l'encombrement n'est donc pas recommandée. UDP PEUT être utilisé dans des déploiements où les exportateurs et les collecteurs communiquent toujours sur des liaisons dédiées qui ne sont pas susceptibles d'encombrement, c'est-à-dire, des liaisons qui sont sur-provisionnées par rapport au taux maximum d'exportation des exportateurs.

10.3.2 Fiabilité

UDP n'est pas un protocole de transport fiable et ne peut pas garantir la livraison des messages. Les messages IPFIX envoyés du processus d'exportation au processus de collecte en utilisant UDP peuvent donc être perdus. UDP NE DOIT PAS être utilisé sauf si l'application peut tolérer des pertes de messages IPFIX.

Le processus de collecte DEVRAIT déduire les pertes et le réarrangement des enregistrements de données IPFIX en cherchant les discontinuités dans les numéros de séquence IPFIX. Dans le cas de UDP, le numéro de séquence IPFIX contient le nombre total d'enregistrements de données IPFIX envoyés pour la session de transport avant la réception de ce message IPFIX, modulo 2^32. Un collecteur DEVRAIT détecter les messages IPFIX hors séquence, abandonnés, ou dupliqués en retraçant les numéros de séquence.

Un processus d'exportation qui exporte des messages IPFIX via UDP DOIT inclure une somme de contrôle UDP valide [RFC0768] dans les datagrammes UDP incluant des messages IPFIX.

10.3.3 MTU

La taille maximum des messages exportés DOIT être configurée de telle façon que la taille totale de paquet n'excède pas la PMTU. Si la PMTU est inconnue, une taille maximum de paquet de 512 octets DEVRAIT être utilisée.

10.3.4 Établissement et fermeture de session

Comme UDP est un protocole sans connexion, il n'y a pas de réel établissement ou fermeture de session pour IPFIX sur UDP. Un processus d'exportation commence à envoyer des messages IPFIX à un processus de collecte à un moment et arrête d'en envoyer à un autre moment. Cela peut conduire à des complications dans la gestion de gabarits, comme souligné au paragraphe 8.4 ci-dessus.

10.3.5 Reprise sur défaillance et duplication de session

Parce que UDP n'est pas un protocole en mode connexion, le processus d'exportation est incapable de déterminer à partir du protocole de transport que le processus de collecte n'est plus capable de recevoir les messages IPFIX. Donc, il ne peut pas invoquer un mécanisme de reprise sur défaillance. Cependant, le processus d'exportation PEUT dupliquer le message IPFIX à plusieurs processus de collecte.

10.4 TCP

Ce paragraphe décrit comment IPFIX est transporté sur TCP [RFC0793].

10.4.1 Évitement d'encombrement

TCP contrôle le taux d'envoi des données du processus d'exportation au processus de collecte, en utilisant un mécanisme qui tient compte de l'encombrement dans le réseau et des capacités du receveur.

Donc, un processus d'exportation IPFIX peut n'être pas capable d'envoyer des messages IPFIX au taux où le processus de mesures les génère, soit à cause d'encombrement dans le réseau, soit parce que le processus de collecte ne peut pas traiter les messages IPFIX assez vite. Tant que l'encombrement est temporaire, le processus d'exportation peut mettre les messages IPFIX en mémoire tampon pour leur transmission. Mais une telle mise en mémoire tampon est nécessairement limitée, à cause des limitations de ressources et à cause des exigences de temps, de sorte qu'un encombrement durable et/ou sévère peut conduire à une situation où le processus d'exportation est bloqué.

Quand un processus d'exportation a des enregistrements de données à exporter mais que la mémoire tampon de transmission est pleine, et qu'il veut éviter un blocage, il peut décider d'abandonner certains enregistrements de données. Les enregistrements de données abandonnés DOIVENT être pris en compte, afin que le nombre d'enregistrements perdus puisse plus tard être rapporté comme décrit au paragraphe 4.3.

10.4.2 Fiabilité

TCP assure la livraison fiable des données du processus d'exportation au processus de collecte.

10.4.3 MTU

Comme TCP offre un service de flux au lieu d'un service de datagrammes ou de paquets séquencés, les messages IPFIX transportés sur TCP sont plutôt séparés en utilisant le champ Longueur dans l'en-tête de message IPFIX. Le processus d'exportation peut choisir toute longueur valide pour les messages IPFIX exportés, car TCP prend en charge la segmentation.

Le processus d'exportation peut choisir des longueurs de message IPFIX inférieures au maximum afin d'assurer l'exportation en temps utile des enregistrements de données.

10.4.4 Établissement et fermeture de connexion

Le processus d'exportation IPFIX initie une connexion TCP au processus de collecte. Un processus d'exportation PEUT prendre en charge plus d'une connexion active aux différents processus de collecte (incluant le cas de différents processus de collecte sur le même hôte). Un processus d'exportation PEUT prendre en charge plus d'une connexion active au même processus de collecte pour éviter le blocage de tête de ligne à travers les domaines d'observation.

L'exportateur PEUT enregistrer une alarme si la temporisation de l'établissement de la connexion TCP sous-jacente arrive à expiration ; cette temporisation devrait être configurable chez l'exportateur.

Quand un processus d'exportation est fermé, il DEVRAIT fermer la connexion TCP. Quand un processus de collecte ne veut plus recevoir de messages IPFIX, il DEVRAIT fermer son extrémité de la connexion. Le processus de collecte DEVRAIT continuer de lire les messages IPFIX jusqu'à ce que le processus d'exportation ait clos son extrémité.

Quand un processus de collecte détecte que la connexion TCP au processus d'exportation s'est terminée de façon anormale, il DOIT continuer d'écouter pour une nouvelle connexion.

Quand un processus d'exportation détecte que la connexion TCP au processus de collecte s'est terminée anormalement, il DEVRAIT essayer de rétablir la connexion. Les temporisations de connexion et les programmations de réessais DEVRAIENT être configurables. Dans la configuration par défaut, un processus d'exportation NE DOIT PAS tenter d'établir une connexion plus fréquemment qu'une fois par minute.

10.4.5 Reprise sur défaillance

Si le processus de collecte n'accuse pas réception d'une tentative du processus d'exportation d'établir une connexion, TCP va automatiquement réessayer l'établissement de connexion en utilisant un retard exponentiel. L'exportateur PEUT enregistrer une alarme si la temporisation de l'établissement de la connexion TCP sous-jacente arrive à expiration ; cette temporisation devrait être configurable chez l'exportateur.

Le processus d'exportation PEUT ouvrir à l'avance une connexion TCP de sauvegarde avec le processus de collecte, si il prend en charge la reprise sur défaillance du processus de collecte.

11. Considérations sur la sécurité

Les considérations sur la sécurité du protocole IPFIX ont été déduites d'une analyse des menaces potentielles sur la sécurité, discutées dans la section des considérations sur la sécurité du document sur les exigences pour IPFIX [RFC3917]. Les exigences pour la sécurité de IPFIX sont les suivantes :

1. IPFIX doit fournir un mécanisme pour assurer la confidentialité des données IPFIX transférées d'un processus d'exportation à un processus de collecte, afin d'empêcher la divulgation des enregistrements de flux transportés via IPFIX.

2. IPFIX doit fournir un mécanisme pour assurer l'intégrité des données IPFIX transférées d'un processus d'exportation à un processus de collecte, afin d'empêcher l'injection de données incorrectes ou d'informations de contrôle (par exemple, des gabarits) ou la duplication des messages, dans un flux de messages IPFIX.

3. IPFIX doit fournir un mécanisme pour authentifier les processus IPFIX de collecte et d'exportation, pour empêcher la collecte de données par un processus d'exportation non autorisé ou l'exportation de données à un processus de collecte non autorisé.

Parce que IPFIX peut être utilisé pour collecter des informations sur l'articulation du réseau et à des fins de facturation, des attaques conçues pour embrouiller, désactiver, ou capturer des informations d'un système de collecte IPFIX peuvent être vues comme le principal objectif d'une attaque sophistiquée contre le réseau.

Un attaquant en position d'injecter de faux messages dans un flux de messages IPFIX peut affecter l'application en utilisant IPFIX (en falsifiant les données) ou le processus de collecte IPFIX lui-même (en modifiant ou révoquant des gabarits, ou en changeant les options) ; pour cette raison, l'intégrité du message IPFIX est importante.

Les messages IPFIX eux-mêmes peuvent aussi contenir des information de valeur pour un attaquant, incluant des informations sur la configuration du réseau ainsi que sur le trafic de l'utilisateur final et les données de charge utile, donc il faut veiller à confiner leur visibilité aux utilisateurs autorisés. Quand un élément d'information contenant des informations de charge utile de l'utilisateur final est exporté, il DEVRAIT être transmis au processus de collecte en utilisant un moyen qui sécurise son contenu contre l'espionnage. Les mécanismes convenables incluent d'utiliser soit une connexion directe en point à point supposée être inaccessible aux attaquants, soit un mécanisme de chiffrement. Il est de la responsabilité du processus de collecte de fournir un degré de sécurité satisfaisant pour ces données collectées, incluant, si nécessaire, le chiffrement et/ou l'anonymisation de toutes les données rapportées ; voir le paragraphe 11.8.

11.1 Applicabilité de TLS et DTLS

La sécurité de la couche transport (TLS, *Transport Layer Security*) [RFC5246] et la sécurité de la couche de transport de datagrammes (DTLS, *Datagram Transport Layer Security*) [RFC6347] ont été conçues pour fournir les assurances de confidentialité, intégrité, et authentification requises par le protocole IPFIX, sans avoir besoin de clés pré-partagées.

Les processus d'exportation et de collecte IPFIX qui utilisent TCP DOIVENT prendre en charge TLS version 1.1 et DEVRAIT prendre en charge TLS version 1.2 [RFC5246], incluant les suites de chiffrement obligatoires spécifiées dans chaque version. Les processus d'exportation et les processus de collecte IPFIX qui utilisent UDP ou SCTP DOIVENT prendre en charge DTLS version 1.0 et DEVRAIENT prendre en charge DTLS version 1.2 [RFC6347], incluant les suites de chiffrement obligatoires spécifiées dans chaque version.

Noter que DTLS est choisi comme mécanisme de sécurité pour SCTP. Bien que les liens de TLS à SCTP soient définis dans la [RFC3436], ils exigent que toute communication soit sur des flux fiables bidirectionnels, et aussi une connexion TLS par flux. Cet arrangement n'est pas compatible avec les raisons du choix de SCTP comme protocole de transport IPFIX.

Noter que l'utilisation de DTLS a une vulnérabilité à l'attaque par interposition qui n'est pas présente dans TLS, qui permet qu'un message soit supprimé du flux sans que l'envoyeur ou le receveur le sachent. De plus, en utilisant DTLS sur SCTP, un attaquant pourrait injecter des informations de contrôle SCTP et fermer l'association SCTP, causant une perte de messages IPFIX si ces messages sont mis en mémoire tampon en dehors de l'association SCTP. Des techniques telles que celles décrites dans la [RFC6083] pourraient être utilisées pour surmonter ces vulnérabilités.

Quand il utilise DTLS sur SCTP, le processus d'exportation DOIT s'assurer que chaque message IPFIX est envoyé sur le même flux SCTP qui serait utilisé en envoyant le même message IPFIX directement sur SCTP. Noter que DTLS peut envoyer ses propres messages de contrôle sur le flux 0 avec une fiabilité complète ; cependant, cela ne va pas interférer avec le traitement des messages IPFIX du flux 0 au processus de collecte, parce que DTLS consomme ses propres messages de contrôle avant de passer les messages IPFIX à la couche d'application.

Quand on utilise DTLS sur SCTP ou UDP, l'extension Heartbeat [RFC6520] DEVRAIT être utilisée, en particulier sur des sessions de transport de longue durée, pour assurer que l'association reste active.

Les processus d'exportation et de collecte NE DOIVENT PAS demander, offrir, ou utiliser de version de la couche de connexion sécurisée (SSL, *Secure Socket Layer*) ou de version de TLS antérieure à 1.1, à cause des vulnérabilités de sécurité connues dans les versions antérieures de TLS ; voir l'Appendice E de la [RFC5246] pour plus d'informations.

11.2 Usage

Le processus d'exportation IPFIX initie la communication du processus de collecte IPFIX et agit comme un client TLS ou DTLS selon les [RFC5246] et [RFC6347], tandis que le processus de collecte IPFIX agit comme un serveur TLS ou DTLS. Le client DTLS ouvre une connexion sécurisée sur l'accès SCTP 4740 du serveur DTLS si SCTP est choisi comme protocole de transport. Le client TLS ouvre une connexon sécurisée sur l'accès TCP 4740 du serveur TLS si TCP est choisi comme protocole de transport. Le client DTLS ouvre une connexion sécurisée sur l'accès UDP 4740 du serveur DTLS si UDP est choisi comme protocole de transport.

11.3 Authentification mutuelle

Quand ils utilisent TLS ou DTLS, les processus d'exportation et de collecte IPFIX DEVRAIENT être identifiés par un certificat contenant l'identifiant DNS discuté au paragraphe 6.4 de la [RFC6125] ; l'inclusion de noms communs (CN-ID) dans les certificats qui identifient des processus d'exportation ou de collecte IPFIX N'est PAS RECOMMANDÉE.

Pour prévenir des attaques par interposition de processus d'exportation ou de collecte imposteurs, l'acceptation de données provenant de processus d'exportation non autorisés, ou l'exportation de données de processus de collecte non autorisés, l'authentification mutuelle DOIT être utilisés pour TLS et DTLS. Les processus d'exportation DOIVENT vérifier les identifiants de référence des processus de collecte auxquels ils exportent des messages IPFIX par rapport à ceux mémorisés dans les certificats. De même, les processus de collecte DOIVENT vérifier les identifiants de référence des processus d'exportation d'où ils reçoivent les messages IPFIX par rapport à ceux mémorisés dans les certificats. Les processus d'exportation NE DOIVENT PAS exporter à des processus de collecte non vérifiés, et les processus de collecte NE DOIVENT PAS accepter de messages IPFIX provenant de processus d'exportation non vérifiés.

Les processus d'exportation et les processus de collecte DOIVENT prendre en charge la vérification des certificats par rapport à une liste de certificats d'homologues explicitement autorisés identifiés par un nom commun et DEVRAIENT prendre en charge la vérification des identifiants de référence en confrontant l'identifiant DNS ou l'identifiant de nom commun avec une recherche DNS de l'homologue.

Les processus d'exportation et de collecte IPFIX DOIVENT utiliser des suites de chiffrement non NULLES pour l'authentification, l'intégrité, et la confidentialité.

11.4 Protection contre les attaques de DoS

Un attaquant peut monter une attaque de déni de service (DoS) contre un système de collecte IPFIX soit directement, en envoyant de grandes quantités de trafic à un processus de collecte, soit indirectement, en générant de grandes quantités de trafic à mesurer par un processus de mesures.

Les attaques de DoS directes peuvent aussi invoquer l'épuisement d'état, à la couche transport (par exemple, en créant un grand nombre de connexions en cours) ou dans le processus de collecte IPFIX lui-même (par exemple, en envoyant des gabarits d'enregistrements de flux en instance ou d'informations de portée, ou de grandes quantités d'enregistrements de gabarit d'options, etc.).

SCTP rend obligatoire un mécanisme d'échange de mouchards conçu pour se défendre contre les attaque de DoS par épuisement d'état SCTP. De même, TCP fournit le mécanisme de "mouchard SYN" pour atténuer l'épuisement d'état ; les mouchards SYN DEVRAIENT être utilisés par tout processus de collecte qui accepte les connexions TCP. DTLS fournit aussi un échange de mouchards pour protéger contre l'épuisement d'état du serveur DTLS.

Le lecteur devrait noter qu'il n'y a pas de moyen d'empêcher le traitement d'un message IPFIX falsifié (et la création d'état) pour la communication UDP et SCTP. L'utilisation de TLS et DTLS peut évidemment empêcher la création de faux états, mais elle est elle-même encline aux attaques d'épuisement d'état. Donc, la limitation du taux de collecte DEVRAIT être utilisée pour protéger TLS et DTLS (comme de limiter le nombre de nouvelles sessions TLS ou DTLS par seconde à un nombre raisonnable).

Les attaques en épuisement d'état IPFIX peuvent être atténuées en limitant le taux d'ouverture de nouvelles connexions ou associations par le processus de collecte, en limitant le taux auquel les messages IPFIX vont être acceptés par le processus de collecte, et en limitant de façon adaptative la quantité d'état conservé, en particulier pour les enregistrements en attente d'un gabarit. Ces limites de taux et d'état PEUVENT être fournies par un processus de collecte, et si elles sont fournies, les limites DEVRAIENT être configurables par l'utilisateur.

De plus, un processus de collecte IPFIX peut éliminer le risque d'attaques d'épuisement d'état provenant de nœuds non fiables en exigeant l'authentification mutuelle TLS ou DTLS, causant l'acceptation par le processus de collecte des seuls messages IPFIX provenant de sources de confiance.

À l'égard du déni de service indirect, le comportement de IPFIX dans des conditions de surcharge dépend du protocole de transport utilisé. Pour IPFIX sur TCP, le contrôle d'encombrement TCP va causer le retard du flux de messages IPFIX et finalement son arrêt, aveuglant le système IPFIX. SCTP améliore un peu cette situation, car certains messages IPFIX vont continuer d'être reçus par le processus de collecte du fait de l'évitement du blocage de tête de ligne par les caractéristiques de flux multiples et de fiabilité partielle de SCTP, permettant éventuellement une certaine visibilité de l'attaque. La situation est similaire avec UDP, car certains datagrammes peuvent continuer d'être reçus au processus de collecte, appliquant effectivement l'échantillonnage au flux de messages IPFIX et impliquant que des informations sur l'attaque vont être disponibles.

Pour minimiser la perte de messages IPFIX dans des conditions de surcharge, certains mécanismes de service différencié pourraient être utilisés pour prioriser le trafic IPFIX par rapport aux autres trafics sur la même liaison. Autrement, les messages IPFIX peuvent être transportés sur un réseau dédié. Dans ce cas, il faut veiller à s'assurer que le réseau dédié peut traiter le trafic de pointe de messages IPFIX attendu.

11.5 Quand DTLS ou TLS n'est pas une option

L'utilisation de DTLS ou TLS pourrait n'être pas possible dans certains cas, du fait des problèmes de performances ou autres questions de fonctionnement.

Sans l'authentification mutuelle TLS ou DTLS, les processus d'exportation et de collecte IPFIX peuvent revenir à l'utilisation des adresses IP de source pour authentifier leurs homologues. Une politique d'allocation des adresses IP des processus d'exportation et de collecte à partir des gammes d'adresses spécifiées, et en utilisant le filtrage d'entrée pour empêcher l'usurpation d'identité, peut améliorer l'utilité de cette approche. Là encore, la ségrégation complète du trafic IPFIX sur un réseau dédié, lorsque possible, peut améliorer encore la sécurité. Dans tous les cas, l'utilisation de processus de collecte ouverts (ceux qui vont accepter des messages IPFIX de tout processus d'exportation sans égard à l'adresse IP ou à l'identité) est déconseillée.

Les mises en œuvre modernes de TCP et SCTP sont résistantes aux attaques d'insertion aveugle (voir les [RFC4960] et [RFC6528]) ; cependant, UDP n'offre pas une telle protection. Pour cette raison, le trafic de messages IPFIX transporté via UDP et non sécurisé via DTLS DEVRAIT être protégé via la ségrégation sur un réseau dédié.

11.6 Enregistrement d'une attaque contre IPFIX

Les processus de collecte IPFIX DOIVENT détecter une potentielle insertion de message IPFIX ou des conditions de pertes en suivant les numéros de séquence IPFIX et DEVRAIENT fournir un mécanisme d'enregistrement pour rapporter les messages hors séquence. Noter qu'un attaquant peut être capable d'exploiter le traitement des messages hors séquence au processus de collecte, donc on devrait faire attention lors du traitement de ces conditions. Par exemple, un processus de collecte qui réinitialise simplement le numéro de séquence attendu à réception du numéro de séquence en retard pourrait être temporairement aveuglé par une injection délibérée de numéros de séquence en retard.

Les processus d'exportation et de collecte IPFIX DEVRAIENT enregistrer toute tentative de connexion qui échoue suite à un échec d'authentification, due à la présentation d'un certificat non autorisé ou discordant durant l'authentification mutuelle TLS ou DTLS, ou due à une tentative de connexion provenant d'une adresse IP non autorisée quand TLS ou DTLS n'est pas utilisé.

Les processus d'exportation et de collecte IPFIX DEVRAIENT détecter et enregistrer toute réinitialisation d'association SCTP ou de connexion TCP.

11.7 Sécurisation du collecteur

La sécurité du collecteur et de sa mise en œuvre est importante pour réaliser la sécurité globale ; cependant, un ensemble complet de lignes directrices pour la sécurité d'une mise en œuvre de collecteur sort du domaine du présent document.

Comme IPFIX utilise des codages de préfixe de longueur, les mises en œuvre de collecteur devraient veiller à s'assurer de la détection des valeurs incohérentes qui pourraient impacter le décodage de message IPFIX, et le bon fonctionnement en présence de telles valeurs incohérentes.

Précisément, la cohérence des longueurs de message IPFIX, d'ensemble, et d'élément d'information de longueur variable doit être vérifiée pour éviter les vulnérabilités liées à la taille de mémoire tampon.

Les mises en œuvre de collecteur devraient aussi porter une attention particulière au codage UTF-8 des types de données de chaîne, car des vulnerabilités peuvent exister dans l'interprétation de valeurs UTF-8 mal formées ; voir le paragraphe 6.1.6.

11.8 Considérations de confidentialité pour les données collectées

Les données de flux exportées par le processus d'exportation et collectées par le processus de collecte contiennent normalement des informations sur le trafic du réseau observé. Ces informations peuvent être personnellement identifiables et sensibles pour la vie privée. La mémorisation de ces données doit être protégée via des moyens techniques ainsi que de politique pour assurer que la vie privée des utilisateurs du réseau mesuré est protégée. Une spécification complète de ces moyens sort du domaine d'application de ce document et est spécifique de l'application et de la technologie de mémorisation utilisée.

12. Considérations de gestion

La [RFC6615] a spécifié un module de MIB qui définit des objets gérés pour surveiller les appareils IPFIX, incluant de configuration de base. Cette MIB peut être utilisée pour mesurer l'impact de l'exportation IPFIX sur le réseau de surveillance ; elle contient des tableaux couvrant :

la session de transport,
la définition de l'antémémoire,
la définition du point d'observation,
la définition du gabarit et du gabarit d'options,
les caractéristiques de l'exportation (reprise sur défaillance, équilibrage de charge, duplication) et
les statistiques de l'exportation par processus, session, et gabarit.

Du point de vue du fontionnement, une importante fonction de ce module de MIB est fournie par le tableau statistique de session de transport, qui contient le taux (en octets par seconde) auquel le collecteur reçoit ou l'exportateur envoie les messages IPFIX. D'un intérêt particulier pour les opérations, le tableau statistique de session de transport du paragraphe 5.8.1 de ce module de MIB expose le taux de collecte ou d'export des messages IPFIX, qui permet la mesure de la bande passante utilisée par l'exportation IPFIX.

La [RFC6727] décrit des extensions au module de MIB IPFIX-SELECTOR spécifié dans la [RFC6615] et contient les objets gérés pour fournir des informations sur les fonctions de choix de paquet appliquées et leurs paramètres (filtrage et échantillonnage).

Comme les modules IPFIX-SELECTOR-MIB [RFC6615] et PSAMP-MIB [RFC6727] contiennent seulement des objets en lecture seule, ils ne peuvent pas être utilisés pour la configuration des appareils IPFIX. La [RFC6728] spécifie un modèle de données de configuration pour les protocoles IPFIX et PSAMP, en utilisant le protocole de configuration de réseau (NETCONF, *Network Configuration Protocol*). Ce modèle de données couvre les processus de choix, les antémémoires, les processus d'exportation, et les processus de collecte sur les appareils IPFIX et PSAMP, et est défini en utilisant les diagrammes de classe du langage de modélisation unifié (UML, *Unified Modeling Language*) et est formellement spécifié en utilisant YANG. Les données de configuration sont codées en langage de balisage extensible (XML, *Extensible Markup Language*).

Quelques mécanismes spécifiés à côté du protocole IPFIX peuvent aider à surveiller et réduire la bande passante utilisée pour l'exportation IPFIX :
- une méthode d'économie de la bande passante pour exporter les informations redondantes dans IPFIX [RFC5473]
- une méthode efficace pour exporter les flux bidirectionnels [RFC5103]
- une méthode pour définir et exporter des structures de données complexes [RFC6313].

Autrement, PSAMP [RFC5474] peut être utilisé pour exporter des paquets échantillonnés par des méthodes statistiques et autres, qui peuvent être applicables à de nombreux domaines de surveillance pour lesquels IPFIX est aussi adapté. PSAMP fournit aussi le contrôle de l'impact sur le réseau mesuré par son taux d'échantillonnage. L'ensemble de techniques de choix de paquet (échantillonnage, filtrage, et hachage) normalisées par PSAMP est décrit dans la [RFC5475]. PSAMP définit aussi une limite de taux d'exportation explicitement configurable au paragraphe 8.4 de la [RFC5474].

13. Considérations relatives à l'IANA

IANA a mis à jour le registre "Éléments d'information IPFIX" [IANA-IPFIX] afin que toutes les références qui pointaient précédemment sur la RFC 5101 pointent maintenant sur le présent document.

Les messages IPFIX utilisent deux champs avec des valeurs allouées. Ce sont le numéro de version IPFIX, qui indique quelle version du protocole IPFIX a été utilisée pour exporter un message IPFIX, et l'identifiant d'ensemble IPFIX, qui indique le type de chaque ensemble d'informations au sein d'un message IPFIX.

Les éléments d'information utilisés par IPFIX, et les sous registres des valeurs d'élément d'information, sont gérés par l'IANA [IANA-IPFIX], comme le sont les numéros d'entreprise privée utilisés par les éléments d'information spécifiques d'entreprise [IANA-PEN]. Le présent document ne fait pas de changement à ces registres.

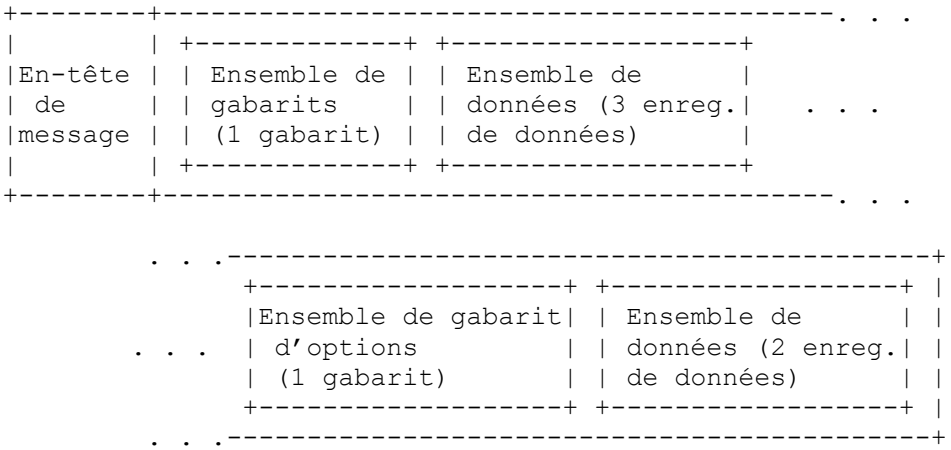
La valeur de numéro de version IPFIX de 0x000a (10) est réservée pour le protocole IPFIX spécifié dans le présent document. Les valeurs d'identifiant d'ensemble de 0 et 1 ne sont pas utilisées, pour des raisons historiques [RFC3954]. La valeur d'identifiant d'ensemble de 2 est réservée pour l'ensemble de gabarit. La valeur d'identifiant d'ensemble de 3 est réservée pour l'ensemble de gabarits d'options. Toutes les autres valeurs d'identifiant d'ensemble de 4 à 255 sont réservées pour une future utilisation. Les valeurs d'identifiant d'ensemble au dessus de 255 sont utilisées pour les ensembles de données.

De nouvelles allocations dans les sous registres "Numéro de version IPFIX" ou "Identifiants d'ensemble IPFIX" exigent une action de normalisation [RFC5226], c'est-à-dire, qu'elles sont faites par une RFC sur la voie de la normalisation approuvée par l'IESG.

Appendice A. Exemples de codage IPFIX

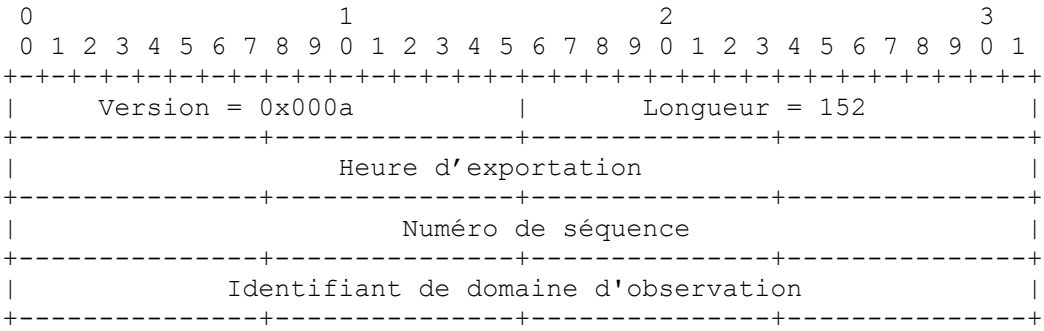
Cet appendice, qui n'est pas normatif, contient des exemples de codage IPFIX.
Considérons l'exemple d'un message IPFIX composé d'un ensemble de gabarits, d'un ensemble de données (qui contient trois enregistrements de données) d'un ensemble de gabarits d'options, et d'un autre ensemble de données (qui contient deux enregistrements de données relatifs au précédent enregistrement de gabarit d'options).

Message IPFIX :



A.1 Exemples d'en-tête de message

L'en-tête de message est composé de :

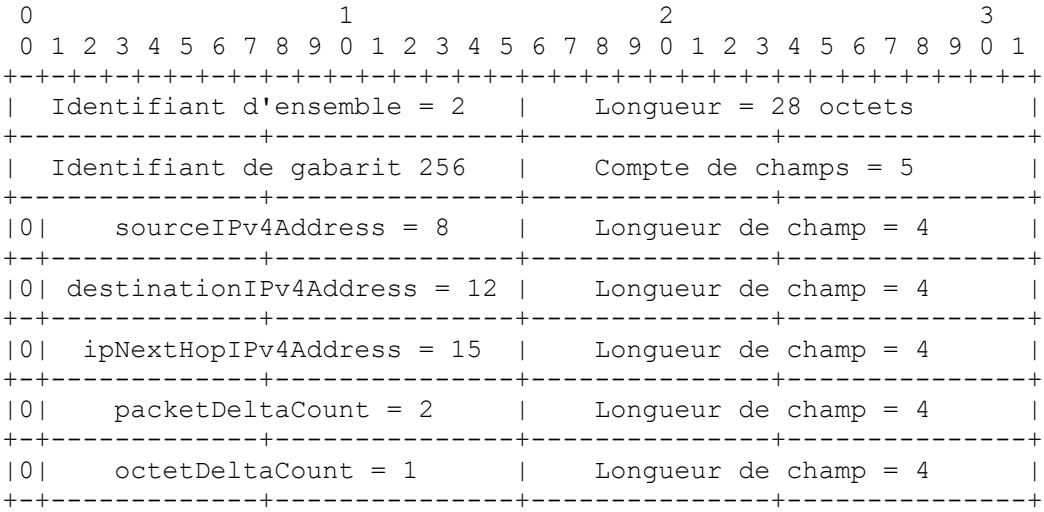


A.2 Exemples d'ensemble de gabarits

A.2.1 Ensemble de gabarit utilisant des éléments d'information de l'IANA

On veut rapporter les éléments d'information suivants :
- Adresse IPv4 de source : sourceIPv4Address [IANA-IPFIX], avec une longueur de 4 octets
- Adresse IPv4 de destination : destinationIPv4Address [IANA-IPFIX], avec une longueur de 4 octets
- Adresse IP de prochain bond (IPv4) : ipNextHopIPv4Address [IANA-IPFIX], avec une longueur de 4 octets
- Nombre de paquets du flux : packetDeltaCount [IANA-IPFIX], avec une longueur de 4 octets
- Nombre d'octets du flux : octetDeltaCount [IANA-IPFIX], avec une longueur de 4 octets

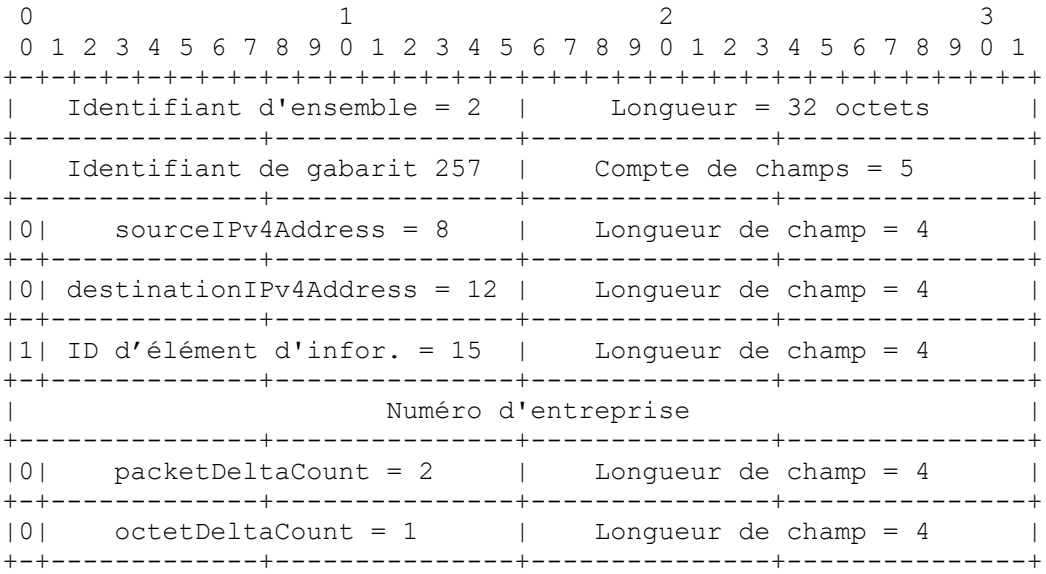
Donc, l'ensemble de gabarit va être composé comme suit :



A.2.2 Ensemble de gabarits utilisant des éléments d'information spécifiques de l'entreprise

On veut rapporter les éléments d'information suivants :
- Adresse IPv4 de source : sourceIPv4Address [IANA-IPFIX], avec une longueur de 4 octets
- Adresse IPv4 de destination : destinationIPv4Address [IANA-IPFIX], avec une longueur de 4 octets
- Un élément d'information spécifique d'entreprise représentant des informations propriétaires, avec un type de 15 et une longueur de 4 octets
- Nombre de paquets du flux : packetDeltaCount [IANA-IPFIX], avec une longueur de 4 octets
- Nombre d'octets du flux : octetDeltaCount [IANA-IPFIX], avec une longueur de 4 octets

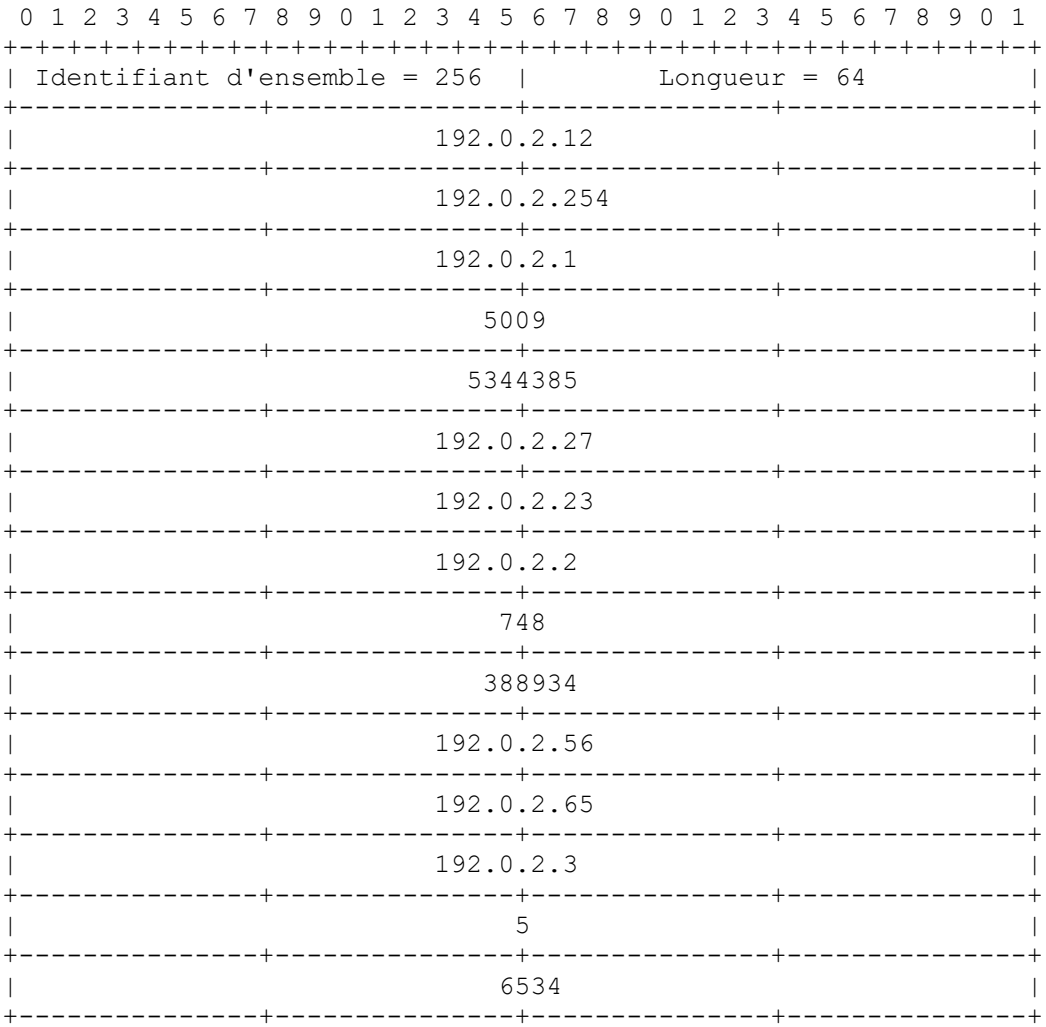
Donc, l'ensemble de gabarit va être composé comme suit :



A.3 Exemple d'ensemble de données

Dans cet exemple, on rapporte les trois enregistrements de flux suivants :

Adr. IP de source	Adr. IP de dst.	Adr. de proch. bond	Nb de paquets	Nb d'octets
192.0.2.12	192.0.2.254	192.0.2.1	5009	5344385
192.0.2.27	192.0.2.23	192.0.2.2	748	388934
192.0.2.56	192.0.2.65	192.0.2.3	5	6534



Noter que le bourrage n’est pas nécessaire dans cet exemple.

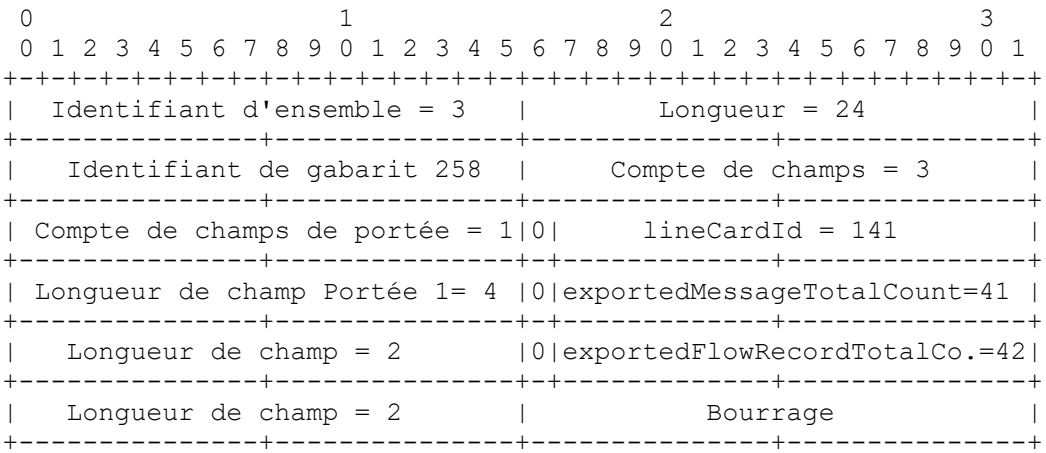
A.4 Exemples d’ensemble de gabarits d'options

A.4.1 Ensemble de gabarits d'options utilisant des éléments d'information de l’IANA

Par carte de ligne (le routeur étant composé de deux certes de ligne) on veut rapporter les éléments d'information suivants :
- Nombre total de messages IPFIX : exportedMessageTotalCount [IANA-IPFIX], avec une longueur de 2 octets
- Nombre total de flux exportés : exportedFlowRecordTotalCount [IANA-IPFIX], avec une longueur de 2 octets

La carte de ligne, qui est représentée par l’élément d'information lineCardId [IANA-IPFIX], est utilisée comme champ de portée.

Donc, l’ensemble de gabarits d'options va être :

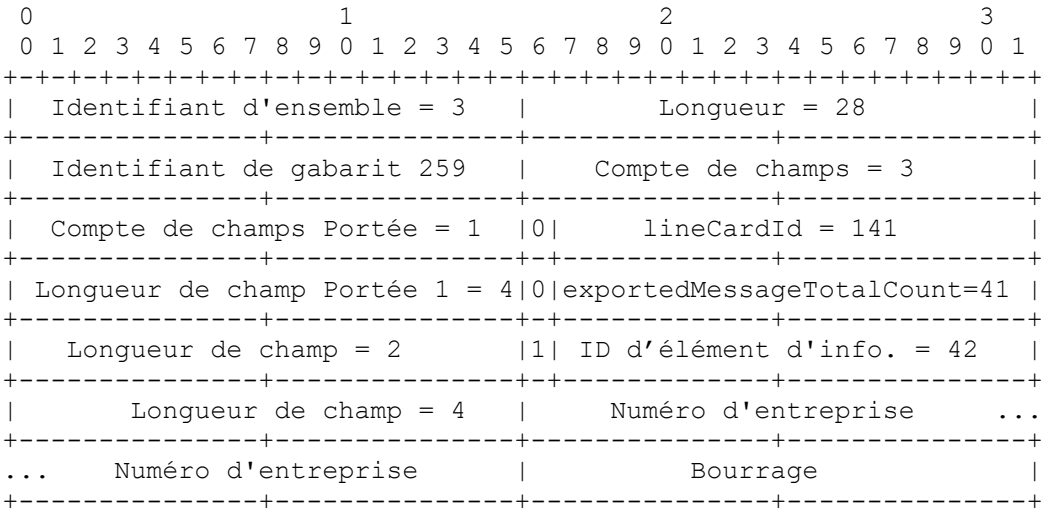


A.4.2 Ensemble de gabarits d'options utilisant des éléments d'information spécifiques de l’entreprise

Par carte de ligne (le routeur étant composé de deux cartes de ligne) on veut rapporter les éléments d'information suivants :
- Nombre total de messages IPFIX : exportedMessageTotalCount [IANA-IPFIX], avec une longueur de 2 octets
- un nombre de flux spécifiques d'entreprise exportés, avec un type de 42 et une longueur de 4 octets.

La carte de ligne, qui est représentée par l’élément d'information lineCardId [IANA-IPFIX], est utilisée comme champ de portée.

Le format de l’ensemble de gabarits d'options est comme suit :



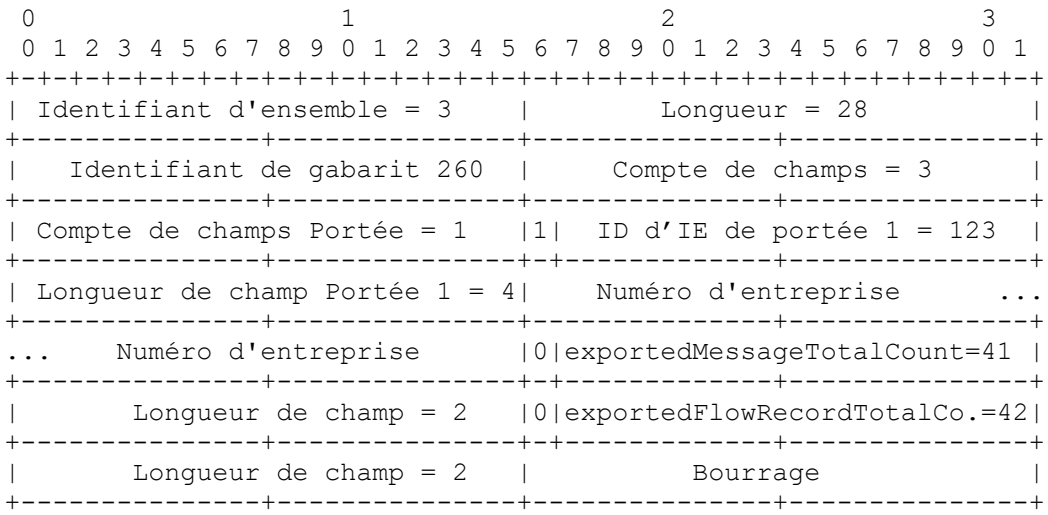
A.4.3 Ensemble de gabarits d'options utilisant une portée spécifique de l'entreprise

Dans cet exemple, on veut exporter les mêmes informations que dans l’exemple du paragraphe A.4.1 :

- Nombre total de messages IPFIX : exportedMessageTotalCount [IANA-IPFIX], avec une longueur de 2 octets
- Nombre total de flux exportés : exportedFlowRecordTotalCount [IANA-IPFIX], avec une longueur de 2 octets

Mais cette fois, les informations relèvent d'une portée propriétaire, identifiée par le numéro d'élément d'information spécifique d'entreprise 123.

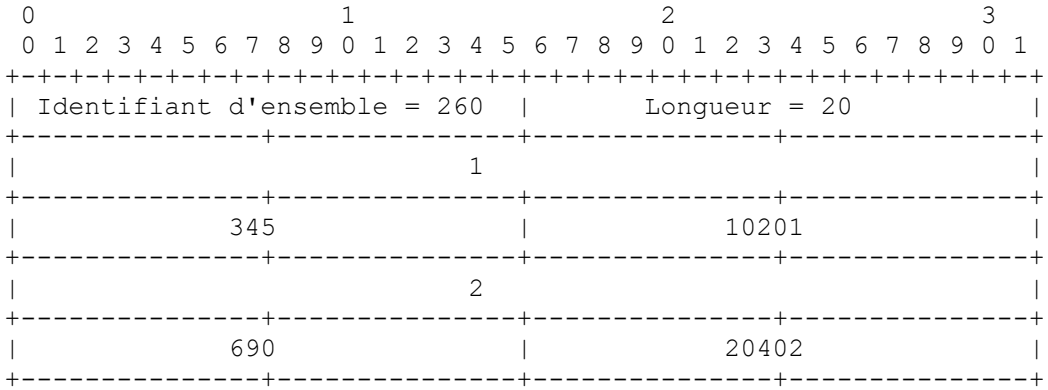
Le format de l'ensemble de gabarits d'options est maintenant comme suit :



A.4.4 Ensemble de données utilisant une portée spécifique de l'entreprise

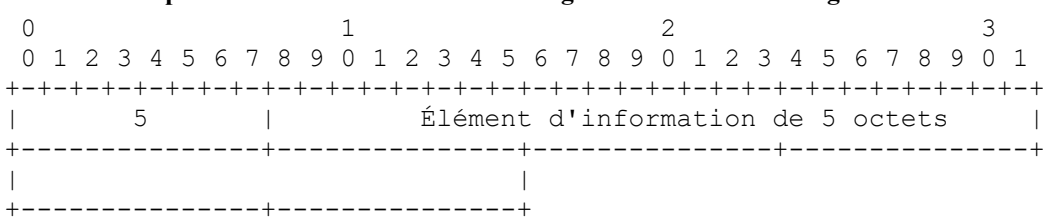
Dans cet exemple, on rapporte les deux enregistrements de données suivants :

Champ Entreprise 123	Message IPFIX	Enregistrements de flux exporté
1	345	10 201
2	690	20 402

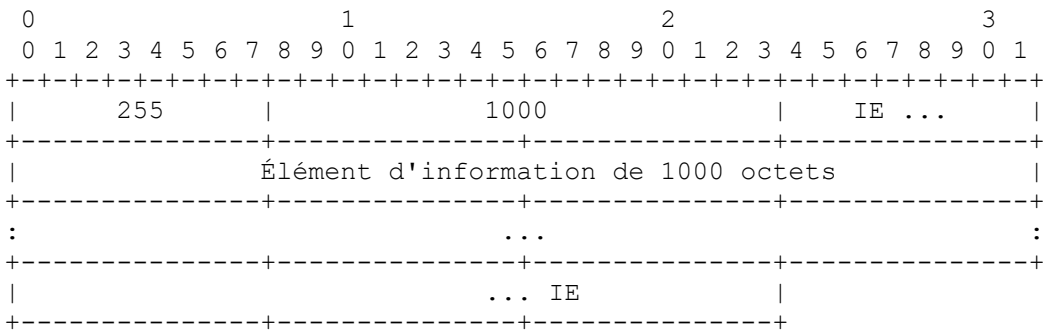


A.5 Exemples d'élément d'information de longueur variable

A.5.1 Exemple d'élément d'information de longueur variable avec longueur de moins de 255 octets



A.5.2 Exemple d'élément d'information de longueur variable avec codage de longueur de 3 octets



Références normatives

[IANA-IPFIX] IANA, "IP Flow Information Export (IPFIX) Entities", <http://www.iana.org/assignments/ipfix/>.

[RFC0768] J. Postel, "Protocole de datagramme d'utilisateur (UDP)", (STD 6), 28 août 1980.

[RFC0793] J. Postel (éd.), "Protocole de commande de transmission – Spécification du protocole du programme Internet DARPA", STD 7, septembre 1981. (*Remplacée par* RFC9293)

[RFC1014] Sun microsystems, "XDR : norme de représentation de données externes", juin 1987.

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. *(MàJ par* RFC8174)

[RFC3436] A. Jungmaier, E. Rescorla, M. Tuexen, "Sécurité de la couche Transport sur le protocole de transmission de contrôle de flux", décembre 2002. *(P.S.)*

[RFC3629] F. Yergeau, "UTF-8, un format de transformation de la norme ISO 10646", STD 63, novembre 2003.

[RFC3758] R. Stewart et autres, "Extension de fiabilité partielle du protocole de transmission de contrôle de flux (SCTP)", mai 2004. *(P.S.)*

[RFC4960] R. Stewart, éd., "Protocole de transmission de commandes de flux (SCTP)", septembre 2007. (*Remplace* RFC2960, RFC3309 ; *P.S. ; Remplacée par* RFC9260)

[RFC5226] T. Narten et H. Alvestrand, "Lignes directrices pour la rédaction d'une section Considérations relatives à l'IANA dans les RFC", BCP 26, mai 2008. (*Remplace* RFC2434 *; remplacée par* RFC8126)

[RFC5246] T. Dierks, E. Rescorla, "Version 1.2 du protocole de sécurité de la couche Transport (TLS)", août 2008. *(P.S.* ; *remplace* RFC3268, 4346, 4366 ; *MàJ* RFC4492 ; *rendue obsolète par la* RFC8446)

[RFC5905] D. Mills, J. Martin, J. Burbank, W. Kasch, "Protocole de l'heure du réseau version 4 (NTPv4) : Spécification du protocole et des algorithmes ", juin 2010. (*Remplace* RFC1305, RFC4330) *(P. S ; MàJ par* RFC7822, 8573, 9109)

[RFC6125] P. Saint-André, J. Hodges, "Représentation et vérification d'identité de service d'application fondé sur le domaine au sein de l'infrastructure Internet de clé publique utilisant les certificats X.509 (PKIX) dans le contexte de la sécurité de la couche Transport (TLS)", mars 2011. *(P.S.)*

[RFC6347] E. Rescorla, N. Modadugu, "Sécurité de la couche transport de datagrammes, version 1.2", janvier 2012. *(P.S. ;Remplace la* RFC4347 ; *MàJ par* RFC7905, 9146) ; *Remplacée par* RFC9147)

[RFC6520] R. Seggelmann, M. Tuexen, M. Williams, "Extension Heartbeat à la sécurité de la couche Transport (TLS) et à la sécurité de la couche transport de datagrammes (DTLS)", février 2012. *(P.S.)*

[RFC7012] B. Claise, B. Trammell, "Modèle d'information pour l'exportation d'informations de flux IP (IPFIX)", septembre 2013. (*Remplace* RFC5102) *(P.S.)*

Références pour information

[IEEE.754.2008] Institute of Electrical and Electronics Engineers, "IEEE Standard for Floating-Point Arithmetic", IEEE Standard 754, août 2008.

[IANA-PEN] IANA, "Private Enterprise Numbers", <http://www.iana.org/assignments/enterprise-numbers/>.

[POSIX.1] IEEE 1003.1-2008, "IEEE Standard for Information Technology - Portable Operating System Interface (POSIX(R))", 2008.

[RFC2579] K. McCloghrie, D. Perkins, J. Schoenwaelder, "Conventions textuelles pour SMIv2", avril 1999. (STD0058)

[RFC3550] H. Schulzrinne, S. Casner, R. Frederick et V. Jacobson, "RTP : un protocole de transport pour les applications en temps réel", STD 64, juillet 2003. *(MàJ par* RFC7164, RFC7160, RFC8083, RFC8108, RFC8860)

[RFC3917] J. Quittek, T. Zseby, B. Claise, S. Zander, "Exigences pour l'exportation d'informations de flux IP (IPFIX)", octobre 2004. *(Information)*

[RFC3954] B. Claise, éd., "Format d'exportation de données pour la version 9 des services NetFlow de Cisco Systems", octobre 2004. *(Information)*

[RFC5101] B. Claise, éd., "Spécification du protocole d'exportation d'informations de flux IP (IPFIX) pour l'échange d'informations de flux de trafic IP", janvier 2008. *(P.S.) (Obsolète, voir* RFC7011, STD77)

[RFC5103] B. Trammell, E. Boschi, "Exportation de flux bidirectionnels en utilisant l'exportation d'informations de flux IP (IPFIX)", janvier 2008. *(P.S.)*

[RFC5153] E. Boschi et autres, "Lignes directrices pour la mise en œuvre de IPFIX", avril 2008. *(Information)*

[RFC5470] G. Sadasivan et autres, "Architecture pour l'exportation d'informations de flux IP", mars 2009. *(Information)*

[RFC5471] C. Schmoll, P. Aitken et B. Claise,"Lignes directrices pour l'essai d'exportation d'informations de flux IP (IPFIX)", mars 2009. *(Information)*

[RFC5472] T. Zseby et autres, "Applicabilité de l'exportation d'information de flux IP (IPFIX)", mars 2009. *(Information)*

[RFC5473] E. Boschi et autres, "Réduction de redondance dans les rapports d'exportation d'informations de flux IP (IPFIX) et d'échantillonnage de paquet (PSAMP)", mars 2009. *(Information)*

[RFC5474] N. Duffield et autres, "Cadre de travail pour la sélection et le rapport de paquet", mars 2009. *(Information)*

[RFC5475] T. Zseby et autres, "Techniques d'échantillonnage et de filtrage pour sélection de paquet IP", mars 2009. *(P.S.)*

[RFC5476] B. Claise et autres "Spécification du protocole d'échantillonnage de paquet (PSAMP)", mars 2009. *(P.S.)*

[RFC5477] T. Dietz et autres, "Modèle d'information pour l'exportation d'échantillonnage de paquet", mars 2009. *(P.S.)*

[RFC5610] E. Boschi, B. Trammell, L. Mark, T. Zseby, "Exportation des informations de type pour les éléments d'information Exportation d'information de flux IP (IPFIX)", juillet 2009. *(P. S.)*

[RFC5655] B. Trammell, E. Boschi, L. Mark, T. Zseby, A. Wagner, "Spécification du format de fichier d'exportation d'informations de flux IP (IPFIX)" octobre 2009. *(P. S.)*

[RFC6083] M. Tüexen, R. Seggelmann et E. Rescorla, "Sécurité de la couche Transport de datagrammes (DTLS) pour le protocle de transmission de contrôle de flux (SCTP)", janvier 2011. *(P.S.)*

[RFC6183] A. Kobayashi et autres, "Cadre de médiation pour l'exportation d'informations sur les flux IP (IPFIX)", avril

2011. *(MàJ la* RFC5470) *(Information)*

[RFC6313] B. Claise, G. Dhandapani, P. Aitken, S. Yates, "Exportation de données structurées dans l'exportation d'informations de flux IP (IPFIX)", juillet 2011. *(MàJ la* RFC5102) *(P.S.)*

[RFC6526] B. Claise et autres, "Exportation des informations de flux IP (IPFIX) par flux du protocole de transmission de contrôle de flux (SCTP)", mars 2012. *(P.S.)*

[RFC6528] F. Gont, S. Bellovin, "Se défendre contre les attaques de numéro de séquence", février 2012. (*Remplace la* RFC1948) (*MàJ la* RFC0793) *(P.S. ; remplacée par* RFC9293*)*

[RFC6615] T. Dietz et autres, "Définitions des objets gérés pour l'exportation d'informations de flux IP", juin 2012. (*Remplace la* RFC5815) *(P.S.)*

[RFC6727] T. Dietz, B. Claise, J. Quittek, "Définitions des objets gérés pour l'échantillonnage de paquet", octobre 2012. *(P.S.)*

[RFC6728] G. Muenz, B. Claise, P. Aitken, "Modèle de données de configuration pour les protocoles d'exportation d'informations de flux IP (IPFIX) et l'échantillonnage de paquet (PSAMP)", octobre 2012. *(P.S.)*

[RFC7119] B. Claise, A. Kobayashi, B. Trammell, "Fonctionnement du protocole d'exportation d'information de flux IP (IPFIX) sur des médiateurs IPFIX", février 2014. *(P.S.)*

[UTF8-EXPLOIT] Davis, M. and M. Suignard, "Unicode Technical Report #36: Unicode Security Considerations", The Unicode Consortium, juillet 2012.

Remerciements

Nous tenons à remercier Ganesh Sadasivan de sa contribution significative durant les phases initiales de la spécification du protocole. Des remerciements vont également à Juergen Quittek pour la coordination entre IPFIX et PSAMP, à Nevil Brownlee, Dave Plonka, et Andrew Johnson pour leurs relectures, à Randall Stewart et Peter Lei pour leur expertise de SCTP et leurs contributions, à Martin Djernaes pour le premier essai sur la section SCTP, à Michael Behringer et Eric Vyncke pour leurs avis et apports sur la sécurité, à Michael Tüexen pour son aide sur la section DTLS, à Elisa Boschi pour sa contribution sur l'amélioration de la section SCTP, à Mark Fullmer, Sebastian Zander, Jeff Meyer, Maurizio Molina, Carter Bullard, Tal Givoly, Lutz Mark, David Moore, Robert Lowe, Paul Calato, Andrew Feren, Gerhard Muenz, Sue Hares, et de nombreux autres, pour les relectures techniques et leurs réactions. Finalement, une mention particulière à Adrian Farrel pour son attention aux aspects de gestion et de fonctionnement.

Contributeurs

Stewart Bryant
Cisco Systems
10 New Square, Bedfont Lakes
Feltham, Middlesex TW18 8HA
United Kingdom
mél : stbryant@cisco.com

Simon Leinen
SWITCH
Werdstrasse 2
P.O. Box 8021
Zurich
Switzerland
mél : simon.leinen@switch.ch

Thomas Dietz
NEC Laboratories Europe
Network Research Division
Kurfuersten-Anlage 36
69115 Heidelberg
Germany
mél : Thomas.Dietz@nw.neclab.eu

Adresse des auteurs

Benoît Claise
Cisco Systems, Inc.
De Kleetlaan 6a b1
1831 Diegem
Belgium
téléphone : +32 2 704 5622
mél : bclaise@cisco.com

Brian Trammell
Swiss Federal Institute of Technology
Gloriastrasse 35
8092 Zurich
Switzerland
téléphone : +41 44 632 70 13
mél : trammell@tik.ee.ethz.ch

Paul Aitken
Cisco Systems, Inc.
96 Commercial Quay
Commercial Street, Edinburgh EH6 6LX
United Kingdom
téléphone : +44 131 561 3616
mél : paitken@cisco.com