Groupe de travail Réseau
Request for Comments : 5424
RFC rendue obsolète : 3164
Catégorie : Sur la voie de la normalisation

R. Gerhards, Adiscon GmbH
mars 2009

Traduction Claude Brière de L'Isle

Protocole Syslog

Résumé
Le présent document décrit le protocole syslog, qui est utilisé pour porter les messages de notification d'événement. Ce protocole utilise une architecture en couches, qui permet l'utilisation d'un nombre quelconque de protocoles de transport pour la transmission des messages syslog. Il fournit aussi un format de message qui permet que des extensions spécifiques de fabricant soient fournies de façon structurée.

Le présent document a été écrit en visant les buts de conception originaux du syslog traditionnel. Le besoin d'une nouvelle spécification est apparu parce que les efforts de normalisation d'extensions syslog fiables et sécurisées souffrent du manque de RFC sur la voie de la normalisation et indépendantes du transport. Sans ce document, chaque autre norme doit définir ses propres formats de paquet syslog et mécanismes de transport, qui au fil du temps introduiraient de subtils problèmes de compatibilité. Le présent document essaye de construire des fondations sur lesquelles les extensions syslog puissent s'appuyer. Cette approche d'une architecture en couches donne aussi une base solide qui permet d'écrire du code une fois pour chaque caractéristique syslog plutôt que pour chaque transport.

Le présent document rend obsolète la RFC 3164.

Table des matières

1. Introduction

Le présent document décrit une architecture en couches pour syslog. Le but de cette architecture est de séparer le contenu du message du transport du message tout en permettant une extensibilité facile pour chaque couche.

Le présent document décrit le format standard pour les messages syslog et souligne le concept de transpositions de transport. Il décrit aussi des éléments de données structurés, qui peuvent être utilisés pour transmettre des informations structurées aisément analysables, et permet des extensions de fabricant.

Le présent document ne décrit aucun format de mémorisation pour les messages syslog. Cela sort du domaine d'application du protocole syslog et est inutile pour l'interopérabilité du système.

Le présent document a été écrit en pensant aux buts originaux de la conception traditionnelle de syslog. Le besoin d'une nouvelle spécification en couches est apparu parce que les efforts de normalisation d'extensions fiables et sûres à syslog souffraient de l'absence de RFC sur la voie de la normalisation indépendante du transport. Sans ce document, chaque autre norme aurait besoin de définir son propre format de paquet syslog et son propre mécanisme de transport, ce qui au fil du temps introduirait de subtils problèmes de compatibilité. Le présent document essaye de donner des bases sur lesquelles les extensions à syslog puissent bâtir. Cette approche d'une architecture en couches donne aussi une base solide qui permet d'écrire du code une seule fois pour chaque caractéristique syslog au lieu d'une fois pour chaque transport.

Le présent document rend obsolète la RFC 3164, qui est un document d'information décrivant des mises en œuvre de ce domaine.

2. Conventions utilisées dans le document

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].

3. Définitions

Syslog utilise trois couches :
- Le "contenu syslog" est l'information de gestion contenue dans un message syslog.
- La couche "application syslog" traite la génération, l'interprétation, l'acheminement, et la mémorisation des messages syslog.
- La couche "transport syslog" met les messages sur le réseau et les sort du réseau.

Certains types de fonctions sont effectués à chaque couche conceptuelle :
- Un "générateur" génère le contenu syslog à porter dans un message.
- Un "collecteur" rassemble le contenu syslog pour analyse ultérieure.
- Un "relais" transmet les messages, accepte les messages provenant des générateurs ou des autres relais, et les envoie aux collecteurs ou autres relais.
- Un "envoyeur de transport" passe les messages syslog à un protocole de transport spécifique.
- Un "receveur de transport" prend les messages syslog provenant d'un protocole de transport spécifique.

Le diagramme 1 montre les différentes entités séparées par couche.

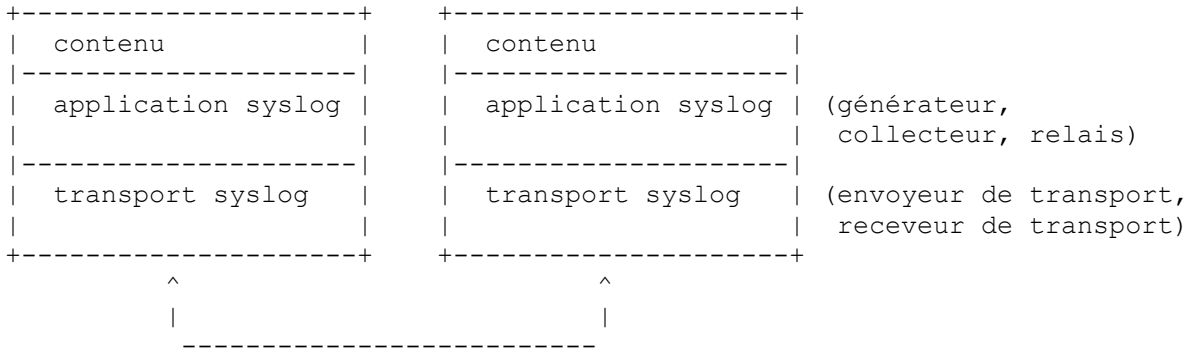

Diagramme 1 : Couches Syslog

4. Principes de base

Les principes suivants s'appliquent à la communication syslog :

- Le protocole syslog ne fournit pas d'accusé de réception de la livraison du message. Bien que certains transports puissent fournir des informations d'état, par sa conception, syslog est un protocole de pures communications unidirectionnelles.

- Les générateurs et relais peuvent être configurés à envoyer le même message à plusieurs collecteurs et relais.

- Les fonctions de générateur, relais, et collecteur peuvent résider sur le même système.

4.1 Exemple de scénarios de déploiement

Le diagramme 2 montre des scénarios de déploiement. D'autres arrangements de ces exemples sont aussi acceptables. Comme noté, dans le diagramme suivant, les relais peuvent envoyer tout ou partie des messages qu'ils reçoivent et aussi envoyer des messages qu'ils génèrent en interne. Les boîtes représentent des applications à capacité syslog.

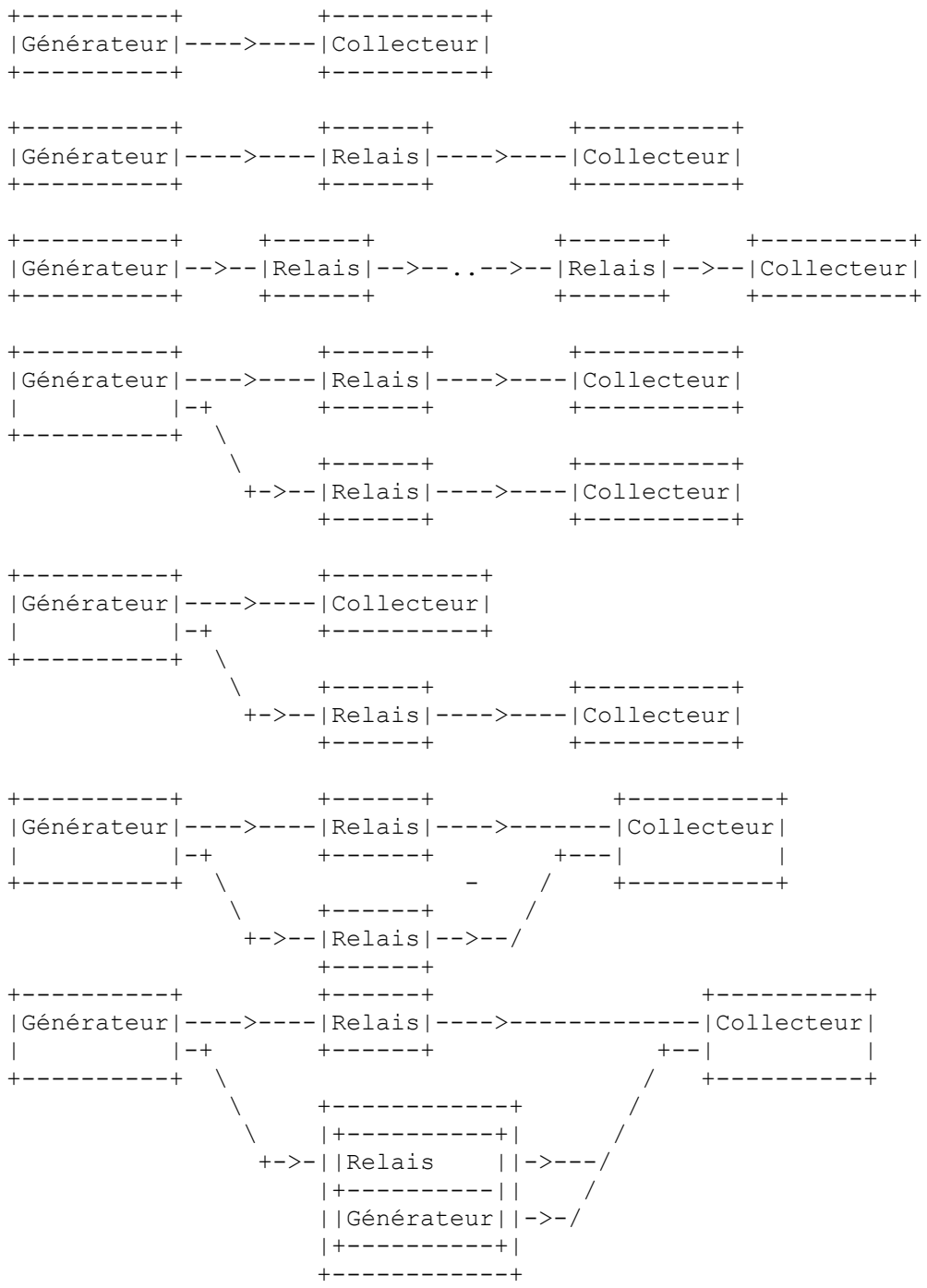

Diagramme 2 : Quelques scénarios possibles de déploiement de syslog

5. Protocole de couche transport

Le présent document ne spécifie aucun protocole de couche transport. Il décrit plutôt le format d'un message syslog d'une façon indépendante de la couche transport. Les transports syslog sont définis dans d'autres documents. Un de ces transports est défini dans la [RFC5426] et est cohérent avec le transport UDP traditionnel. Ce transport est nécessaire pour maintenir l'interopérabilité car le transport UDP a été utilisé historiquement pour la transmission des messages syslog.

Aucun protocole de transport syslog NE DOIT altérer délibérément le message syslog. Si le protocole de transport a besoin d'effectuer des transformations temporaires chez l'envoyeur de transport, ces transformations DOIVENT être inversées par le protocole de transport chez le receveur de transport afin que le relais ou collecteur voit une copie exacte du message généré par le générateur ou relais. Autrement, les vérificateurs de chiffrement de bout en bout (comme les signatures) vont

être cassés. Bien sûr, l'altération de message pourrait se produire à cause d'erreurs de transmission ou autres problèmes. La protection contre de telles altérations sort du domaine d'application de ce document.

5.1 Transposition minimum de transport requise

Toutes les mises en œuvre de cette spécification DOIVENT prendre en charge un transport fondé sur TLS comme décrit dans la [RFC5425].

Toutes les mises en œuvre de cette spécification DEVRAIENT aussi prendre en charge un transport fondé sur UDP, comme décrit dans la [RFC5426].

Il est RECOMMANDÉ que les déploiements de cette spécification utilisent le transport fondé sur TLS.

6. Format de message Syslog

Le message syslog a la définition d'ABNF [RFC5234] suivante :

```
SYSLOG-MSG = HEADER SP STRUCTURED-DATA [SP MSG]

HEADER  = PRI VERSION SP TIMESTAMP SP HOSTNAME SP APP-NAME SP PROCID SP MSGID
PRI  = "<" PRIVAL ">"
PRIVAL = 1*3DIGIT ; range 0 .. 191
VERSION = NONZERO-DIGIT 0*2DIGIT
HOSTNAME = NILVALUE / 1*255PRINTUSASCII

APP-NAME = NILVALUE / 1*48PRINTUSASCII
PROCID = NILVALUE / 1*128PRINTUSASCII
MSGID = NILVALUE / 1*32PRINTUSASCII

TIMESTAMP = NILVALUE / FULL-DATE "T" FULL-TIME
FULL-DATE = DATE-FULLYEAR "-" DATE-MONTH "-" DATE-MDAY
  DATE-FULLYEAR = 4DIGIT
  DATE-MONTH = 2DIGIT  ; 01-12
  DATE-MDAY = 2DIGIT  ; 01-28, 01-29, 01-30, 01-31 based on ; month/year
FULL-TIME = PARTIAL-TIME TIME-OFFSET
PARTIAL-TIME = TIME-HOUR ":" TIME-MINUTE ":" TIME-SECOND [TIME-SECFRAC]
  TIME-HOUR  = 2DIGIT  ; 00-23
  TIME-MINUTE = 2DIGIT  ; 00-59
  TIME-SECOND = 2DIGIT  ; 00-59
  TIME-SECFRAC = "." 1*6DIGIT
TIME-OFFSET = "Z" / TIME-NUMOFFSET
TIME-NUMOFFSET  = ("+" / "-") TIME-HOUR ":" TIME-MINUTE

STRUCTURED-DATA = NILVALUE / 1*SD-ELEMENT
SD-ELEMENT = "[" SD-ID *(SP SD-PARAM) "]"
SD-PARAM = PARAM-NAME "=" %d34 PARAM-VALUE %d34
SD-ID = SD-NAME
PARAM-NAME = SD-NAME
PARAM-VALUE = UTF-8-STRING              ; les caractères "", '\' et ; ']' DOIVENT être échappés.
SD-NAME = 1*32PRINTUSASCII              ; sauf '=', SP, ']', %d34 (")

MSG = MSG-ANY / MSG-UTF8
MSG-ANY = *OCTET                        ; ne commence pas par BOM
MSG-UTF8 = BOM UTF-8-STRING
BOM = %xEF.BB.BF

UTF-8-STRING = *OCTET                   ; chaîne UTF-8 comme spécifié dans la RFC 3629
```

OCTET = %d00-255
SP = %d32
PRINTUSASCII = %d33-126
NONZERO-DIGIT = %d49-57
DIGIT = %d48 / NONZERO-DIGIT
NILVALUE = "-"

6.1 Longueur de message

Les limites de taille de message Syslog sont dictées par la transposition de transport syslog utilisée. Il n'y a pas de limite supérieure en soi. Chaque transposition de transport définit la longueur minimum maximum de message exigée, et le minimum maximum DOIT être d'au moins 480 octets.

Tout receveur de transport DOIT être capable d'accepter des messages jusqu'à et inclus 480 octets. Toutes les mises en œuvre de receveur de transport DEVRAIENT être capables d'accepter des messages de jusqu'à et inclus 2048 octets. Les receveurs de transport PEUVENT recevoir des messages de plus de 2048 octets. Si un receveur de transport reçoit un message d'une longueur supérieure à ce qu'il prend en charge, le receveur de transport DEVRAIT tronquer la charge utile. Autrement, il PEUT éliminer le message.

Si un receveur de transport tronque des messages, la troncature DOIT se produire à la fin du message. Après la troncature, le message PEUT contenir un codage UTF-8 invalide ou des STRUCTURED-DATA invalides. Le receveur de transport PEUT éliminer le message ou PEUT essayer d'en traiter autant que possible dans ce cas.

6.2 HEADER

Le jeu de caractères utilisé dans HEADER *(en-tête)* DOIT être de sept bits ASCII dans un champ de huit bits comme décrit dans la [RFC5234]. Ce sont les codes ASCII comme définis dans [ANSI.X3-4] "Codes USA standard pour les échanges d'informations".

Le format d'en-tête est conçu pour fournie l'interopérabilité avec le vieux syslog fondé sur BSD. Pour les détails, voir l'Appendice A.1.

6.2.1 PRI

La partie PRI DOIT avoir trois, quatre, ou cinq caractères et va être enclose dans des crochets angulaires comme premier et dernier caractères. La partie PRI commence par un caractère "<" ("moins que", %d60) suivi par un nombre, qui est suivi par un caractère ">" ("supérieur à", %d62). Le nombre contenu dans ces crochets angulaires est appelé la valeur de priorité (PRIVAL) et représente la facilité et la sévérité. La valeur de priorité consiste en un, deux, ou trois entiers décimaux (ABNF DIGITS) utilisant les valeurs de %d48 (pour "0") à %d57 (pour "9").

Les valeurs de facilité et de sévérité ne sont pas normatives mais souvent utilisées. Elles sont décrites dans les tableaux suivants pour information. Les valeurs de facilité DOIVENT être dans la gamme de 0 à 23 inclus.

Code numérique	Facilité
0	messages du noyau
1	messages de niveau utilisateur
2	système de messagerie
3	automates système
4	messages de sécurité/autorisation
5	messages générés en interne par syslog
6	sous système d'imprimante en ligne
7	sous système de nouvelles du réseau
8	sous système UUCP
9	automate d'horloge
10	messages de sécurité/autorisation
11	automate FTP
12	sous système NTP
13	journal d'audit
14	journal d'alerte

15	automate d'horloge (note 2)
16	utilisation locale 0 (local0)
17	utilisation locale 1 (local1)
18	utilisation locale 2 (local2)
19	utilisation locale 3 (local3)
20	utilisation locale 4 (local4)
21	utilisation locale 5 (local5)
22	utilisation locale 6 (local6)
23	utilisation locale 7 (local7)

Tableau 1 : Facilités de message Syslog

Chaque priorité de message a aussi un indicateur de niveau de sévérité décimal. Ils sont décrits dans le tableau suivant avec leurs valeurs numériques. Les valeurs de sévérité DOIVENT être dans la gamme de 0 à 7 inclus.

Code numérique	Sévérité
0	Urgence : les système est inutilisable
1	Alerte : une action doit être prise immédiatement
2	Critique : conditions critiques
3	Erreur : conditions d'erreur
4	Avertissement : conditions d'avertissement
5	Remarque : condition normale mais significative
6	Information : messages d'information
7	Débogage : messages de niveau débogage

Tableau 2 : Sévérités de message Syslog

La valeur de priorité est calculée en multipliant d'abord le numéro de facilité par 8 et en ajoutant ensuite la valeur numérique de la sévérité. Par exemple, un message du noyau (Facilité = 0) avec une sévérité de Urgence (Sévérité = 0) aurait une valeur de priorité de 0. Aussi, un message "utilisation locale 4" (Facilité = 20) avec une sévérité de Remarque (Sévérité = 5) aurait une valeur de priorité de 165. Dans le PRI d'un message syslog, ces valeurs vont être placées entre les crochets angulaires respectivement comme <0> et <165>. La seule fois où une valeur de "0" suit le "<" est pour la valeur de priorité de "0". Autrement, des "0" en tête NE DOIVENT PAS être utilisés.

6.2.2 VERSION

Le champ VERSION note la version de la spécification de protocole syslog. Le numéro de version DOIT être incrémenté pour toute nouvelle spécification de protocole syslog qui change une partie du format de HEADER. Les changements incluent l'ajout ou la suppression de champs, ou un changement de la syntaxe ou de la sémantique des champs existants. Le présent document utilise une valeur de VERSION de "1". Les valeurs de VERSION sont allouées par l'IANA (paragraphe 9.1) via la méthode Action de normalisation décrite dans la [RFC5226].

6.2.3 TIMESTAMP

Le champ TIMESTAMP est un horodatage formalisé dérivé de la [RFC3339].

Alors que la [RFC3339] permet plusieurs syntaxes, le présent document impose des restrictions supplémentaires. La valeur de TIMESTAMP DOIT suivre ces restrictions :

- Les caractères "T" et "Z" dans cette syntaxe DOIVENT être en majuscules.
- L'usage du caractère "T" est EXIGÉ.
- Les secondes sautées NE DOIVENT PAS être utilisées.

Le générateur DEVRAIT inclure TIME-SECFRAC si sa précision d'horloge et ses performances le permettent. L'identifiant de SD "timeQuality" décrit au paragraphe 7.1 permet au générateur de spécifier la précision et la fiabilité de l'horodatage.

Une application syslog DOIT utiliser NILVALUE comme TIMESTAMP si l'application syslog est incapable d'obtenir l'heure du système.

6.2.3.1 Exemples

Exemple 1 : 1985-04-12T23:20:50.52Z
Cela représente 20 minutes et 50,52 secondes après la 23ème heure le 12 avril 1985 en UTC.

Exemple 2 : 1985-04-12T19:20:50.52-04:00
Cela représente la même heure que dans l'exemple 1, mais exprimée en heure standard US Est (observant l'heure d'hiver).

Exemple 3 : 2003-10-11T22:14:15.003Z
Cela représente le 11 octobre 2003 à 10:14:15 après midi, 3 millisecondes dans la seconde suivante. L'horodatage est en UTC. L'horodatage donne une résolution de milliseconde. Le créateur peut avoir en fait une meilleure résolution, mais fournir seulement trois chiffres pour la partie fraction de seconde ne nous le dit pas.

Exemple 4 : 2003-08-24T05:14:15.000003-07:00
Cela représente le 24 août 2003 à 05:14:15 du matin, 3 microsecondes dans la prochaine seconde. La résolution de microseconde est indiquée par les chiffres supplémentaires dans TIME-SECFRAC. L'horodatage indique que son heure locale est à -7 heures de l'UTC. Cet horodatage pourrait être créé dans la zone horaire US Pacifique durant l'heure d'hiver.

Exemple 5 : un TIMESTAMP invalide : 2003-08-24T05:14:15.000000003-07:00
Cet exemple est presque le même que l'exemple 4, mais il spécifie TIME-SECFRAC en nanosecondes. Il en résulte un TIME-SECFRAC plus long que les six chiffres permis, ce qui l'invalide.

6.2.4 HOSTNAME

Le champ HOSTNAME *(nom d'hôte)* identifie la machine qui a envoyé à l'origine le message syslog.

Le champ HOSTNAME DEVRAIT contenir le nom d'hôte et le nom de domaine du générateur dans le format spécifié dans le STD 13 [RFC1034]. Ce format est appelé un nom de domaine pleinement qualifié (FQDN, *Fully Qualified Domain Name*) dans ce document.

En pratique, toutes les applications syslog ne sont pas capables de fournir un FQDN. À ce titre, d'autres valeurs PEUVENT aussi être présentes dans HOSTNAME. Le présent document prend des dispositions pour utiliser d'autres valeurs dans de telles situations. Une application syslog DEVRAIT fournir d'abord la valeur disponible la plus spécifique. L'ordre de préférence pour le contenu du champ HOSTNAME est le suivant :

1. FQDN
2. Adresse IP statique
3. Nom d'hôte
4. Adresse IP dynamique
5. NILVALUE

Si une adresse IPv4 est utilisée, elle DOIT être dans le format de notation décimale séparée par des points comme utilisé dans le STD 13 [RFC1035]. Si une adresse IPv6 est utilisée, une représentation textuelle valide comme décrite dans la [RFC4291], paragraphe 2.2, DOIT être utilisée.

Les applications syslog DEVRAIENT utiliser de façon cohérente la même valeur dans le champ HOSTNAME aussi longtemps que possible.

La valeur NILVALUE DEVRAIT seulement être utilisée quand l'application syslog n'a pas de moyen d'obtenir son nom d'hôte réel. Cette situation est considérée comme hautement improbable.

6.2.5 APP-NAME

Le champ APP-NAME DEVRAIT identifier l'appareil ou l'application qui a généré le message. C'est une chaîne sans autre sémantique. Elle est destinée au filtrage des messages à un relais ou un collecteur.

La valeur NILVALUE PEUT être utilisée quand l'application syslog n'a pas d'idée de son APP-NAME ou ne peut pas fournir cette information. Il se peut qu'un appareil soit incapable de fournir cette information soit parce que c'est une décision de politique locale, soit parce que l'information n'est pas disponible, ou non applicable, sur l'appareil.

Ce champ PEUT être alloué par l'opérateur.

6.2.6 PROCID

PROCID *(identifiant de processus)* est une valeur qui est incluse dans le message, et n'a pas de signification d'interopérabilité, sauf qu'un changement de la valeur indique qu'il y a eu une discontinuité dans le système de rapports syslog. Le champ n'a pas de syntaxe ou sémantique spécifique ; la valeur dépend de la mise en œuvre et/ou est allouée par l'opérateur. La valeur NILVALUE PEUT être utilisée quand aucune valeur n'est fournie.

Le champ PROCID est souvent utilisé pour fournir le nom du processus ou l'identifiant de processus associé à un système syslog. La NILVALUE pourrait être utilisée quand un identifiant de processus n'est pas disponible. Sur un système incorporé dans l'identifiant de processus de système d'exploitation, PROCID pourrait être un identifiant de réamorçage.

PROCID peut permettre aux analyseurs de journaux de détecter des discontinuités dans les rapports syslog en détectant un changement de l'identifiant de processus syslog. Cependant, PROCID n'est pas une identification fiable de redémarrage de processus car le processus syslog redémarré pourrait avoir alloué le même identifiant de processus que le précédent processus syslog.

PROCID peut aussi être utilisé pour identifier quels messages appartiennent à un groupe de messages. Par exemple, un agent de transfert de messagerie SMTP pourrait mettre son identifiant de transaction SMTP dans le PROCID, ce qui permettrait au collecteur ou relais de grouper des messages sur la base de la transaction SMTP.

6.2.7 MSGID

Le MSGID *(identifiant de message)* DEVRAIT identifier le type of message. Par exemple, un pare-feu pourrait utiliser le MSGID "TCPIN" pour le trafic TCP entrant et le MSGID "TCPOUT" pour le trafic TCP sortant. Les messages avec le même MSGID devraient refléter des événements de même sémantique. Le MSGID lui-même est une chaîne sans autre sémantique. Il est destiné au filtrage des messages sur un relais ou un collecteur.

La NILVALUE DEVRAIT être utilisée quand l'application syslog ne fournit pas, ou ne peut pas fournir de valeur.

Ce champ PEUT être alloué par l'opérateur.

6.3 STRUCTURED-DATA

STRUCTURED-DATA *(données structurées)* fournit un mécanisme pour exprimer des informations dans un format de données bien défini, facilement analysable et interprétable. Il y a plusieurs scénarios d'usage. Par exemple, il peut exprimer des méta-informations sur le message syslog ou des informations spécifiques de l'application comme des compteurs de trafic ou des adresses IP.

STRUCTURED-DATA peut contenir zéro, un, ou plusieurs éléments de données structurées, qui sont appelées des "SD-ELEMENT" *(éléments de données structurées)* dans ce document.

Dans le cas d'éléments de données structurées zéro, le champ STRUCTURED-DATA DOIT contenir la NILVALUE.

Le jeu de caractères utilisé dans STRUCTURED-DATA DOIT être en ASCII à sept bits dans un champ de huit bits comme décrit dans la [RFC5234]. Ce sont les codes ASCII définis dans [ANSI.X3-4]. Une exception est le champ PARAM-VALUE (voir le paragraphe 6.3.3) dans lequel un codage UTF-8 DOIT être utilisé.

Un collecteur PEUT ignorer les éléments STRUCTURED-DATA mal formés. Un relais DOIT transmettre les STRUCTURED-DATA mal formés sans aucune altération.

6.3.1 SD-ELEMENT

Un SD-ELEMENT *(élément de données structurées)* consiste en un nom et une paire de paramètre nom-valeur. Le nom est appelé un SD-ID *(identifiant de données structurées)*. Les paires nom-valeur sont appelées des "SD-PARAM".

6.3.2 SD-ID

Les SD-ID sont sensibles à la casse et identifient de façon univoque le type et l'objet du SD-ELEMENT. Le même SD-ID NE DOIT PAS exister plus d'une fois dans un message.

Il y a deux formats pour les noms de SD-ID :

- Les noms qui ne contiennent pas de signe "@", (ABNF %d64) sont réservés pour être alloués par revue de l'IETF, comme décrit dans le BCP26 [RFC5226]. Actuellement, ce sont les noms définis à la Section 7. Les noms de ce format sont seulement valides si ils sont d'abord enregistrés par l'IANA. Les noms enregistrés NE DOIVENT PAS contenir de signe "@", (ABNF %d64) de signe égal ('=', ABNF %d61) d'accolade de clôture (']', ABNF %d93), de guillemets ('"', ABNF %d34), d'espace, ou de caractères de contrôle (code ASCII 127 et 32 ou moins).

- Tout le monde peut définir des SD-ID supplémentaires en utilisant des noms de format nom@<numéro d'entreprise privée>, par exemple, "notreSDID@32473". Le format de la partie précédant le signe "@" n'est pas spécifié ; cependant, ces noms DOIVENT être des chaînes US-ASCII imprimables, et NE DOIVENT PAS contenir de signe "'@" (ABNF %d64) de signe égal ('=', ABNF %d61), d'accolade de clôture (']', ABNF %d93), de guillemets ('"', ABNF %d34), d'espace, ou de caractères de contrôle. La partie qui suit le signe "@" DOIT être un numéro d'entreprise privée comme spécifié au paragraphe 7.2.2. Noter que dans ce document la valeur de 32473 est utilisé pour tous les numéros d'entreprise privée. Cette valeur a été réservée par l'IANA pour être utilisée comme exemple de numéro dans la documentation. Les mises en œuvre devront utiliser leur propre numéro d'entreprise privée pour le paramètre enterpriseId, et quand elles créent des noms de SD-ID localement extensibles.

6.3.3 SD-PARAM

Chaque SD-PARAM *(paramètre de données structurées)* consiste en un nom, appelé PARAM-NAME, et une valeur, appelée PARAM-VALUE.

PARAM-NAME est sensible à la casse. L'IANA contrôle tous les PARAM-NAME, à l'exception de ceux des SD-ID dont le nom contient un signe "@". La portée de PARAM-NAME est au sein d'un SD-ID spécifique. Donc, des valeurs de PARAM-NAME de nom égal contenues dans deux SD-ID différents ne sont pas les mêmes.

Pour prendre en charge les caractères internationaux, le champ PARAM-VALUE DOIT être codé en utilisant UTF-8. Une application syslog PEUT produire toute séquence UTF-8 valide. Une application syslog DOIT accepter toute séquence UTF-8 valide dans la "plus courte forme". Elle NE DOIT PAS échouer si des caractères de contrôle sont présents dans PARAM-VALUE. L'application syslog PEUT modifier les messages contenant des caractères de contrôle (par exemple, en changeant un octet de valeur 0 (USASCII NUL) en les quatre caractères "#000"). Pour les raisons mentionnées dans UNICODE TR36 [UNICODE], paragraphe 3.1, un générateur DOIT coder les messages dans la "plus courte forme" et un collecteur ou relais NE DOIT PAS interpréter les messages dans la "forme non la plus courte".

Dans PARAM-VALUE, les caractères '"' (ABNF %d34) '\' (ABNF %d92) et ']' (ABNF %d93) DOIVENT être échappés. Ceci est nécessaire pour éviter des erreurs d'analyse. L'échappement de ']' ne serait pas strictement nécessaire mais est EXIGÉ par cette spécification pour éviter des erreurs de mise en œuvre d'application syslog. Chacun de ces trois caractères DOIT être échappé comme respectivement '\"', '\\', et '\]'. La barre oblique inverse est utilisée pour l'échappement de caractère de contrôle pour la cohérence avec son utilisation pour l'échappement dans les autres parties du message syslog ainsi que dans le syslog traditionnel.

Une barre oblique inverse ('\') qui n'est suivie d'aucun des trois caractères décrits est considérée comme une séquence d'échappement invalide. Dans ce cas, la barre oblique inverse DOIT être traitée comme une barre oblique inverse régulière et le caractère suivant comme un caractère régulier. Donc, la séquence invalide NE DOIT PAS être altérée.

Un SD-PARAM PEUT être répété plusieurs fois dans un SD-ELEMENT.

6.3.4 Contrôle des changements

Une fois les SD-ID et PARAM-NAME définis, la syntaxe et la sémantique de ces objets NE DOIT PAS être altérée. Si un changement à un objet existant est désiré, un nouveau SD-ID ou PARAM-NAME DOIT être créé et l'ancien rester inchangé. Des PARAM-NAME FACULTATIFS PEUVENT être ajoutés à un SD-ID existant.

6.3.5 Exemples

Tous les exemples de ce paragraphe montrent seulement la partie données structurées du message. Les exemples devraient être considérés comme étant sur une ligne. Ils sont montrés sur plusieurs lignes dans ce document pour la lisibilité. Une description est donnée après chaque exemple.

Exemple 1 - Valide

```
[exempleSDID@32473 iut="3" eventSource="Application"eventID="1011"]
```

Cet exemple est un élément de données structurées avec un SD-ID non contrôlé par l'IANA de type "exampleSDID@32473", qui a trois paramètres.

Exemple 2 - Valide

```
[exempleSDID@32473 iut="3" eventSource="Application"
eventID="1011"][examplePriority@32473 class="high"]
```

C'est le même exemple qu'en 1, mais avec un second élément de données structurées. Noter que l'élément de données structurées suit immédiatement le premier (il n'y a pas de SP entre eux).

Exemple 3 - Invalide

```
[exempleSDID@32473 iut="3" eventSource="Application"
eventID="1011"] [examplePriority@32473 class="high"]
```

C'est presque le même exemple qu'en 2, mais il a une petite erreur -- il y a un caractère SP entre les deux éléments de données structurées ("]SP["). Ceci est invalide. Il va causer la fin du champ STRUCTURED-DATA après le premier élément. Le second élément va être interprété comme faisant partie du champ MSG.

Exemple 4 - Invalide

```
[ exempleSDID@32473 iut="3" eventSource="Application"
eventID="1011"][examplePriority@32473 class="high"]
```

Cet exemple est presque le même que en 2. Il a une autre petite erreur -- le caractère SP se produit après la crochet initial. Un élément de données structurées SD-ID DOIT suivre immédiatement le crochet de début, donc le caractère SP invalide les STRUCTURED-DATA. Une application syslog PEUT éliminer ce message.

Exemple 5 - Valide

```
[sigSig ver="1" rsID="1234" ... signature="..."]
```

L'exemple 5 est valide. Il montre un SD-ID hypothétique alloué par l'IANA. Les ellipses (...) notent du contenu manquant, qui a été laissé en dehors de cet exemple pour abréger.

6.4 MSG

La partie MSG contient un message de forme libre qui donne des informations sur l'événement.

Le jeu de caractères utilisé dans MSG DEVRAIT être UNICODE, codé en utilisant UTF-8 comme spécifié dans la [RFC3629]. Si l'application syslog ne peut pas coder le MSG en Unicode, elle PEUT utiliser tout autre codage.

L'application syslog DEVRAIT éviter des valeurs d'octet en dessous de 32 (la gamme traditionnelle de caractères de contrôle US-ASCII sauf DEL). Ces valeurs sont légales, mais une application syslog PEUT modifier ces caractères à réception. Par exemple, elle pourrait les changer en une séquence d'échappement (par exemple, la valeur 0 peut être changée en "\0"). Une application syslog NE DEVRAIT PAS modifier d'autres valeurs d'octet.

Si une application syslog code MSG en UTF-8, la chaîne DOIT commencer par la marque d'ordre d'octet (BOM, *byte order mask*) Unicode, qui pour UTF-8 est l'ABNF %xEF.BB.BF. L'application syslog DOIT coder dans la "plus courte forme" et PEUT utiliser toute séquence UTF-8 valide.

Si une application syslog traite un MSG commençant par une BOM et si le MSG contient de l'UTF-8 qui n'est pas la plus courte forme, le MSG NE DOIT PAS être interprété comme étant codé en UTF-8, pour les raisons mentionnées dans [UNICODE], paragraphe 3.1. Des directives sur ce point sont données à l'Appendice A.8.

Aussi, conformément à UNICODE TR36 [UNICODE], une application syslog NE DOIT PAS interpréter les messages dans la "forme non la plus courte". Elle NE DOIT PAS interpréter les séquences UTF-8 invalides.

6.5 Exemples

Voici des exemples de messages syslog valides. Une description de chaque exemple se trouve en dessous. Les exemples se fondent sur les exemples similaires de la [RFC3164] et peuvent être familiers aux lecteurs. La BOM Unicode par ailleurs non imprimable est représentée par "BOM" dans les exemples.

Exemple 1 - sans STRUCTURED-DATA

```
<34>1 2003-10-11T22:14:15.003Z mamachine.exemple.com su - ID47
- BOM'su root' failed for lonvick on /dev/pts/8
```

Dans cet exemple, la VERSION est 1 et la facilité a la valeur 4. La sévérité est 2. Le message a été créé le 11 octobre 2003 à 10:14:15 après midi en UTC, 3 millisecondes dans la prochaine seconde. Le message a pour origine un hôte qui s'identifie comme "mamachine.exemple.com". Le APP-NAME est "su" et le PROCID est inconnu. Le MSGID est "ID47". Le MSG est "'su root' failed for lonvick...", codé en UTF-8. Le codage est défini par la BOM. Aucune STRUCTURED-DATA n'est présente dans le message ; c'est indiqué par "-" dans le champ STRUCTURED-DATA.

Exemple 2 - sans STRUCTURED-DATA

```
<165>1 2003-08-24T05:14:15.000003-07:00 192.0.2.1
myproc 8710 - - %% Il est temps de faire les do-nuts.
```

Dans cet exemple, la VERSION est encore 1. La facilité est 20, la sévérité 5. Le message a été créé le 24 août 2003 à 5:14:15 du matin, avec un décalage de - 7 heures sur l'UTC, 3 microsecondes dans la prochaine seconde. Le HOSTNAME est "192.0.2.1", donc l'application syslog ne connaissait pas son FQDN et a utilisé à la place une de ses adresses IPv4. Le APP-NAME est "myproc" et le PROCID est "8710" (par exemple, ce pourrait être le PID UNIX). Il n'y a pas de STRUCTURED-DATA présentes dans le message ; c'est indiqué par "-" dans le champ STRUCTURED-DATA. Il n'y a pas de MSGID spécifique et c'est indiqué par le "-" dans le champ MSGID.
Le message est "%% Il est temps de faire les do-nuts.". Comme la BOM Unicode manque, l'application syslog ne connaît pas le codage de la partie MSG.

Exemple 3 - avec STRUCTURED-DATA

```
<165>1 2003-10-11T22:14:15.003Z mamachine.exemple.com
evntslog - ID47 [exempleSDID@32473 iut="3" eventSource=
"Application" eventID="1011"] BOMAn application event log entry...
```

Cet exemple est modélisé d'après l'exemple 1. Cependant, cette fois, il conteint des STRUCTURED-DATA, un seul élément avec la valeur "[exempleSDID@32473 iut="3" eventSource="Application" eventID="1011"]". Le MSG lui-même est "An application event log entry..." La BOM au début du MSG indique le codage UTF-8.

Exemple 4 - seulement de STRUCTURED-DATA

```
<165>1 2003-10-11T22:14:15.003Z mamachine.exemple.com
evntslog - ID47 [exempleSDID@32473 iut="3" eventSource=
"Application" eventID="1011"][examplePriority@32473class="high"]
```

Cet exemple montre un message avec seulement des STRUCTURED-DATA et pas de partie MSG. C'est un message valide.

7. Identifiants de données structurées

Cette Section définit les SD-ID initiaux enregistrés par l'IANA. Voir au paragraphe 6.3 la définition de éléments de données structurées. Tous les SD-ID définis ici sont FACULTATIFS.

Dans certains des SD-ID suivants, une longueur maximum est quantifiée pour les valeurs de paramètre. Dans chacun de ces cas, l'application syslog DOIT être prête à recevoir le nombre de caractères défini dans tout codet UTF-8 valide. Comme chaque caractère peut faire jusqu'à 6 octets, il est RECOMMANDÉ que chaque application syslog soit prête à recevoir jusqu'à 6 octets par caractère.

7.1 timeQuality

Le SD-ID "timeQuality" *(qualité horaire)* PEUT être utilisé par le générateur pour décrire sa notion de l'heure du système. Ce SD-ID DEVRAIT être écrit si le générateur n'est pas proprement synchronisé avec une source d'heure externe fiable ou si il ne sait pas si ses informations de zone horaire sont correctes. Le principal usage de cet élément de données structurées est de fournir des informations sur le niveau de confiance qu'il a dans le TIMESTAMP décrit au paragraphe 6.2.3. Tous les paramètres sont FACULTATIFS.

7.1.1 tzKnown

Le paramètre "tzKnown" *(zone horaire connue)* indique si le générateur connaît sa zone horaire. Si oui, la valeur "1" DOIT être utilisée. Si les informations de zone horaire sont douteuses, la valeur "0" DOIT être utilisée. Si le générateur connaît sa zone horaire mais décide d'émettre l'heure en UTC, la valeur "1" DOIT être utilisée (parce que la zone horaire est connue).

7.1.2 isSynced

Le paramètre "isSynced" *(est synchrone)* indique si le générateur est synchronisé à une source horaire externe fiable, par exemple, via NTP. Si le générateur est synchronisé, la valeur "1" DOIT être utilisée. Sinon, la valeur "0" DOIT être utilisée.

7.1.3 syncAccuracy

Le paramètre "syncAccuracy" *(précision de synchronisation)* indique comment le générateur pense que sa synchronisation est précise. C'est un entier qui décrit le nombre maximum de microsecondes pendant lequel son horloge peut être non active entre les intervalles de synchronisation.

Si la valeur "0" est utilisée pour "isSynced", ce paramètre NE DOIT PAS être spécifié. Si la valeur "1" est utilisée pour "isSynced" mais si le paramètre "syncAccuracy" est absent, un collecteur ou relais peut supposer que les informations horaires fournies sont assez précises pour être considérées correctes. Le paramètre "syncAccuracy" DOIT être écrit seulement si le générateur a réellement connaissance de la fiabilité de la source horaire externe. Dans la plupart des cas, il va obtenir une connaissance réelle par la configuration de l'opérateur.

7.1.4 Exemples

Voici un exemple d'un générateur qui ne sait pas sa zone horaire ou si il est synchronisé :

```
[timeQuality tzKnown="0" isSynced="0"]
```

Avec cette information, le générateur indique que ses informations horaires ne sont pas fiables. Cela peut être l'indication pour le collecteur ou le relais d'utiliser son heure locale au lieu du TIMESTAMP fourni par le message pour la corrélation de plusieurs messages provenant de générateurs différents.

Voici un exemple d'un générateur qui connaît sa zone horaire et sait qu'il est proprement synchronisé à une source externe fiable :

```
[timeQuality tzKnown="1" isSynced="1"]
```

Voici un exemple d'un générateur qu sait sa zone horaire et qu'il est synchronisé à une source externe. Il sait aussi la

précision de sa synchronisation externe :

[timeQuality tzKnown="1" isSynced="1" syncAccuracy="60000000"]

La différence avec l'exemple précédent est que le générateur s'attend à ce que son horloge reste dans les 60 secondes de l'heure officielle. Donc, si le générateur rapporte qu'il est 9:00:00, il n'est pas plus tôt que 8:59:00 et pas plus tard que 9:01:00.

7.2 origin

Le SD-ID "origin" PEUT être utilisé pour indiquer l'origine d'un message syslog. Les paramètres suivants peuvent être utilisés. Tous les paramètres sont FACULTATIFS.

La spécification de ces paramètres est principalement une aide pour les analyseurs de journaux et applications similaires.

7.2.1 ip

Le paramètre "ip" note une adresse IP dont le générateur sait qu'il l'avait au moment de la génération du message. Il DOIT contenir la représentation textuelle d'une adresse IP comme mentionné au paragraphe 6.2.4.

Ce paramètre peut être utilisé pour fournir des informations d'identification en plus de ce qui est présent dans le champ HOSTNAME. Cela pourrait être particulièrement utile si l'adresse IP de l'hôte est incluse dans le message alors que le champ HOSTNAME contient le FQDN. C'est aussi utile pour décrire toutes les adresses IP d'un hôte multi rattachements.

Si un générateur a plusieurs adresses IP, il PEUT mentionner une de ses adresses IP dans le paramètre "ip" ou il PEUT inclure plusieurs paramètres "ip" dans un seul élément "origin" de données structurées.

7.2.2 enterpriseId

Le paramètre "enterpriseId" DOIT être un "code d'entreprise privée de gestion de réseau SMI", conservé par l'IANA, dont le préfixe est "iso.org.dod.internet.private.enterprise" (1.3.6.1.4.1). Le numéro qui suit DOIT être unique et DOIT être enregistré par l'IANA selon la [RFC2578]. Une entreprise n'est autorisée à allouer des valeurs qu'au sein de la sous arborescence "iso.org.dod.internet.private.enterprise.<private enterprise number>" allouée par l'IANA à cette entreprise. Le "enterpriseId" DOIT contenir seulement une valeur de la sous arborescence "iso.org.dod.internet.private.enterprise.<private enterprise number>". En général, seulement le numéro d'entreprise privée alloué par l'IANA est nécessaire (un seul numéro). Une entreprise pourrait décider d'utiliser des sous-identifiants en dessous de son numéro d'entreprise privée. Si des sous-identifiants sont utilisés, ils DOIVENT être séparés par des points et être représentés par des nombres décimaux. Un exemple en serait "32473.1.2". Noter que l'identifiant "32473.1.2" est juste un exemple et NE DOIT PAS être utilisé. La liste complète à jour des numéros d'entreprise privée (PEN, *Private Enterprise Number*) est tenue par l'IANA.

En spécifiant un numéro d'entreprise privée, le fabricant permet un traitement plus spécifique du message.

7.2.3 software

Le paramètre "software" identifie de façon univoque le logiciel qui a généré le message. Si il est utilisé, "enterpriseId" DEVRAIT aussi être spécifié, afin qu'un logiciel spécifique du fabricant puisse être identifié. Le paramètre "software" n'est pas le même que le champ d'en-tête APP-NAME. Il DOIT toujours contenir le nom du logiciel générateur, tandis que APP-NAME peut contenir n'importe quoi d'autre, y compris une valeur configurée par l'opérateur.

Le paramètre "software" est une chaîne. Il NE DOIT PAS faire plus de 48 caractères.

7.2.4 swVersion

Le paramètre "swVersion" identifie de façon univoque la version du logiciel qui a généré le message. Si il est utilisé, les paramètres "software" et "enterpriseId" DEVRAIT aussi être fournis.

Le paramètre "swVersion" est une chaîne. Il NE DOIT PAS faire plus de 32 caractères.

7.2.5 Exemple

Voici un exemple avec plusieurs adresses IP :

```
[origin ip="192.0.2.1" ip="192.0.2.129"]
```

Dans cet exemple, le générateur indique qu'il a deux adresses IP, une de 192.0.2.1 et l'autre de 192.0.2.129.

7.3 meta

Le SD-ID "meta" PEUT être utilisé pour fournir des méta-informations sur le message. Les paramètres suivants peuvent être utilisés. Tous les paramètres sont FACULTATIFS. Si le SD-ID "meta" est utilisé, au moins un paramètre DEVRAIT être spécifié.

7.3.1 sequenceId

Le paramètre "sequenceId" *(identifiant de séquence)* trace la séquence dans laquelle le générateur soumet les messages au transport syslog pour envoi. C'est un entier qui DOIT être réglé à 1 quand la fonction syslog est lancée et DOIT être augmenté à chaque message jusqu'à une valeur maximum de 2 147 483 647. Si cette valeur est atteinte, le message suivant DOIT être envoyé avec un sequenceId de 1.

7.3.2 sysUpTime

Le paramètre "sysUpTime" PEUT être utilisé pour inclure le paramètre SNMP "sysUpTime" dans le message. Sa syntaxe et sa sémantique sont définies dans la [RFC3418].

Comme syslog ne prend pas en charge directement la syntaxe SNMP "INTEGER", la valeur DOIT être représentée comme un entier décimal (pas de virgule décimale) utilisant seulement les caractères "0", "1", "2", "3", "4", "5", "6", "7", "8", et "9".

Noter que la sémantique de la RFC 3418 est "l'heure (en centièmes de seconde) depuis que la portion gestion de réseau du système a été réinitialisée". Cela se rapporte bien sûr à la portion de gestion relative à SNMP du système, qui PEUT être différente de la portion de gestion relative à syslog du système.

7.3.3 language

Le paramètre "language" PEUT être spécifié par le générateur pour porter des informations sur le langage naturel utilisé dans le message. Si il est spécifié, il DOIT contenir un identifiant de langage comme défini dans le BCP 47 [RFC4646].

8. Considérations sur la sécurité

8.1 UNICODE

Le présent document utilise le codage UTF-8 pour les champs PARAM-VALUE et MSG. Il y a un certain nombre de problèmes de sécurité avec UNICODE. Toute mise en œuvre et tout opérateur devrait revoir le UNICODE TR36 [UNICODE] (UTR36) pour en savoir plus sur ces problèmes. Le présent document met en garde contre les problèmes techniques mentionnés dans UTR36 en EXIGEANT le codage "de plus courte forme" pour les applications syslog. Cependant, l'usurpation visuelle due à la confusion de caractères persiste. Le présent document essaye de minimiser les effets de l'usurpation visuelle en ne permettant UNICODE que lorsque le script local est attendu et nécessaire. Dans tous les autres champs, l'US-ASCII est EXIGÉ. Aussi, les champs PARAM-VALUE et MSG ne devraient pas être la principale source des informations d'identification, réduisant encore les risques associés à l'usurpation visuelle.

8.2 Caractères de contrôle

Le présent document n'impose aucune restriction obligatoire au contenu de MSG ou PARAM-VALUE. À ce titre, ils PEUVENT contenir des caractères de contrôle, y compris le caractère NUL.

Dans certains langages de programmation (notamment C et C++) le caractère NUL (ABNF %d00) a traditionnellement une signification particulière comme terminaison de chaîne. La plupart des mises en œuvre de ces langages supposent qu'une chaîne ne va pas s'étendre au delà du premier caractère NUL. C'est principalement une restriction aux librairies de support de démarrage. Cette restriction est souvent importée dans les programmes et langages de script écrits dans ces langages. À ce titre, les caractères NUL doivent être examinés avec grand soin et être traités de façon appropriée. Un attaquant peut délibérément inclure des caractères NUL pour cacher des informations après eux. Un traitement incorrect du caractère NUL peut aussi invalider des sommes de contrôle cryptographique qui sont transmises dans le message.

De nombreux éditeurs de texte populaires sont aussi écrits dans des langages qui ont cette restriction. Coder les caractères NUL quand on écrit des fichiers de texte est conseillé. Si ils sont mémorisés sans codage, le fichier peut devenir illisible.

D'autres caractères de contrôle peuvent aussi être problématiques. Par exemple, un attaquant peut délibérément inclure des caractères d'espace arrière pour rendre des parties de message d'enregistrement illisibles. Des problèmes similaires existent pour presque tous les caractères de contrôle.

Finalement, des séquences UTF-8 invalides peuvent être utilisées par un attaquant pour injecter des caractères de contrôle ASCII.

La présente spécification permet à une application syslog de reformater les caractères de contrôle reçus. Entre autres, les risques de sécurité associés aux caractères de contrôle ont été une force importante qui a conduit à cette restriction. Les générateurs sont informés que si un codage autre que ASCII et UTF8 est utilisé, le receveur peut corrompre le message en tentant de filtrer les caractères de contrôle ASCII.

8.3 Troncature de message

La troncature de message peut être détournée par un attaquant pour cacher des informations d'enregistrement vitales. Les messages au delà de la taille minimum prise en charge peuvent être éliminés ou tronqués par le receveur de transport. À ce titre des informations d'enregistrement vitales peuvent être perdues.

Afin d'empêcher la perte d'informations, les messages ne devraient pas être plus longs que la taille minimum maximum requise au paragraphe 6.1. Pour les meilleures performances et la fiabilité, les messages devraient être aussi petits que possible. Les informations importantes devraient être placées aussi tôt que possible dans le message parce que des informations au début du message ont moins de chances d'être éliminées par un receveur de transport limité en taille.

Un générateur devrait limiter la taille de toutes données fournies par l'utilisateur au sein d'un message syslog. Si il ne le fait pas, un attaquant peut fournir de grandes données dans l'espoir d'exploiter une faiblesse potentielle.

8.4 Répétition

Il n'y a pas de mécanisme dans le protocole syslog pour détecter la répétition de message. Un attaquant peut enregistrer un ensemble de messages qui indiquent une activité normale d'une machine. Plus tard, cet attaquant peut retirer cette machine du réseau et répéter les messages syslog au relais ou collecteur. Même avec le champ TIMESTAMP dans la partie HEADER, un attaquant peut enregistrer les paquets et simplement les modifier pour refléter l'heure courante avant de les retransmettre. Les administrateurs peuvent ne rien trouver d'anormal dans les messages reçus, et leur réception indiquerait faussement une activité normale de la machine.

La signature cryptographique des messages pourrait empêcher l'altération des TIMESTAMP et donc de l'attaque en répétition.

8.5 Livraison fiable

Parce qu'il n'y a pas de mécanisme décrit dans le présent document pour assurer la livraison, et parce que le transport sous-jacent peut être non fiable (par exemple, UDP) certains messages peuvent être perdus. Ils peuvent soit être éliminés par l'encombrement du réseau, soit ils peuvent être interceptés et éliminés dans une intention malveillante. Les conséquences de l'élimination d'un ou plusieurs messages syslog ne peuvent pas être déterminées. Si les messages sont de simples mises à jour d'état, alors leur non réception peut n'être pas remarquée ou peut causer des ennuis pour les opérateurs du système. Par ailleurs, si les messages sont plus critiques, alors les administrateurs ne peuvent pas être avertis du développement et de problèmes potentiellement sérieux. Les messages peuvent aussi être interceptés et éliminés par un attaquant comme moyen de cacher des activités non autorisées.

Il peut aussi être désirable d'inclure des caractéristiques de limitation de débit dans les générateurs et relais syslog. Cela peut réduire les problèmes potentiels d'encombrement quand des salves de messages se produisent.

La livraison fiable peut n'être pas toujours désirable. Livraison fiable signifie que le générateur ou relais syslog doit se bloquer quand le relais ou collecteur n'est pas capable d'accepter plus de messages. Dans certains systèmes d'exploitation, à savoir Unix/Linux, le générateur ou relais syslog fonctionne à l'intérieur de processus système à haute priorité (syslogd). Si ce processus se bloque, le système entier s'arrête. La même chose se produit si il y a une situation de blocage entre syslogd et par exemple, le serveur DNS.

Pour prévenir ces problèmes, la livraison fiable peut être mise en œuvre d'une façon qui élimine intentionnellement les messages quand l'application syslog se bloquerait autrement. L'avantage de la livraison fiable dans ce cas est que le générateur ou relais syslog élimine le message en connaissance de cause et est capable de le notifier au relais ou collecteur. Ainsi, le relais ou collecteur reçoit l'information que quelque chose est perdu. Avec la livraison non fiable, le message serait simplement perdu sans aucune indication que la perte s'est produite.

8.6 Contrôle d'encombrement

Parce que syslog peut générer des quantités illimitées de données, le transfert de ces données sur UDP est généralement problématique, parce que UDP n'a pas de mécanisme de contrôle de l'encombrement. Les mécanismes de contrôle de l'encombrement qui répondent à l'encombrement en réduisant le trafic et établissent un certain degré d'équité entre les flux qui partagent le même chemin sont vitaux pour le fonctionnement stable de l'Internet [RFC2914]. C'est pourquoi la mise en œuvre du transport de syslog sur TLS est EXIGÉE et RECOMMANDÉE pour une utilisation générale.

Les seuls environnements où le transport de syslog sur UDP PEUT être utilisé comme solution de remplacement au transport sur TLS sont les réseaux gérés, où le chemin de réseau a été explicitement provisionné pour le trafic UDP syslog par des mécanismes d'ingénierie du trafic, comme la limitation de débit ou la réservation de capacités. Dans tous les autres environnements, le transport TLS DEVRAIT être utilisé.

Dans toute mise en œuvre, il peut se produire une situation dans laquelle un générateur ou relais va avoir besoin de bloquer l'envoi de messages. Un cas courant est quand une file d'attente interne est pleine. Cela peut arriver à cause d'une limitation de débit ou de performances ralenties de l'application syslog. Dans tous les cas, il est fortement RECOMMANDÉ qu'aucun message ne soit éliminé, mais ils devraient être temporairement mémorisés jusqu'à ce qu'ils puissent être transmis. Cependant, si ils doivent être éliminés, il est RECOMMANDÉ que le générateur ou relais élimine les messages de moindre sévérité en faveur des messages de sévérité supérieure.

Les messages avec une valeur numérique de SEVERITY inférieure ont une sévérité pratique plus élevée que ceux avec une valeur numérique plus élevée. Dans cette situation, les messages qui sont à éliminer DEVRAIT simplement l'être. L'application syslog peut notifier à un collecteur ou relais le fait qu'il a éliminé des messages.

8.7 Intégrité de message

En plus d'être éliminés, les messages syslog peuvent être endommagés dans le transit, ou un attaquant peut les modifier par malveillance. Dans ces cas, le contenu original du message ne va pas être livré au collecteur ou relais. De plus, si un attaquant est positionné entre l'envoyeur de transport et le receveur de transport des messages syslog, il peut être capable d'intercepter et modifier ces messages pendant qu'ils sont en transit pour cacher des activités non autorisées.

8.8 Observation de message

Bien qu'il n'y ait pas de lignes directrices strictes concernant le format du message, la plupart des messages syslog sont générés sous une forme lisible pas l'homme avec l'hypothèse que les administrateurs capables devraient pouvoir les lire et comprendre leur signification. Le protocole syslog n'a pas de mécanisme pour assurer la confidentialité des messages en transit. Dans la plupart des cas, passer les messages en clair est un avantage pour le personnel opérateur si il observe les paquets dans le réseau. Le personnel opérateur peut être capable de lire les messages et de les associer à d'autres événements provenant d'autres paquets qui traversent le réseau pour suivre et corriger les problèmes. Malheureusement, un attaquant peut aussi être capable d'observer le contenu lisible par l'homme des messages syslog. L'attaquant peut alors utiliser les connaissance obtenues de ces messages pour compromettre une machine ou faire d'autres dommages.

Il est conseillé aux opérateurs d'utiliser une transposition de transport sûre pour éviter ce problème.

8.9 Configuration inappropriée

Parce qu'il n'y a pas d'informations de contrôle distribuées sur les messages ou configurations, il est entièrement de la responsabilité de l'administrateur de réseau de s'assurer que les messages vont réellement aux receveurs prévus. Des cas ont été notés où des applications syslog étaient configurées par inadvertance à envoyer des messages syslog aux mauvais relais ou collecteurs. Dans de nombreux cas, les relais ou collecteurs non avertis ne peuvent pas être configurés à recevoir des messages syslog et vont probablement les éliminer. Dans certains autres cas, la réception de messages syslog a causé des problèmes au receveur imprévu. Si des messages ne vont pas au receveur prévu, ils ne peuvent alors pas être revus ou traités.

Utiliser une transposition de transport fiable peut aider à identifier certains de ces problèmes. Par exemple, elle peut identifier un problème où un message est envoyé à un système qui n'est pas configuré à recevoir des messages. Elle ne peut pas identifier l'envoi de messages à une mauvaise machine qui accepte les messages.

8.10 Boucle de transmission

Comme le montre le diagramme 2, les machines peuvent être configurées à relayer les messages syslog aux relais suivants avant d'atteindre un collecteur. Dans un cas particulier, un administrateur trouve qu'il a par erreur configuré deux relais à transmettre des messages avec certaines valeurs de SEVERITY l'un à l'autre. Quand l'une de ces machines reçoit ou génère ce type de message, elle va le transmettre à l'autre relais. Ce relais va à son tour le retransmettre. Ce cycle cause une dégradation au réseau intermédiaire ainsi qu'à la disponibilité de traitement sur les deux appareils. Les administrateurs de réseau doivent veiller à ne pas causer une telle spirale mortelle.

8.11 Considérations de charge

Les administrateurs de réseau doivent prendre le temps d'estimer la capacité appropriée du collecteur syslog. Un attaquant peut effectuer une attaque de déni de service en remplissant le disque du collecteur avec de faux messages. Placer les enregistrements dans un fichier circulaire peut atténuer ce risque mais a pour conséquence de ne pas assurer qu'un administrateur va être capable de revoir les enregistrements à l'avenir. Dans ce domaine, un receveur de transport doit avoir une interface réseau capable de recevoir les messages qui lui sont envoyés.

Les administrateurs et les planificateurs de réseau doivent aussi revoir de façon critique les chemins du réseau entre les générateurs, les relais, et les collecteurs. Les messages syslog générés ne devraient submerger les liaisons du réseau.
Afin de réduire l'impact de ce problème, l'utilisation de transports qui garantissent la livraison est recommandée.

8.12 Déni de service

Comme avec tout système, un attaquant peut juste submerger un receveur de transport en lui envoyant plus de messages qu'il ne peut être traité par l'infrastructure ou l'appareil lui-même. Les auteurs de mise en œuvre devraient tenter de fournir des caractéristiques qui minimisent cette menace, de façon à accepter seulement des messages syslog provenant d'adresses IP connues.

9. Considérations relatives à l'IANA

9.1 Version

L'IANA a créé un registre intitulé "Valeurs de version syslog" des valeurs de VERSION comme décrit au paragraphe 6.2.2. Les numéros de version DOIVENT être incrémentés pour toute nouvelle spécification de protocole syslog qui change une partie de HEADER. Les changements incluent l'ajout ou la suppression de champs ou un changement de syntaxe ou de sémantique des champs existants.
Les numéros de VERSION doivent être enregistrés via la méthode Action de normalisation décrite dans la [RFC5226]. L'IANA a enregistré les VERSION montrées dans le Tableau 3 ci-dessous.

VERSION	FORMAT
1	Défini dans la [RFC5424]

Tableau 3 : Versions enregistrées par l'IANA

9.2 Identifiants de données structurées

L'IANA a créé un registre intitulé "Valeurs d'identifiant de données structurées syslog" des identifiants de données structurées (SD-ID) avec leurs valeurs associées de PARAM-NAME comme décrit à la Section 7.

De nouvelles valeurs de SD-ID et de PARAM-NAME doivent être enregistrées par la méthode de revue de l'IETF décrite dans la [RFC5226].

Une fois que des SD-ID et des SD-PARAM sont définis, la syntaxe et la sémantique de ces objets NE DOIT PAS être altérée. Si un changement d'un objet existant est désiré, un nouveau SD-ID ou SD-PARAM DOIT être créé et l'ancien rester inchangé.

Une disposition est faite ici pour les noms localement extensibles. L'IANA ne va pas enregistrer, ni contrôler les noms qui portent le signe @ (ABNF %d64) en eux.

L'IANA a enregistré les SD-ID et PARAM-NAME montrés dans le Tableau 4 ci-dessous .

SD-ID	PARAM-NAME	
timeQuality		FACULTATIF
	tzKnown	FACULTATIF
	isSynced	FACULTATIF
	syncAccuracy	FACULTATIF
origin		FACULTATIF
	ip	FACULTATIF
	enterpriseId	FACULTATIF
	software	FACULTATIF
	swVersion	FACULTATIF
meta		FACULTATIF
	sequenceId	FACULTATIF
	sysUpTime	FACULTATIF
	language	FACULTATIF

Tableau 4 : SD-ID et leurs PARAM-NAME enregistrés par l'IANA

10. Groupe de travail

Le groupe de travail peut être contacté via la liste de diffusion : syslog@ietf.org

Les présidents actuels du groupe de travail peuvent être contactés à :
Chris Lonvick, Cisco Systems, mél : clonvick@cisco.com
David Harrington, Huawei Technologies USA, mél : dbharrington@comcast.net

11. Remerciements

Les auteurs souhaitent remercier Chris Lonvick, Jon Callas, Andrew Ross, Albert Mietus, Anton Okmianski, Tina Bird, Devin Kowatch, David Harrington, Sharon Chisholm, Richard Graveman, Tom Petch, Dado Colussi, Clément Mathieu, Didier Dalmasso, et toutes les autres personnes qui ont commenté les diverses versions de cettes proposition.

12. Références

12.1 Références normatives

[ANSI.X3-4] American National Standards Institute, "USA Code for Information Interchange", ANSI X3.4, 1968.

[RFC1034] P. Mockapetris, "Noms de domaines - Concepts et facilités", STD 13, novembre 1987. (*MàJ par* RFC1101, 1183, 1348, 1876, 1982, 2065, 2181, 2308, 2535, 4033, 4034, 4035, 4343, 4035, 4592, 5936, 8020, 8482, 8767)

[RFC1035] P. Mockapetris, "Noms de domaines – Mise en œuvre et spécification", STD 13, novembre 1987. (*MàJ par* RFC1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2673, 2845, 3425, 3658, 4033, 4034, 4035, 4343, 5936, 5966, 6604, 7766, 8482, 8767)

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

[RFC2578] K. McCloghrie, D. Perkins, J. Schoenwaelder, "Structure des informations de gestion, version 2 (SMIv2)", avril 1999. (STD0058)

[RFC2914] S. Floyd, "Principes du contrôle d'encombrement", BCP 41, septembre 2000.

[RFC3339] G. Klyne, C. Newman, "La date et l'heure sur l'Internet : horodatages", juillet 2002. *(P.S.)*

[[RFC3418] R. Presuhn, éd., "Base de données d'informations de gestion (MIB) pour le protocole simple de gestion de réseau (SNMP)", décembre 2002. (STD0062)

[RFC3629] F. Yergeau, "UTF-8, un format de transformation de la norme ISO 10646", STD 63, novembre 2003, DOI 10.17487/RFC3629.

[RFC4291] R. Hinden, S. Deering, "Architecture d'adressage IP version 6", février 2006. (*MàJ par* 5952 et 6052, 8064) *(D.S.)*

[RFC4646] A. Phillips, M. Davis, "Étiquettes d'identification des langues", BCP0047 septembre 2006. *(Remplacée par* RFC5646)

[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)

[RFC5234] D. Crocker, P. Overell, "BNF augmenté pour les spécifications de syntaxe : ABNF", janvier 2008. (STD0068)

[RFC5425] F. Miao et autres, "Sécurité de la couche Transport (TLS) : transposition de Transport pour Syslog", mars 2009. *(P. S.)*

[RFC5426] A. Okmianski, "Transmission de messages Syslog sur UDP" mars 2009. *(P. S.)*

[UNICODE] Davis, M. et M. Suignard, "UNICODE Security Considerations", July 2005.

12.2 Références pour information

[RFC3164] C. Lonvick, "Protocole BSD de Syslog", août 2001. *(Information)*

Appendice A. Directives de mise en œuvre

Les informations de cette Section sont données comme une aide aux mises en œuvres. Bien que ces informations soient considérées comme utiles elles ne sont pas normatives. À ce titre, il N'EST PAS EXIGÉ d'une mise en œuvre qu'elle les suive afin de revendiquer la conformité à cette spécification.

A.1 Relations avec Syslog BSD

Bien que syslog BSD soit largement utilisé, son format n'a jamais été formellement normalisé. La [RFC3164] décrit les formats observés. C'est une RFC d'information, et la pratique montre qu'il y a de nombreuses mises en œuvre différentes. Les recherches durant la création de ce document ont montré qu'il y a très peu en commun entre les différentes mises en

œuvre de syslog sur les différentes plates-formes. La seule chose sur laquelle toutes s'accordent est que les messages commencent par "<" PRIVAL ">". À part cela, les messages syslog traditionnels ne sont pas formatés de façon cohérente. Par conséquent, la RFC 3164 ne décrit pas d'éléments spécifiques à l'intérieur d'un message syslog. Elle déclare que tout message destiné à l'accès UDP syslog doit être traité comme un message syslog, quel que soit son format ou contenu.

Le présent document retient la syntaxe et la sémantique de la valeur de PRI. Cela va permettre aux mises en œuvre syslog traditionnelles de mettre les messages générés par les applications syslog conformes à cette spécification dans les bonnes cases.

La plupart des mises en œuvre existantes prennent en charge UDP comme protocole de transport pour syslog. La présente spécification prend en charge le transport UDP, mais ne le recommande pas. Le déploiement du support TLS requis est recommandé. Des protocoles de transport supplémentaires peuvent être utilisés.

La RFC 3164 décrit le comportement du relais. Le présent document ne spécifie pas de comportement de relais. Cela pourrait être fait dans un document distinct.

Le TIMESTAMP décrit dans la RFC 3164 offre moins de précision que l'horodatage spécifié dans le présent document. Il lui manque aussi l'année et les informations de zone horaire. Si un message formaté conformément au présent document a besoin d'être reformaté pour être dans le format de la RFC 3164, il est suggéré que la zone horaire locale du générateur soit utilisée, et que les informations de zone horaire et d'année soit éliminées. Si un message dans le format de la RFC 3164 est reçu et doit être transformé pour être conforme au présent document, l'année en cours devrait être ajoutée et la zone horaire du relais ou collecteur PEUT être utilisée.

Le HOSTNAME de la RFC 3164 est moins spécifique, mais son format est encore pris en charge dans ce document comme une des représentations de remplacement de HOSTNAME.

La partie MSG du message est décrite comme TAG et CONTENT dans la RFC 3164. Dans le présent document, MSG est ce qui était appelé CONTENT dans la RFC 3164. Le TAG fait maintenant partie de l'en-tête, mais pas comme un seul champ. Le TAG a été partagé en APP-NAME, PROCID, et MSGID. Cela ne ressemble pas totalement à l'usage de TAG, mais fournit la même fonctionnalité dans la plupart des cas.

Dans la RFC 3164, les STRUCTURED-DATA n'étaient pas décrites. Si un message conforme au présent document contient des STRUCTURED-DATA et doit être reformaté selon la RFC 3164, les STRUCTURED-DATA deviennent simplement une partie du texte CONTENT de forme libre de la RFC 3164.

En général, le présent document essaye de fournir un en-tête facilement analysable avec de claires séparations de champs, tandis que le syslog BSD traditionnel souffre de règles de séparation de champ difficiles à analyser développées historiquement.

A.2 Longueur de message

Les auteurs de mises en œuvre devraient noter les limitations de taille de message mentionnées au paragraphe 6.1 et essayer de garder les données les plus importantes au début du message (dans la longueur minimum garantie). Cela assure que les données vont être vues par le collecteur ou relais même si un receveur de transport à un relais sur le chemin du message a tronqué le message.

La raison pour laquelle les receveurs de transport syslog ont seulement besoin de prendre en charge la réception de jusqu'à et inclus 480 octets a, entre autres choses, à voir avec les difficiles problèmes de livraison dans un réseau cassé. Les messages syslog peuvent utiliser une transposition de transport UDP avec cette restriction à 480 octets pour éviter des frais généraux de session et la fragmentation du message. Dans un réseau à problèmes, la probabilité d'avoir un message d'un seul paquet livré avec succès est plus forte que celle d'avoir deux fragments de message livrés avec succès. Donc, utiliser une plus grande taille peut empêcher l'opérateur d'obtenir des informations critiques sur le problème, alors qu'utiliser de petits messages pourrait donner ces informations à l'opérateur. Il est recommandé que les messages destinés à la correction de problèmes ne soient pas plus grands que 480 octets. Pour renforcer ce point, il a aussi été observé que certaines mises en œuvre UDP ne prennent généralement pas en charge des tailles de message de plus de 480 octets. Ce comportement est très rare et peut ne plus être un problème.

Il y a d'autres cas d'utilisation où les messages syslog sont utilisés pour transmettre des informations longues par nature, par exemple, des données d'audit. En n'appliquant pas de limite supérieure sur la taille de message, les applications syslog peuvent être mises en œuvre avec toute taille nécessaire et être quand même conformes au présent document. Dans ce cas,

il est de la responsabilité de l'opérateur de s'assurer que tous les composants d'une infrastructure syslog prennent en charge les tailles de message requises. Les transpositions de transport peuvent recommander des limites spécifiques de taille de message qui doivent être mises en œuvre pour être conforme.

On rappelle aux auteurs de mises en œuvre que la longueur des messages est spécifiée en octets. Il y a une différence potentiellement grande entre la longueur en caractères et la longueur en octets pour les chaînes UTF-8.

On doit noter que la MTU IPv6 est d'environ 2,5 fois 480. Une mise en œuvre qui cible un environnement seulement IPv6 pourrait donc supposer cela comme plus grande taille minimum.

A.3 Valeurs de sévérité

Ce paragraphe décrit des lignes directrices pour utiliser la sévérité mentionnée au paragraphe 6.2.1.

Toutes les mises en œuvre devraient essayer d'allouer la sévérité la plus appropriée à leurs messages. Plus important, les messages destinés à permettre le débogage ou l'essai de logiciel devraient recevoir la sévérité 7. La sévérité 0 devrait être réservée aux messages de très haute importance (comme de sérieuses défaillances du matériel ou une coupure de courant imminente). Un mise en œuvre peut utiliser les sévérités 0 et 7 pour d'autres fins si elle est configurée par l'administrateur.

Comme les sévérités sont très subjectives, un relais ou collecteur ne devrait pas supposer que tous les générateurs ont la même définition de sévérité.

A.4 Précision de TIME-SECFRAC

Le TIMESTAMP décrit au paragraphe 6.2.3 prend en charge les fractions de seconde. Cela donne lieu à des erreurs de codage très communes, où les zéros en tête sont retirés des secondes fractionnaires. Par exemple, l'horodatage "2003-10-11T22:13:14.003" peut être écrit de façon erronée "2003-10-11T22:13:14.3". Cela indiquerait 300 millisecondes au lieu des 3 millisecondes réellement désignées.

A.5 Convention de casse pour les noms

Les noms sont utilisés en divers endroits de ce document, par exemple pour les SD-ID et PARAM-NAME. Le présent document utilise de façon soutenue la "casse chameau minuscule". Avec elle, chaque nom commence par une lettre minuscule et chaque nouveau mot incorporé commence par une lettre majuscule, sans tiret ni autre délimiteur. Un exemple est "timeQuality".

Bien qu'une mise en œuvre soit libre d'utiliser toute autre convention de casse pour des noms expérimentaux, il est suggéré que la convention de casse mentionnée ci-dessus soit suivie.

A.6 Applications Syslog sans connaissance de l’heure

Au paragraphe 6.2.3, l'utilisation de la valeur NILVALUE a été permise pour les générateurs qui n'ont pas connaissance de l'heure. C'est fait pour prendre en charge le cas particulier où une application syslog n'a pas du tout connaissance de l'heure. On peut discuter qu'une telle application syslog puisse réellement être trouvée dans l'infrastructure de technologies de l'information d'aujourd'hui. Cependant, la discussion a indiqué que ces choses peuvent exister en pratique et qu'à ce titre il devrait y avoir des directives établies pour ce cas.

Cependant, un mise en œuvre DEVRAIT émettre un TIMESTAMP valide si le système d'exploitation sous-jacent, le système de programmation, et le matériel, supportent une fonction d'horloge. Un TIMESTAMP approprié devrait être émis même si il est difficile d'obtenir l'heure système. La NILVALUE devrait seulement être utilisée quand il est réellement impossible d'obtenir les informations d'heure. Cette règle ne devrait pas être utilisée comme excuse pour les mises en œuvre paresseuses.

A.7. Notes sur le SD-ID timeQuality

Il est recommandé que la valeur "0" soit celle par défaut pour le paramètre "tzKnown" (paragraphe 7.1.1). Il devrait seulement être changé en "1" après que l'administrateur a spécifiquement configuré la zone horaire. La valeur "1" peut être utilisée comme celle par défaut si le système d'exploitation sous-jacent fournit des informations précises de zone horaire. Il

est toujours conseillé que l'administrateur examine la correction des informations de zone horaire.

Il est important de ne pas créer une fausse impression de précision avec le SD-ID timeQuality (paragraphe 7.1). Un générateur devrait seulement indiquer une précision donnée si il sait réellement qu'il est dans ces limites. Il est généralement supposé que le générateur obtient une connaissance précise par la configuration de l'opérateur. Par défaut, aucune précision ne devrait être fournie.

A.8 Codage UTF-8 et BOM

Le présent document spécifie que les SD-PARAMS doivent toujours être codés en UTF-8. D'autres codages de message dans la portion MSG, incluant ASCIIPRINT, ne sont pas permis pour un appareil conforme à la présente spécification. Il y a deux cas qui doivent être traités ici. D'abord, une application syslog conforme à la présente spécification peut n'être pas capable de s'assurer que les informations qui lui sont données par un générateur sont codées en UTF-8. Si elle ne peut pas déterminer cela avec certitude, l'application syslog peut choisir de ne pas incorporer la BOM dans le MSG. Si l'application syslog a une bonne indication que le contenu du message est codé en UTF-8, alors elle devrait inclure la BOM. Dans le second cas, un relais syslog peut transmettre un message provenant d'un appareil qui ne se conforme pas à la présente spécification. Dans ce cas, l'appareil ne va probablement pas inclure la BOM sauf si il s'est assuré que le message reçu était codé en UTF-8.

Adresse de l’auteur

Rainer Gerhards
Adiscon GmbH
Mozartstrasse 21
Grossrinderfeld, BW 97950
Germany
mél : rgerhards@adiscon.com