Groupe de travail Réseau
Request for Comments : 3195
Catégorie : En cours de normalisation
Traduction Claude Brière de L'Isle

D. New & M. Rose
Dover Beach Consulting, Inc.
novembre 2001

Livraison fiable pour syslog

Statut de ce mémoire
Le présent document spécifie un protocole Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et des suggestions pour son amélioration. Prière de se reporter à l'édition actuelle du STD 1 "Normes des protocoles officiels de l'Internet" pour connaître l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé
Le protocole Syslog BSD décrit un certain nombre d'options de service qui se rapportent à la propagation de messages d'événement. Le présent mémoire décrit deux transpositions du protocole syslog sur des connexions TCP, toutes deux utiles pour la livraison fiable des messages d'événement. La première donne une transposition triviale qui maximise la rétro compatibilité. La seconde donne une transposition plus complète. Toutes deux fournissent un degré de robustesse et de sécurité dans la livraison du message qui est indisponible au protocole syslog usuel fondé sur UDP, en fournissant le chiffrement et l'authentification sur un protocole en mode connexion.

Table des Matières

1. Introduction

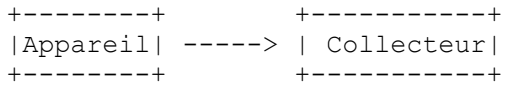
Le protocole syslog [RFC3164] présente un spectre d'options de service pour approvisionner un service de connexion fondé sur l'événement sur un réseau. Chaque option est associée à des avantages et des inconvénients. En conséquence, le choix d'approvisionner une combinaison d'options est une décision à la fois d'ingénierie et administrative. Le présent mémoire décrit comment réaliser le protocole syslog lorsque une livraison fiable est choisie comme service demandé. Il sort du domaine d'application du présent mémoire de discuter pour, ou contre, l'utilisation de la livraison fiable pour le protocole syslog.

Dans le présent document, les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMETE", "PEUT", et "FACULTATIF" sont à interpréter comme décrit dans le BCP 14, [RFC2119].

2. Modèle

Le service syslog prend en charge trois rôles de fonctionnement : appareil, relais, et collecteur.

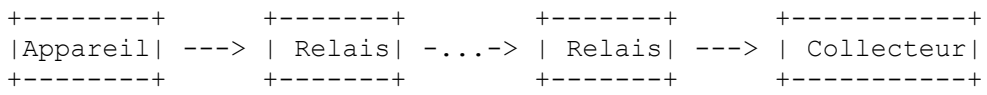
Appareils et collecteurs agissent respectivement comme source et comme réceptacle, des entrées syslog. Dans le cas le plus simple, seuls un appareil et un collecteur sont présents. Par exemple,



Les relations entre les appareils et les collecteurs sont potentiellement de plusieurs à plusieurs. C'est-à-dire qu'un appareil peut communiquer avec de nombreux collecteurs ; de même, un collecteur peut communiquer avec de nombreux appareils.

Un relais fonctionne selon les deux modes, acceptant des entrées syslog des appareils et autres relais et transmettant ces entrées aux collecteurs et autres relais.

Par exemple,



Comme le montre la figure, plus d'un relais peut être présent entre un appareil et un collecteur.

Un relais peut être nécessaire pour des raisons administratives. Par exemple, un relais peut fonctionner comme un mandataire d'application sur un pare-feu. Aussi, il peut y avoir un relais par département d'une entreprise, qui authentifie tous les appareils du département, et qui à son tour s'authentifie auprès d'un collecteur général de l'entreprise.

Un relais peut aussi servir à filtrer les messages. Par exemple, un relais peut collecter les informations syslog provenant d'un concentrateur serveur à l'échelle de la Toile toute entière, récapitulant les comptes de touches pour générer des rapports, transmettant les messages "page non trouvée" (qui indique une possible rupture de liaison) à un collecteur qui le présente au webmestre, et envoie des messages plus urgents (tels que les rapports de défaillance de matériel) à un collecteur qui les fait passer à un téléavertisseur. Un relais peut aussi être utilisé pour convertir les formats de sortie d'un appareil en entrées d'un collecteur.

On devrait noter que le rôle d'un appareil, relais ou collecteur, n'est pertinent que pour un certain canal BEEP (voir ci-dessous). Un seul serveur peut servir d'appareil, de relais, et de collecteur, tout à la fois, si il y est configuré. Il peut même servir comme relais et comme collecteur pour le même appareil en même temps en utilisant des canaux BEEP différents sur la même session en mode connexion ; cela peut être utile pour collecter des états et relayer des messages d'erreur urgents.

Pour assurer une livraison fiable lors de l'exécution du protocole syslog, le présent mémoire définit deux profils BEEP. BEEP [RFC3080] est un cadre générique de protocole d'application pour les interactions asynchrones en mode connexion. Au sein de BEEP sont fournies, à travers la retransmission, des caractéristiques telles que l'authentification, la confidentialité, et la fiabilité. Deux profils sont définis dans ce mémoire :

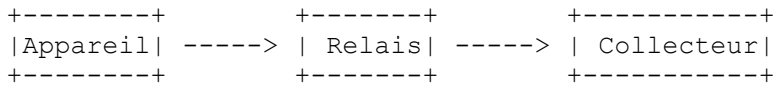
- Le profil RAW est conçu pour fournir de bonnes performances et un faible impact, en utilisant essentiellement le même format que le service syslog existant fondé sur UDP.
- Le profil COOKED est conçu pour fournir un format d'entrée structuré, dans lequel les entrées individuelles sont

acquittées (positivement ou négativement).

Noter que les deux profils fonctionnent sur BEEP. BEEP définit des "transpositions de transport", qui spécifient comment les messages BEEP sont portés sur les technologies de transport sous-jacentes. Au moment de la rédaction de ce document, un seul de ces transports est défini, dans la [RFC3081], qui spécifie BEEP sur TCP. Toutes les transpositions de transport sont obligées de prendre en charge une fiabilité et un séquençage suffisants pour permettre que tous les messages BEEP sur un certain canal soient livrés de façon fiable et dans l'ordre. Donc, les deux profils RAW et COOKED fournissent la livraison fiable de leurs messages.

Le choix du profil est indépendant des rôles opérationnels exposés plus haut.

Par exemple, la liaison appareil relais pourrait être configurée à utiliser le profil RAW, tandis que la liaison relais collecteur pourrait être configurée à utiliser le profil COOKED. (Par exemple, le relais peut analyser les messages syslog RAW à partir de l'appareil, sachant les détails de leurs formats, avant de les passer à un collecteur plus générique.) Bien sûr, le même appareil peut utiliser des profils différents, selon le collecteur auquel il envoie les entrées.



Les appareils et relais PEUVENT découvrir les relais et collecteurs via l'algorithme DNS SRV [RFC2782]. Si il est configuré à cela, le service utilisé est "syslog" et le protocole utilisé est "tcp". Cela permet une administration centrale de l'adressage, la récupération des relais et collecteurs en échec, et l'équilibrage de charge statique. Les politiques de sécurité et les configurations de matériel peuvent être telles que cette configuration d'appareil soit plus sûre que le serveur DNS. Les appareils peuvent avoir des ressources si limitées que l'accès DNS SRV soit inapproprié. Les pare-feu et autres mécanismes d'acheminement restrictifs peuvent devoir être traités avant qu'une connexion syslog fiable puisse être établie. Dans ce cas, le DNS peut n'être pas le mécanisme de configuration le plus approprié.

3. Profil RAW

3.1 Généralités sur le profil RAW

Le profil RAW est conçu pour un effort minimal de mise en œuvre, une forte efficacité, et une rétro compatibilité. Il est particulièrement approprié dans les cas où le traitement syslog traditionnel va être appliqué.

Il devrait être noté que bien que le profil RAW utilise le même format pour les charges utiles de message que la version UDP qu'utilise syslog, la livraison est fiable. Le profil RAW syslog est un profil de BEEP [RFC3080], et BEEP garantit une livraison fiable dans l'ordre des messages au sein de chaque canal individuel.

Lorsque le profil commence, aucune données portées ne sont fournies. Tous les messages BEEP dans le profil RAW sont spécifiés comme ayant un type de contenu MIME [RFC2046] d'application/flux d'octets. Une fois que le canal est ouvert, celui qui écoute (pas l'initiateur) envoie un message MSG indiquant qu'il est prêt à agir comme collecteur syslog. (Se reporter au paragraphe 2.1 de la [RFC3080] pour la discussion des rôles qu'un homologue BEEP peut effectuer, y compris les définitions des termes "écouteur", "initiateur", "client", et "serveur".)

L'initiateur utilise les réponses ANS pour fournir une ou plusieurs entrées syslog dans le format UDP courant, comme spécifié à la Section 3 de la [RFC3164]. Lorsque l'initiateur n'a plus d'entrées à envoyer, il termine avec une réponse NUL et ferme le canal.

Un exemple pourrait être comme suit :

```
L: <attente de connexion entrante>
I: <établissement de connexion>
L: RPY 0 0 . 0 201
L: Content-type: application/beep+xml
L:
L: <greeting>
L:   <profile
L:     uri='http://xml.resource.org/profiles/syslog/COOKED' />
L:   <profile uri='http://xml.resource.org/profiles/syslog/RAW' />
L: </greeting>
L: END
I: RPY 0 0 . 0 52
```

I: Content-type: application/beep+xml
I:
I: <greeting />
I: END
I: MSG 0 1 . 52 133
I: Content-type: application/beep+xml
I:
I: <start number='1'>
I: <profile uri='http://xml.resource.org/profiles/syslog/RAW' />
I: </start>
I: END
L: RPY 0 1 . 201 100
L: Content-type: application/beep+xml
L:
L: <profile uri='http://xml.resource.org/profiles/syslog/RAW' />
L: END
L: MSG 1 0 . 0 50
L:
L: Services centraux. Ceci n'a pas été un enregistrement.
L: END
I: ANS 1 0 . 0 61 0
I:
I: <29>Oct 27 13:21:08 ductwork imxpd[141]: Alerte de chauffage.END
I: ANS 1 0 . 61 58 1
I:
I: <29>Oct 27 13:22:15 ductwork imxpd[141]: Contacter Tuttle.END
I: NUL 1 0 . 119 0
I: END
L: MSG 0 3 . 301 70
L: Content-Type: application/beep+xml
L:
L: <close number='1' code='200' />
L: END
I: RPY 0 3 . 185 46
I: Content-Type: application/beep+xml
I:
I: <ok />
I: END
I: MSG 0 4 . 231 72
I: Content-Type: application/beep+xml
I:
I: <close number='0' code='200' />
I: END
L: RPY 0 4 . 371 46
L: Content-type: application/beep+xml
L:
L: <ok />
L: END
L: <ferme la connexion>
I: <ferme la connexion>
L: <attente de la prochaine connexion>

On voit ici une session BEEP établie, suivie par l'utilisation du profil RAW. L'initiateur est un appareil, alors que l'écouteur est un collecteur. L'initiateur ouvre le canal, mais l'écouteur envoie le premier MSG. Cela permet à l'initiateur d'envoyer un nombre quelconque de réponses ANS portant des messages d'événement syslog. L'initiateur envoie une réponse NUL pour indiquer qu'il a fini. À réception du NUL, l'écouteur ferme le canal RAW. L'initiateur a le choix de fermer la session BEEP entière ou d'ouvrir un nouveau canal syslog (RAW ou COOKED) pour d'autres transferts. Dans cet exemple, l'initiateur choisit de clore la session BEEP entière.

La redondance pour une trame ANS est d'environ trente octets, une fois que les prises de contact initiales ont été échangées. Si cette redondance est trop élevée, les messages vont alors être vraisemblablement générés à un rythme élevé. Dans ce cas, plusieurs messages syslog peuvent être agrégés en une seule trame ANS, chacune séparée par une séquence CRLF de la précédente. Le message final NE DOIT quand même PAS se terminer par un CRLF.

Par exemple,

```
L: MSG 1 0 . 0 50
L:
L: Services centraux. Ceci n'a pas été un enregistrement.
L: END
I: ANS 1 0 . 0 119 0
I:
I: <29>Oct 27 13:21:08 ductwork imxpd[141]: Alerte de chauffage.
I: <29>Oct 27 13:21:09 ductwork imxpd[141]: Contacter Tuttle.END
I: NUL 1 0 . 119 0
I: END
```

3.2 Identification et initialisation du profil RAW

Le profil RAW syslog est identifié par

http://xml.resource.org/profiles/syslog/RAW

dans l'élément BEEP "profile" durant la création de canal.

Aucune données ne sont portées durant la création de canal.

3.3 Syntaxe du message de profil RAW

Tous les messages BEEP dans ce profil ont un type de contenu MIME de application/flux d'octets. Le premier message BEEP de l'écouteur est ignoré et peut bien sûr être vide excepté les en-têtes ; donc, toute syntaxe est acceptable.

Les réponses ANS qu'envoie l'initiateur en réponse DOIVENT être formées conformément à la Section 4 de la [RFC3164]. En particulier, si le receveur agit comme relais, il DOIT suivre les règles établies au paragraphe 4.2.2 de la [RFC3164].

Si plusieurs messages syslog sont inclus dans une seule réponse ANS, chacune est séparée de la précédente par un CRLF. Il n'y a pas de délimiteur de fin, mais chaque longueur de corps de message d'événement syslog DOIT être de 1024 octets au plus, excluant la redondance de tramage BEEP. Noter qu'il NE DOIT PAS y avoir de CRLF entre le texte du message d'événement syslog final et la marque "END" de la queue de la trame BEEP.

3.4 Sémantique du message de profil RAW

Le message MSG d'ouverture de BEEP de l'écouteur n'a pas de sémantique. (C'est le bon endroit pour placer un message d'accueil identifiant.) Les réponses ANS de l'initiateur DOIVENT spécifier une facilité, sévérité, et un message textuel, comme décrit dans la [RFC3164].

4. Profil COOKED

4.1 Généralités sur le profil COOKED

Le profil COOKED est conçu pour les nouvelles mises en œuvre capables de traiter le protocole syslog. Il fournit une granularité beaucoup plus fine de l'étiquetage d'informations, permettant un plus haut degré d'automatisation du traitement. Naturellement, il comporte aussi à l'appui de cela un niveau de redondance plus important.

Le profil COOKED prend en charge trois éléments intéressants :

- L'élément "iam" identifie l'envoyeur pour le receveur, permettant à chaque homologue de se désigner auprès de l'autre, et de spécifier les rôles (appareil, relais, ou collecteur) que chacun joue.
- L'élément "entry" donne une version analysée de l'entrée syslog, avec la déclaration des divers champs intéressants.
- L'élément "path" identifie une liste de relais à travers lesquels est passée une collection étiquetée d'éléments "entry", ainsi qu'un ensemble de fanions indiquant quelles assurances de sécurité ont été effectives jusqu'à sa livraison.

4.2 Identification et initialisation du profil COOKED

Le profil syslog COOKED est identifié par

http://xml.resource.org/profiles/syslog/COOKED

dans l'élément BEEP "profile" durant la création de canal.

Durant la création de canal, l'élément "profile" correspondant dans l'élément BEEP "start" peut contenir un élément "iam". Si la création de canal réussit, avant d'envoyer la réponse correspondante, l'homologue BEEP traite alors l'élément "iam" et inclut la réponse résultante dans la réplique. Cette réponse sera un élément "ok" ou un élément "error". Le choix de l'élément retourné dépend du provisionnement local du receveur. L'inclusion d'un "iam" dans l'élément "start" initial a exactement la même sémantique que de le passer comme premier message MSG sur le canal.

4.3 Syntaxe de message du profil COOKED

Tous les messages BEEP dans ce profil ont un type de contenu MIME [RFC2046] de application/beep+xml. La syntaxe des éléments individuels est spécifiée à la Section 7.

4.4 Sémantique du message de profil COOKED

Les initiateurs produisent deux éléments : "iam" et "entry", chacun utilisant un message "MSG". L'écouteur produit un "ok" dans les messages "RPY" et "error" dans les messages "ERR". (Voir au paragraphe 2.3.1 de la [RFC3080] les définitions des éléments "error" et "ok".)

4.4.1 Élément IAM

L'élément "iam" sert à identifier un appareil, relais, ou collecteur à une extrémité du canal BEEP avec l'appareil, relais, ou collecteur à l'autre extrémité du canal. L'élément "iam" comporte le type de l'homologue (appareil, relais, ou collecteur) le nom de domaine pleinement qualifié de l'homologue, et l'adresse IP de l'homologue. (L'adresse IP choisie DEVRAIT être l'adresse IP associée au protocole de transport sous-jacent qui porte le canal.) Les données de caractères de l'élément sont du texte de forme libre lisible par l'homme. Elles peuvent être utilisées pour identifier l'homologue, comme en décrivant la localisation physique de la machine.

Un élément "iam" peut être utilisé par l'initiateur du canal à tout moment. L'écouteur répond à un élément "iam" par un "ok" (qui indique l'acceptation) ou par une "error" (qui indique le rejet). L'identité et le rôle effectifs sont spécifiés par le plus récent "iam" auquel on a répondu par un "ok".

Un "iam" pourrait être rejeté (avec un élément "error") par l'écouteur si la confidentialité ou l'authentification qui a été négociée est inadéquate ou si l'usager authentifié n'a pas l'autorisation de servir dans le rôle spécifié. On s'attend à ce que la plupart des installations exigent un "iam" de l'homologue avant d'accepter aucun message "entry".

Par exemple, une création réussie pourrait ressembler à ceci :

```
I: MSG 0 10 . 1832 259
I: Content-type: application/beep+xml
I:
I: <start number='1'>
I:   <profile
I:       uri='http://xml.resource.org/profiles/syslog/COOKED'>
I:     <![CDATA[ <iam fqdn='lowry.example.com' ip='10.0.0.27'
I:       type='device'/> ]]>
I:   </profile>
I: </start>
L: END
L: RPY 0 10 . 704 138
L: Content-type: application/beep+xml
L:
L: <profile uri='http://xml.resource.org/profiles/syslog/COOKED'>
L:   <![CDATA[ <ok /> ]]>
L: </profile>
L: END
```

Une création avec un "iam" incorporé qui échoue pourrait ressembler à ceci :

```
C: MSG 0 12 . 1832 259
C: Content-type: application/beep+xml
C:
C: <start number='1'>
C:   <profile
C:       uri='http://xml.resource.org/profiles/syslog/COOKED'>
C:     <![CDATA[ <iam fqdn='tuttle.example.com' ip='10.0.0.29'
C:       type='relay'/> ]]>
C:   </profile>
C: </start>
C: END
S: RPY 0 12 . 704 241
S: Content-type: application/beep+xml
S:
S: <profile uri='http://xml.resource.org/profiles/syslog/COOKED'>
S:   <![CDATA[
S:     <error code='535'>Usager 'buttle.example.com' non admis
S:       to "iam" for 'tuttle.example.com'</error> ]]>
S: </profile>
S: END
```

Dans ce cas, le code d'erreur indique que l'usager "buttle.example.com" s'est connecté via un profil SASL, mais la mise en œuvre de profil COOKED syslog prétend être "tuttle.example.com", une discordance que le serveur ne permet pas.

4.4.2 Élément ENTRY

L'élément "entry" porte les détails d'une seule entrée syslog. Les attributs d'un élément "entry" comportent "facility", "severity", "horodatage", "nom d'hôte", et "tag". "Facility" et "severity" ont la sémantique définie au paragraphe 4.1 de la [RFC3164]. Les autres attributs ont la sémantique des paragraphes 4.2.1 et 4.2.3 de la [RFC3164]. Un élément "entry" peut aussi contenir un attribut "pathID", décrit plus loin.

Si le client est un relais, l'élément "entry" DEVRAIT aussi contenir les attributs "deviceFQDN" et "deviceIP", qui spécifient le FQDN et l'adresse IP de l'appareil qui a créé l'entrée à l'origine. Ces attributs peuvent être ajoutés par le relais ou par l'appareil d'origine. Si possible, l'appareil DEVRAIT ajouter ces entrées, en se référant à l'interface la plus étroitement associée à l'entrée syslog. Avant qu'un relais transmette une entrée provenant d'un appareil qui ne porte pas ces attributs, il DEVRAIT les ajouter sur la base de l'élément "iam" qu'il a reçu de l'appareil, ou sur la base de l'adresse de connexion du transport sous-jacent. Un relais NE DOIT PAS ajouter ces champs si ils manquent et si un élément "iam" sur le canal a indiqué que les messages viennent d'un autre relais.

L'attribut "pathID" indique le chemin qu'a suivi cette entrée, passant de l'appareil à travers les relais jusqu'au collecteur final. Syntaxiquement, sa valeur est une chaîne de chiffres qui doivent correspondre à l'attribut "pathID" d'un élément "path" envoyé plus tôt sur le canal actuel. Sémantiquement, il indique que la liste des relais et fanions indiqués dans cet élément "path" précédent s'applique à cet élément "entry".

Les données de caractères pour cet élément sont le message d'événement syslog non structuré qui est l'objet de l'enregistrement. Si l'appareil d'origine livre le message pour la première fois via le profil COOKED, il peut avoir n'importe quelle structure au sein de CDATA. Cependant, pour une compatibilité maximale, l'appareil DEVRAIT formater le CDATA du message conformément aux paragraphes 4.2.1 à 4.2.3 de la [RFC3164].

Dans le message relayé, "tag" DEVRAIT être celle de l'appareil d'origine qui génère l'entrée (sauf si l'appareil ne peut pas fournir une étiquette). Le champ "horodatage" DEVRAIT être celui de l'heure de génération de l'entrée d'origine, plutôt que l'heure à laquelle l'entrée a été passée en sortie du relais. Le "nom d'hôte" DEVRAIT être le nom de l'hôte ou l'adresse IP par laquelle l'appareil se connaît lui-même ; ceci DOIT suivre les règles établies aux paragraphes 4.2.1 à 4.2.3 de la [RFC3164]. Le contenu d'origine du message syslog DOIT être préservé dans le CDATA de l'élément "entry" ; ceci inclut la préservation du contenu exact durant la traduction des formats UDP ou RAW. En particulier, les horodatages NE DOIVENT PAS être réécrits dans le CDATA de l'élément "entry", l'étiquette NE DOIT PAS être retirée du CDATA même si elle est présentée aussi dans les attributs "entry", et ainsi de suite.

Pour être cohérent avec l'esprit de la [RFC3164], un relais qui reçoit un message qui ne contient pas une priorité valide, un

horodatage ou nom d'hôte va suivre les mêmes règles générales que décrit au paragraphe 4.2.2 de la [RFC3164] tout en incluant le contenu exact du paquet syslog reçu comme CDATA. Les valeurs de la facilité et de la sévérité seront construites comme étant respectivement 8 et 6 et seront placées dans les attributs appropriés de l'élément "entry". Le nom d'hôte sera le nom de l'appareil comme il est connu du relais et sera aussi inséré dans les attributs de l'élément "entry". L'horodatage devrait être réglé à l'heure de réception, inséré seulement dans les attributs de l'élément "entry". Par exemple, considérons ce message reçu sur l'accès UDP 514 et interprété comme un message syslog traditionnel, en supposant que l'adresse de source IP sous-jacente est celle de la machine "pipeworks" :

```
<.....eeeek!
```

Pour être relayé, il doit être modifié comme suit :

```
C: MSG 1 0 . 2079 156
C: Content-Type: application/beep+xml
C:
C: <entry facility='8' severity='6'
C:   nom d'hôte='pipeworks'
C:   horodatage='Oct 31 23:59:59'
C:  >&amp;lt;.....eeeek!</entry>
C: END
S: RPY 1 0 . 933 45
S: Content-Type: application/beep+xml
S:
S: <ok/>
S: END
```

Dans un autre exemple, considérons un message reçu qui ne respecte pas correctement les conventions décrites au paragraphe 4.2.2 de la [RFC3164]. En particulier, l'horodatage a une année, donnant un format non standard :

```
<166> 1990 Oct 22 01:00:00 bomb tick[0]: BOOM!
```

Cela serait relayé comme suit :

```
C: MSG 1 0 . 2235 242
C: Content-Type: application/beep+xml
C:
C: <entry facility='160' severity='6'
C:   nom d'hôte='bomb'
C:   deviceFQDN='bomb.terrorist.net' deviceIP='10.0.0.83'
C:   horodatage='Oct 22 01:00:04'
C:  >&amp;lt;166> 1990 Oct 22 01:00:00 bomb tick[0]: BOOM!</entry>
C: END
S: RPY 1 0 . 978 45
S: Content-Type: application/beep+xml
S:
S: <ok/>
S: END
```

Noter que la valeur d'étiquette n'était pas directement apparente à partir du message reçu (à cause de l'échec de l'analyse de l'horodatage) de sorte qu'elle n'a pas été incluse dans l'élément "entry".

Il est explicitement permis à un relais d'analyser les messages bruts d'une façon plus sophistiquée, mais toutes les mises en œuvre DOIVENT être capables d'analyser les messages présentés dans le format décrit dans la [RFC3164]. Un relais plus sophistiqué pourrait avoir reconnu l'année et analysé complètement l'heure correcte, l'étiquette, et le nom d'hôte, mais une telle capacité d'analyse supplémentaire est FACULTATIVE.

À l'opposé, considérons l'exemple suivant :

```
<166> Oct 22 01:00:00 bomb tick[0]: BOOM!
```

Ce message conforme serait relayé comme suit :

```
C: MSG 1 0 . 2477 248
```

```
C: Content-Type: application/beep+xml
C:
C: <entry facility='160' severity='6'
C:   nom d'hôte='bomb'
C:   deviceFQDN='bomb.terrorist.net' deviceIP='10.0.0.83'
C:   horodatage='Oct 22 01:00:00' tag='tick'
C:  >&lt;166> Oct 22 01:00:00 bomb tick[0]: BOOM!</entry>
C: END
S: RPY 1 0 . 1023 45
S: Content-Type: application/beep+xml
S:
S: <ok/>
S: END
```

Dans ce cas, l'étiquette est détectée et l'horodatage représente l'heure de génération du message plutôt que l'heure de réception du message.

Finalement, l'élément "entry" peut aussi contenir un attribut "xml:lang", indiquant le langage dans lequel le contenu CDATA de l'étiquette est présenté, comme décrit dans la [RFC3066].

On répond à l'élément "entry" soit par un élément "ok" vide si tout a réussi, soit par un élément standard "error" si il y avait un problème. Un élément "entry" peut être rejeté si aucun élément "iam" n'a été accepté par l'écouteur. Il peut aussi être rejeté si l'usager authentifié dans la session BEEP (si il en est un) n'a pas l'autorité pour générer (comme un appareil) ou relayer cette entrée. Une erreur est aussi possible si l'attribut "pathID" se réfère à un élément "path" inconnu (ou rejeté).

Un échange réussi d'un élément "entry" peut ressembler à ceci :

```
C: MSG 1 0 . 2725 173
C: Content-Type: application/beep+xml
C:
C: <entry facility='24' severity='5'
C:   horodatage='Jan 26 15:16:17'
C:   nom d'hôte='pipework' tag='imxp'>
C:     No 27B/6 available</entry>
C: END
S: RPY 1 0 . 1068 45
S: Content-Type: application/beep+xml
S:
S: <ok/>
S: END
```

Ici, l'adresse IP et le FQDN de l'appareil sont tirés de l'élément "iam", si il en est un, ou des informations de connexion sous-jacente.

Un exemple où un élément "entry" est rejeté avec un élément "error" :

```
C: MSG 1 2 . 2898 223
C: Content-Type: application/beep+xml
C:
C: <entry facility='24' severity='5' horodatage='Jan 02 13:22:15'
C:   deviceFQDN='jack.example.net' deviceIP='10.0.0.83'
C:   tag='imxpd'>
C:     Appareil de remplacement trouvé dans la narine.
C: </entry>
C: END
S: ERR 1 2 . 1113 111
S: Content-Type: application/beep+xml
S:
S: <error code='554'>pas permis de relayer pour jack.example.net</error>
S: END
```

Ici, le client tente de relayer une entrée au nom de jack.example.com, mais l'entrée est refusée par le collecteur pour des raisons administratives. Cela peut arriver, par exemple, si lowry.example.com est dans un département différent de jack.example.com.

4.4.3 Élément PATH

L'élément "path" sert à décrire une liste de relais à travers lesquels cet élément est passé, ainsi qu'un ensemble de fanions qui indiquent les propriétés que partagent toutes les liaisons de l'appareil au relais. Chaque élément "path" contient soit un autre élément "path" soit est vide. Un élément "path" vide identifie un appareil, tandis qu'un élément "path" avec un élément "path" incorporé identifie un relais. Chaque élément "path" désigne un FQDN et une adresse IP de l'interface qui a envoyé l'élément. Chaque élément "path" désigne aussi un FQDN et une adresse IP pour l'interface qui a reçu l'élément. Chaque élément "path" porte aussi un attribut "linkprops" qui spécifie les propriétés de la liaison qu'il décrit.

Chaque élément "path" a un attribut "pathID" qui doit être unique pour tous les éléments "path" envoyés sur ce canal depuis son commencement. Syntaxiquement, l'attribut "pathID" est une chaîne de chiffres. Sémantiquement, il sert à identifier un élément "path" parmi de nombreux autres, et il sert à relier un élément "path" avec un ou plusieurs éléments "entry". Tout attribut "pathID" est sans relation avec tout attribut "pathID" dans les éléments "path" incorporés ou sur d'autres canaux.

Chaque élément "path" a un attribut "fromFQDN" et un attribut "fromIP". L'attribut "fromFQDN" DEVRAIT être le nom de domaine pleinement qualifié de l'interface sur laquelle l'élément "path" a été envoyé. (Le "fromFQDN" peut être omis si cette interface n'a pas d'entrée de DNS.) De même, l'attribut "fromIP" DOIT être l'adresse IP de l'interface sur laquelle l'élément "path" a été envoyé.

Chaque élément "path" a un attribut "toFQDN" et un attribut "toIP". L'attribut "toFQDN" DEVRAIT être le nom de domaine pleinement qualifié de l'interface sur laquelle l'élément "path" a été reçu. (Le "toFQDN" peut être omis si cette interface n'a pas d'entrée de DNS.) De même, l'attribut "toIP" DOIT être l'adresse IP de cette interface sur laquelle l'élément "path" a été reçu.

Finalement, chaque élément "path" porte un attribut "linkprops". Syntaxiquement, c'est une chaîne de caractères individuels, dont chacun indique une propriété du canal sur lequel cet élément "path" est porté. Noter que des éléments "path" externes peuvent avoir de plus fortes garanties que des éléments "path" internes ; il faut veiller à l'interprétation des fanions. La sémantique de chaque caractère possible dans cette chaîne est la suivante :

o : Lorsque présent, "o" (lettre minuscule "o") indique qu'une confidentialité faible a été négociée sur cette liaison, protégeant faiblement contre l'observation du contenu des entrées associées à cet élément "path". (La confidentialité faible est un chiffrement avec moins de 80 bits de clé.)

O : Lorsque présent, "O" (lettre majuscule "O") indique qu'une confidentialité forte a été négociée sur cette liaison, protégeant fortement contre l'observation du contenu des entrées associées à cet élément "path". (Une confidentialité forte est un chiffrement avec 80 bits de clé ou plus, ou un mécanisme de transfert qui est par ailleurs impossible à observer.)

U : Lorsque présent, "U" indique qu'un utilisateur valide a été authentifié (via SASL ou TLS) et un élément "iam" a été accepté.

A : Lorsque présent, "A" indique que cette liaison a été protégée par une couche d'authentification, authentifiant la source de toute "entrée" associée à ce chemin.

R : Lorsque présent, "R" indique que cette liaison a été protégée contre la répétition de message.

I : Lorsque présent, "I" indique que cette liaison a été protégée contre les modifications des messages dans le transfert. ("I" est l'abrégé pour Intégrité de message.)

L : Lorsque présent, "L" indique que cette liaison a été protégée contre la perte des messages. C'est-à-dire que c'est une liaison à livraison fiable.

D : Lorsque présent, "D" indique que le côté "from" de cette liaison est un appareil. Si il n'est pas présent sur l'élément "path" le plus interne, les éléments "entry" associés à ce chemin n'ont pas été portés par le profil COOKED pendant la totalité de leur durée de vie.

À réception d'un élément "path", l'homologue DOIT effectuer les vérifications suivantes :
- Les "fromFQDN" et "fromIP" doivent correspondre à la connexion de transport sous-jacente.
- Les fanions dans l'attribut "linkprops" doivent correspondre aux attributs de la session.
- Les "toFQDN" et "toIP" doivent correspondre à la connexion de transport sous-jacente.
- L'attribut "pathID" doit être unique par rapport à tous les autres éléments "path" reçus sur ce canal.

Si toutes ces vérifications réussissent, l'élément "path" est accepté avec un élément "ok". Autrement, un élément "error" est

généré avec un code approprié. De plus, si un des éléments "path" incorporés se réfère à la machine qui reçoit l'élément, il peut indiquer une boucle d'acheminement dans la configuration du chemin ainsi identifié, et les mesures appropriées devraient être prises.

Si l'homologue qui reçoit un élément "entry" le reçoit directement d'un appareil via un profil syslog-conn, et si l'appareil n'a pas généré un élément "path", le receveur peut générer lui-même un élément "path" approprié, soit à enregistrer dans les journaux (si cet homologue est un collecteur) soit à passer à l'homologue suivant (si cet homologue est un relais). Si un homologue reçoit un message syslog via UDP, il peut facultativement générer un élément "peer" approprié sur la base de toute information cryptographique fournie dans le message lui-même.

Lorsque un homologue reçoit un élément "path", il s'en souvient pour une utilisation ultérieure. Un collecteur va le mémoriser dans le journal pour référence ultérieure. Un relais va s'en souvenir. Lorsque une "entry" arrive qui fait référence à l'élément "path" reçu, et lorsque cette entrée a besoin d'être transmise à un autre relais ou collecteur, et lorsque aucun élément "path" approprié n'a encore été généré, un élément "path" approprié est généré et envoyé sur le canal sortant avant la transmission de l'entrée. Un élément "path" approprié est créé en prenant l'élément "path" reçu, en l'enveloppant dans un nouvel élément "path" avec les attributs appropriés, et en lui allouant un nouvel attribut "pathID". Lorsque de futurs éléments "entry" arrivent avec le même attribut "pathID" entrant, et si ils ont besoin d'être transmis à un canal sur lequel un attribut "pathID" approprié a déjà été envoyé, seul l'attribut "pathID" de l'élément "entry" doit être réécrit pour se référer à l'élément "path" sur le canal sortant.

On notera que la plus grande partie de la complexité de la gestion des éléments "path" ne vient que des relais. En particulier, les appareils n'ont jamais besoin de générer des éléments "path" et les collecteurs ont seulement besoin de les vérifier, de les enregistrer, et éventuellement de les utiliser dans les affichages et les rapports. Les collecteurs n'ont pas besoin de générer des éléments "path" ou de réécrire des éléments "entry". Donc, c'est seulement dans des configurations complexes (où ils sont les plus utiles) que surviennent des configurations complexes de "path".

Par exemple, soit un élément path envoyé de lowry.records.example.com à kurtzman.records.example.com. Il indique que les entrées de lowry à kurtzman étiquetées avec pathID='173' ont été générées par screen.lowry.records.example.com. Il indique que screen.lowry.records.example.com est estimé par lowry.records.example.com être l'appareil générateur, et que les entrées sur ce chemin sont livrées sans perte et sans modification, bien que les messages puissent être répétés ou observés. La liaison entre lowry et kurtzman, évite cependant les attaques en répétition, les pertes de message, et les modifications de message. Alors que screen.lowry.records.example.com ne s'est pas authentifié auprès de lowry.records.example.com, lowry prétend s'être authentifié lui-même auprès de kurtzman.

```
C: MSG 2 1 . 3121 426
C: Content-type: application/beep+xml
C:
C: <path fromFQDN='lowry.records.example.com'
C:      fromIP='10.0.0.50'
C:      toFQDN='kurtzman.records.example.com'
C:      toIP='10.0.0.51'
C:      linkprops='ULRI'
C:      pathID='173'>
C: <path fromFQDN='screen.lowry.records.example.com'
C:      fromIP='10.0.0.47'
C:      toFQDN='lowry.records.example.com'
C:      toIP='10.0.0.50'
C:      linkprops='DLI'
C:      pathID='24'>
C: </path>
C: </path>
C: END
S: ERR 2 1 . 1224 114
S: Content-type: application/beep+xml
S:
S: <error code='530'>linkprops inclut 'U' mais pas d''iam' reçu</error>
S: END
```

Cependant, kurtzman.records.example.com rejette l'élément "path", car l'attribut "linkprops" prétend que lowry s'est authentifié lui-même, mais kurtzman n'est pas d'accord, n'ayant pas reçu d'élément "iam".

Dans un second exemple, cet élément "path" informe collector.example.com que le pare-feu du département d'enregistrement va transmettre des éléments "entry" avec un attribut "pathID" dont la valeur est "17". Ces éléments

"entry" viendront sur l'interface "10.0.0.2" du pare-feu, pour être transmis en sortie sur l'interface "134.130.74.56" du pare-feu. Le bond final a toutes les garanties possibles, bien que les entrées transférées au sein du département des enregistrements (derrière le pare-feu) puissent avoir été observées au passage.

```
C: MSG 2 2 . 3547 813
C: Content-type: application/beep+xml
C:
C: <path fromFQDN='fwall.records.example.com'
C:      fromIP='134.130.74.56'
C:      toFQDN='collector.example.com'
C:      toIP='134.130.74.12'
C:      linkprops='OUARIL'
C:      pathID='17'>
C: <path fromFQDN='kurtzman.records.example.com'
C:      fromIP='10.0.0.50'
C:      toFQDN='fwall.records.example.com'
C:      toIP='10.0.0.2'
C:      linkprops='ULRI'
C:      pathID='120'>
C: <path fromFQDN='lowry.records.example.com'
C:      fromIP='10.0.0.50'
C:      toFQDN='kurtzman.records.example.com'
C:      toIP='10.0.0.51'
C:      linkprops='ULRI'
C:      pathID='173'>
C: <path fromFQDN='screen.lowry.records.example.com'
C:      fromIP='10.0.0.47'
C:      toFQDN='lowry.records.example.com'
C:      toIP='10.0.0.50'
C:      linkprops='DLI'
C:      pathID='24'>
C: </path></path></path></path>
C: END
S: RPY 2 2 . 1338 45
S: Content-type: application/beep+xml
S:
S: <ok/>
S: END
```

En exemple final, un élément "entry" venant de l'écran de Lowry arrive au pare-feu. L'attribut "path" est réécrit, et est transmis au collecteur.

L'entrée arrive sur l'interface 10.0.0.2 :

```
C: MSG 2 3 . 4360 250
C: Content-Type: application/beep+xml
C:
C: <entry facility='24' severity='5'
C:   horodatage='Oct 27 13:24:12'
C:   deviceFQDN='screen.lowry.records.example.com'
C:   deviceIP='10.0.0.47'
C:   pathID='173'
C:   tag='dvd'>
C:     On fait une pause – le patron surveille.
C: </entry>
C: END
S: RPY 2 3 . 1383 45
S: Content-Type: application/beep+xml
S:
S: <ok/>
S: END
```

Il est transmis sur l'interface 134.130.74.56 :

```
C: MSG 7 9 . 9375 276
C: Content-Type: application/beep+xml
C:
C: <entry facility='24' severity='5'
C:   horodatage='Oct 27 13:24:12'
C:   deviceFQDN='screen.lowry.records.example.com'
C:   deviceIP='10.0.0.47'
C:   pathID='17'
C:   tag='dvd'>
C:     On fait une pause – le patron surveille.
C: </entry>
C: END
S: RPY 7 9 . 338 45
S: Content-Type: application/beep+xml
S:
S: <ok/>
S: END
```

La discussion sur le bien fondé de configurer la machine de Lowry à transmettre de tels messages via la machine de Kurtzman sort du domaine d'application de ce document.

5. Dispositions supplémentaires

Dans des configurations plus évoluées, les appareils, relais et collecteurs syslog peuvent être configurés à prendre en charge diverses priorités de livraison. Plusieurs canaux fonctionnant sous le même profil peuvent être ouverts entre deux homologues, avec les messages syslog de priorité supérieure acheminés sur un canal auquel on donne plus de bande passante. Un tel provisionnement est une affaire locale.

Syslog [RFC3164] discute un certain nombre des raisons pour lesquelles la confidentialité et l'authentification des messages d'entrée de syslog peuvent être importantes dans un environnement d'informatique en réseau. La nature de BEEP permet une mise en couche pratique de l'authentification et de la confidentialité sur tout canal BEEP.

5.1 Authenticité de message

Le paragraphe 6.2 de la [RFC3164] discute des dangers des entrées syslog non authentifiées. Pour empêcher que soient acceptés des messages d'événements syslog non authentiques, on configure les homologues syslog à exiger l'utilisation d'une technologie d'authentification forte pour la session BEEP.

Si elle est provisionnée pour l'authentification du message, la mise en œuvre DEVRAIT utiliser le mécanisme SASL DIGEST-MD5 [RFC2831] pour provisionner ce service.

5.2 Répétition de message

Le paragraphe 6.3.4 de la [RFC3164] discute des dangers de la répétition du message syslog. Pour empêcher les messages d'événement syslog d'être répétés, on configure les homologues syslog à exiger l'utilisation d'une technologie d'authentification forte pour la session BEEP.

Si les mises en œuvre sont provisionnées pour détecter la répétition de message, elles DEVRAIENT utiliser le mécanisme SASL DIGEST-MD5 [RFC2831] pour fournir ce service.

5.3 Intégrité de message

Le paragraphe 6.5 de la [RFC3164] discute des dangers des messages d'événements syslog qui sont altérés par un attaquant malveillant. Pour empêcher d'altérer les messages, on configure les homologues syslog à exiger l'utilisation d'une technologie d'authentification forte pour la session BEEP.

Si les mises en œuvre sont provisionnées pour protéger l'intégrité du message, elles DEVRAIENT utiliser les mécanisme SASL DIGEST-MD5 [RFC2831] pour fournir ce service.

5.4 Observation de message

Le paragraphe 6.6 de la [RFC3164] discute des dangers (et des avantages) de rendre visible les messages syslog aux points intermédiaires le long du chemin de transmission entre appareil et collecteur. Pour empêcher les messages d'être vus par un attaquant, on configure les homologues syslog à exiger l'utilisation d'un profil de sécurité du transport pour la session BEEP. (Cependant, d'autres caractéristiques du trafic, par exemple, le volume et le rythme de transmissions, restent observables.)

Si les mises en œuvre sont provisionnées pour sécuriser les messages contre l'observation non autorisée, elles DEVRAIENT utiliser le profil TLS [RFC3080] pour fournir ce service. L'algorithme de chiffrement utilisé DEVRAIT être TLS_RSA_WITH_3DES_EDE_CBC_SHA.

5.5 Résumé des pratiques recommandées

Pour les protections indiquées, les mises en œuvre DEVRAIT être configurées à utiliser les mécanismes indiqués :

Protection désirée	**DEVRAIT être réglée en utilisant**
Authentification	http://iana.org/beep/SASL/DIGEST-MD5
+ Répétition	http://iana.org/beep/SASL/DIGEST-MD5
+ Intégrité	http://iana.org/beep/SASL/DIGEST-MD5
+ Observation	http://iana.org/beep/TLS

Les identités d'homologue BEEP utilisées pour l'authentification DEVRAIENT correspondre au FQDN de l'homologue initiateur. C'est à dire qu'un relais fonctionnant sur relay.example.com devrait utiliser un "identifiant d'utilisateur" de "relay.example.com" au sein des profils d'authentification SASL, ainsi que dans le FQDN de l'élément "iam".

6. Enregistrements initiaux

6.1 Enregistrement : profil RAW

Identification de profil : http://xml.resource.org/profiles/syslog/RAW
Messages échangés durant la création de canal : aucun
Messages commençant les échanges d'un à un : libre
Messages dans les réponses positives : aucun
Messages dans les réponses négatives : aucun
Messages dans les échanges d'un à plusieurs : libre
Syntaxe de message : voir au paragraphe 3.3
Sémantique de message : voir au paragraphe 3.4
Informations de contact : voir la Section "Adresse des auteurs" du présent mémoire

6.2 Enregistrement : profil COOKED

Identification de profil : http://xml.resource.org/profiles/syslog/COOKED
Messages échangés durant la création de canal : iam
Messages commençant les échanges d'un à un : iam, entry, path
Messages dans les réponses positives : ok
Messages dans les réponses négatives : error
Messages dans les échanges d'un à plusieurs : aucun
Syntaxe de message : voir au paragraphe 4.3
Sémantique de message : voir au paragraphe 4.4
Informations de contact : voir la Section "Adresse des auteurs" du présent mémoire

7. DTD syslog

La présente Section donne la déclaration de type de données (DTD, *Data Type Declaration*) qui définit les éléments valides pour la transposition de syslog sur BEEP.

<!-- DTD pour syslog sur BEEP, au 2000-10-10

On se réfère à cette DTD comme :

```
<!ENTITY % SYSLOG PUBLIC "-//Blocks//DTD SYSLOGRELIABLE//EN" ""> %SYSLOG; -->

<!-- Contenu
     Généralités
Inclut
  Résumés de profils
  Définitions d'entités

Opérations
         iam
         entry
         path
    -->

<!-- Généralités
   Paquets syslog livrés via BEEP
    -->

<!-- Inclut -->
        <!ENTITY % BEEP PUBLIC "-//Blocks//DTD BEEP//EN" ""> %BEEP;

<!--  Résumés de profils

Profil BEEP SYSLOG-RAW
rôle       MSG     ANS     ERR
  L        texte   texte   texte

Profil BEEP SYSLOG-COOKED
rôle       MSG     RPY     ERR
I ou L       iam   ok      error
I ou L       entry ok      error
I ou L       path  ok      error

-->

<!—Définitions d'entité

      entité                                  syntaxe/référence       exemple
Nom de domaine pleinement qualifié FQDN       Voir [RFC1034]          www.example.com
Adresse IP séparée par des points             1*3CHIFFRE "." 1*3CHIFFRE "." 1*3CHIFFRE "." 1*3CHIFFRE
         IP                                                           10.0.0.27
Facilité syslog
         FACILITY                             Voir la [RFC3164]
                                              1*3CHIFFRE              80
Sévérité syslog
         SEVERITY                             Voir la [RFC3164]
                                              CHIFFRE                 4
Horodatage                                    Voir la [RFC3164]       Jan 03 18:43:12
         TIMESTAMP
Entier identifiant IDINT                      1*CHIFFRE               1027

-->

<!ENTITY % FQDN          "CDATA">
<!ENTITY % IP            "CDATA">
<!ENTITY % FACILITY      "CDATA">
<!ENTITY % SEVERITY      "CDATA">
<!ENTITY % TIMESTAMP     "CDATA">
<!ENTITY % IDINT         "CDATA">

<!—L'élément iam déclare le rôle et l'identité de l'homologue qui le produit. Le contenu de l'élément peut inclure du texte
informatif lisible par l'homme, comme la localisation physique de l'ordinateur qui produit le "iam".  -->
```

```
<!ELEMENT    iam        (#PCDATA)>
<!ATTLIST    iam
             fqdn       %FQDN;                      #EXIGÉ
             ip         %IP;                        #EXIGÉ
             type       (appareil|relais|collecteur)  #EXIGÉ>

<!—L'élément entry porte un seul message syslog.  -->

<!ELEMENT    entry      (#PCDATA)>
<!ATTLIST    entry
             xml:lang   %LANG;       "i-default"
             facility   %FACILITY;   #EXIGÉ
             severity   %SEVERITY;   #EXIGÉ
             horodatage %TIMESTAMP;  #IMPLICITE
             tag        %ATEXT;      #IMPLICITE
             deviceFQDN %FQDN;       #IMPLICITE
             deviceIP   %IP;         #IMPLICITE
             pathID     %IDINT;      #IMPLICITE>

<!—L'élément path porte une liste des relais à travers lesquels les entrées sont passées.  -->

<!ELEMENT    path       (path?)>
<!ATTLIST    path
             pathID     %IDINT;      #EXIGÉ
             fromFQDN   %FQDN        #IMPLICITE
             fromIP     %IP;         #EXIGÉ
             toFQDN     %FQDN;       #IMPLICITE
             toIP       %IP;         #EXIGÉ
             linkprops  %ATEXT;      #EXIGÉ>

<!—Fin de la DTD -->
```

8. Codes de réponse

Les codes d'erreur suivants sont utilisés dans le protocole :

code	signification
200	réussite
421	service non disponible
451	action demandée interrompue (par exemple, erreur locale dans le traitement)
454	échec temporaire d'authentification
500	erreur générale de syntaxe (par exemple, XML mal formé)
501	erreur de syntaxe dans les paramètres (par exemple, XML non valide)
504	paramètre non mis en œuvre
530	authentification exigée
534	mécanisme d'authentification insuffisant (par exemple, trop faible, séquence épuisée, etc.)
535	échec d'authentification
537	action non autorisée pour l'usager
538	mécanisme d'authentification exigeant le chiffrement
550	action requise non effectuée (par exemple, aucun des profils demandés n'est acceptable)
553	paramètre invalide
554	échec de transaction (par exemple, violation de la politique)

9. Considérations relatives à l'IANA

9.1 Enregistrement : profils BEEP

L'IANA enregistre les profils spécifiés à la Section 6, et choisit les URL spécifiques de l'IANA "http://iana.org/beep/SYSLOG/RAW" et "http://iana.org/beep/SYSLOG/COOKED".

9.2 Enregistrement : numéro d'accès (bien connu) de système TCP pour syslog-conn

Un seul accès bien connu (601) est alloué à syslog-conn. La négociation dans la bande détermine si syslog-conn COOKED ou RAW est utilisé.

Numéro de protocole : TCP
Formats, types, Opcodes, et séquences de message : voir les paragraphes 3.3 et 4.4.
Fonctions : voir les paragraphes 3.3 et 4.4.
Utilisation de diffusion/diffusion groupée : aucune
Nom proposé : Service syslog fiable
Nom abrégé : syslog-conn
Informations de contact : voir la Section "Adresse des auteurs" du présent mémoire

10. Considérations sur la sécurité

Consulter la Section 6 de la [RFC3164] pour un exposé sur les questions de sécurité pour le service syslog. De plus, comme les profils RAW et COOKED sont définis en utilisant le cadre BEEP, consulter la Section 8 de la [RFC3080] pour un exposé sur les questions de sécurité spécifiques de BEEP.
BEEP est utilisé pour assurer la sécurité de la communication mais pas l'intégrité de l'objet. En d'autres termes, le message "en vol" peut être protégé, mais un appareil compromis peut générer de façon indétectable des messages incorrects, et les relais et collecteurs peuvent modifier, insérer, ou supprimer les messages sans être détectés. D'autres techniques doivent être utilisées pour assurer que de telles compromissions sont détectables.

11. Remerciements

Les auteurs remercie de leurs contributions Christopher Calabrese, Keith McCloghrie, Balazs Scheidler, et David Waitzman.

12. Références

[RFC2046] N. Freed et N. Borenstein, "Extensions de messagerie Internet multi-objets (MIME) Partie 2 : Types de support", novembre 1996. *(D. S., MàJ par* 2646, 3798, 5147, 6657*.)*

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

[RFC2782] A. Gulbrandsen, P. Vixie et L. Esibov, "Enregistrement de ressource DNS pour la spécification de la localisation des services (DNS SRV)", février 2000.

[RFC2831] P. Leach et C. Newman, "Utilisation de l'authentification par résumé comme mécanisme SASL", mai 2000. (*Obsolète, voir* RFC6331)

[RFC3066] H. Alvestrand, "Étiquettes pour l'identification des langues", BCP 47, janvier 2001. *(Remplacée par* RFC4646*.)*

[RFC3080] M. Rose, "Cœur du protocole extensible d'échange de blocs (BEEP)", mars 2001. *(P.S.)*

[RFC3081] M. Rose, "Transposition du cœur BEEP en TCP", mars 2001. *(P.S.)*

[RFC3164] C. Lonvick, "Protocole BSD de Syslog", août 2001. *(Remplacée par* RFC5424*, Information)*

Adresses des auteurs

Darren New
5390 Caminito Exquisito
San Diego, CA 92130
USA
téléphone : +1 858 350 9733
mél : dnew@san.rr.com

Marshall T. Rose
Dover Beach Consulting, Inc.
POB 255268
Sacramento, CA 95865-5268
USA
téléphone : +1 916 483 8878
mél : mrose@dbc.mtview.ca.us

Déclaration complète de droits de reproduction

Remerciement
Le financement de la fonction d'édition des RFC est actuellement fourni par l'Internet Society.