Groupe de travail Réseau
Request for Comments : 5235
RFC rendue obsolète : 3685
Catégorie : Sur la voie de la normalisation

C. Daboo

janvier 2008
Traduction Claude Brière de L'Isle

Filtrage de messagerie Sieve : extensions Spamtest et Virustest

Résumé
Les extensions "spamtest", "spamtestplus", et "virustest" au langage de filtrage de messagerie Sieve permettent aux utilisateurs d'utiliser des commandes simples et portables pour les essais sur la présence de pourriels et de virus sur les messages électroniques. Chaque extension fournit un nouvel essai utilisant des confrontations à des "scores" numériques. Il est de la responsabilité de la mise en œuvre Sieve sous-jacente de faire les vérifications réelles qui résultent en des entrées appropriées aux essais.

Table des Matières

1. Introduction et vue d'ensemble

Les scripts Sieve sont fréquemment utilisés pour faire le filtrage de pourriels et virus soit sur la base d'essais de script implicites (par exemple, vérifications d'envoyeurs en "liste noire" directement codés dans le script Sieve) ou via des messages d'essais modifiés par un vérificateur externe de pourriels ou virus qui traite le message avant Sieve. L'utilisation d'outils tiers de vérification de pourriels et virus pose un problème car chaque outil a sa propre façon d'indiquer le résultat de ses vérifications. Elles prennent généralement la forme d'un en-tête ajouté au message, dont le contenu indique l'état en utilisant une syntaxe définie par l'outil particulier. Chaque utilisateur doit alors créer son propre script Sieve pour correspondre au contenu de ces en-têtes pour faire le filtrage. Cela exige que le script reste synchronisé avec l'outil tiers lorsque il est mis à jour ou peut-être remplacé par un autre. Donc, les scripts sont liés à un environnement spécifique et perdent la portabilité.

L'objet du présent document est d'introduire deux essais Sieve qui peuvent être utilisés pour mettre en œuvre des essais "génériques" pour les pourriels et les virus dans les messages traités via des scripts Sieve. Les vérifications de pourriels et de virus elles-mêmes s'appuient sur la mise en œuvre sous-jacente de Sieve de toute manière appropriée, afin que les commandes Sieve de vérifications de pourriels et de virus puissent être utilisées d'une façon portable.

Afin de faire des comparaisons numériques sur les chaînes retournées, les mises en œuvre de serveur DOIVENT aussi prendre en charge l'extention Sieve Relational [RFC5231], en plus des extensions décrites ici. Tous les exemples ci-dessous supposent que l'extension "Relational" est présente.

2. Conventions utilisées dans ce 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].

Les conventions de notation sont celles du paragraphe 1.1 de la [RFC5228]

Le terme de "pourriel est utilisé dans le présent document pour se référer à des messages électroniques non sollicités ou indésirables. Le présent document ne tente pas de définir ce qui constitue exactement un pourriel, ou comment il devrait être identifié, ou quelles actions devraient être prises quand il est détecté.

Le terme "virus" est utilisé dans le présent document pour se référer à tout type de message dont le contenu peut causer des dommages par malveillance. Le présent document ne tente pas de définir ce qui constitue exactement un virus, ou comment il devrait être identifié, ou quelles actions devraient être prises quand il est détecté.

3. Extensions Sieve

3.1 Considérations générales

Les essais "spamtest" et "virustest" décrits ci-dessous évaluent les résultats des vérifications spécifiques de mise en œuvre de pourriels et de virus de façon portable. La mise en œuvre peut, par exemple, chercher des en-têtes d'outils tiers de pourriel et déterminer comment ils se transposent dans la façon dont les commandes d'essai sont utilisées. Pour faire cela, la mise en œuvre sous-jacente de Sieve fournit une chaîne de résultat normalisée comme une des entrées à chaque commande d'essai. La chaîne de résultat normalisée est considérée comme étant la valeur du côté gauche de l'essai, et les valeurs de comparaison données dans la commande de l'essai sont considérées comme étant du côté droit.

Le résultat normalisé commence par une chaîne de chiffres, avec sa valeur numérique dans la gamme des valeurs utilisées par l'essai spécifique, indiquant la sévérité du pourriel ou des virus dans un message ou si des essais ont bien été effectués. Cela peut facultativement être suivi par un caractère espace (%x20) et du texte arbitraire, ou dans un cas spécifique un seul mot clé est retourné. La valeur numérique peut être comparée à des valeurs spécifiques utilisant l'extension Sieve "Relational" [RFC5231] conjointement avec le comparateur "i;ascii-numeric" de la [RFC4790], qui va vérifier la présence d'une valeur numérique au début de la chaîne, en ignorant tout texte supplémentaire dans la chaîne. Le texte facultatif peut être utilisé pour porter des détails spécifiques de la mise en œuvre sur les essais et des commentaires descriptifs sur le résultat. Les essais peuvent être faits en utilisant les comparateurs standard de chaîne par rapport à ce texte si il aide à préciser le comportement ; cependant, cela va casser la portabilité du script car le texte va probablement être spécifique d'une mise en œuvre particulière.

De plus, le type de correspondance ":count" de l'extension Sieve Relational [RFC5231] peut être utilisé pour déterminer si la mise en œuvre sous-jacente a réellement fait un essai. Si l'essai de pourriel ou virus sous-jacent a été fait, le ":count" du résultat normalisé va retourner la valeur numérique "1", tandis que si l'essai n'a pas été fait, ou si la mise en œuvre de Sieve n'a pas pu déterminer si un essai a été fait ou non, la valeur de ":count" va être "0" (zéro).

3.2 Essai spamtest

Usage : spamtest [":percent"] [COMPARATOR] [MATCH-TYPE] <valeur : chaîne>

Les mises en œuvre de Sieve qui utilisent l'essai "spamtest" se servent d'un identifiant de "spamtest" ou "spamtestplus" à utiliser avec le mécanisme de capacité.

Si l'argument ":percent" n'est pas utilisé avec un essai spamtest, alors un des identifiant de capacité "spamtest" ou "spamtestplus", ou les deux, DOIT être présent.

Si l'argument ":percent" est utilisé avec un essai spamtest, alors l'identifiant de capacité "spamtestplus" DOIT être présent. Les mises en œuvre de Sieve DOIVENT retourner une erreur si l'argument ":percent" est utilisé et "spamtestplus" n'est pas spécifié.

Pour être brefs et clairs, les scripts NE DEVRAIENT PAS spécifier les deux identifiants de capacité "spamtestplus" et "spamtest" ensemble.

L'essai "spamtest" s'évalue à vrai si le résultat normalisé de spamtest correspond à la valeur. Le type de correspondance est spécifié par l'argument de correspondance facultatif, qui est ":is" par défaut si il n'est pas spécifié.

3.2.1 spamtest sans argument :percent

Quand l'argument ":percent" n'est pas présent dans l'essai "spamtest", la chaîne de résultat normalisée fournie pour le côté gauche de l'essai commence par une valeur numérique dans la gamme de "0" (zéro) à "10" (dix), dont la signification est résumée ci-dessous :

Valeur de spamtest	Interprétation
0	message non testé pour pourriel, ou Sieve n'a pas pu déterminer si un essai a été fait.
1	message testé et n'est pas un pourriel.
2 à 9	message testé et peut contenir un pourriel ; un numéro plus fort indique une plus forte probabilité de pourriel.
10	message testé et contient à coup sûr un pourriel.

La mise en œuvre sous-jacente de Sieve va transposer toute vérification de pourriel effectuée dans cette gamme numérique, comme approprié.

Exemple :

```
require ["spamtest", "fileinto", "relational", "comparator-i;ascii-numeric"];
        si spamtest :value "eq" :comparator "i;ascii-numeric" "0"
        {
           fileinto "INBOX.unclassified";
        }
        autrement si spamtest :value "ge" :comparator "i;ascii-numeric" "3"
        {
           fileinto "INBOX.spam-trap";
        }
```

Dans cet exemple, tout message qui n'est pas passé par un outil de vérification de pourriel va être dirigé sur la boîte aux lettres "INBOX.unclassified". Tout message avec une valeur de résultat normalisé supérieure ou égale à "3" est dirigé sur une boîte aux lettres appelée "INBOX.spam-trap" dans le magasin de messages de l'utilisateur.

3.2.2 spamtest avec argument :percent

Quand l'argument ":percent" est présent dans l'essai "spamtest", la chaîne de résultat normalisé fournie pour le côté gauche de l'essai commence par une valeur numérique dans la gamme "0" (zéro) à "100" (cent), dont la signification est résumée ci-dessous :

Valeur de spamtest	Interprétation
0	message testé et sans pourriel, ou non testé pour pourriel, ou Sieve n'a pas pu déterminer si un essai a été fait.
1 à 99	message testé et peut contenir un pourriel ; un pourcentage élevé indique une plus grande probabilité de pourriel.
100	message testé et contient bien un pourriel.

La mise en œuvre sous-jacente de Sieve va transposer toute vérification de pourriel effectuée dans cette gamme numérique, comme approprié.

Pour déterminer si le message a ou non été testé comme pourriel, deux options peuvent être utilisées :

a. un essai avec ou sans l'argument ":percent" et le type de correspondance ":count", testant pour la valeur "0" comme décrit au paragraphe 3.1.
b. un essai sans l'argument ":percent" utilisant le type de correspondance ":value", testant pour la valeur de résultat normalisé "0" comme décrit au paragraphe 3.2.1.

Exemples :

```
require ["spamtestplus", "fileinto", "relational", "comparator-i;ascii-numeric"];
        si spamtest :value "eq" :comparator "i;ascii-numeric" "0"
        {
           fileinto "INBOX.unclassified";
        }
        autrement si spamtest :percent :value "eq" :comparator "i;ascii-numeric" "0"
        {
           fileinto "INBOX.not-spam";
        }
        autrement si spamtest :percent :value "lt" :comparator "i;ascii-numeric" "37"
        {
           fileinto "INBOX.spam-trap";
        }
         autrement
        {
           éliminer ;
        }
```

Dans cet exemple, tout message qui n'a pas réussi à passer à travers un outil de vérification de pourriel va être envoyé dans la boîte aux lettres "INBOX.unclassified". Tout message qui est classé comme ne contenant définitivement pas de pourriel (valeur de résultat normalisé de "0") va être mis dans la boîte aux lettres "INBOX.not-spam". Tout message avec une valeur de résultat normalisé inférieure à "37" est mis dans une boîte aux lettres appelée "INBOX.spam-trap" dans le magasin de messages de l'utilisateur. Toute autre valeur de résultat normalisé va résulter en l'élimination du message.

Autrement, le type de correspondance de Sieve Relational [RFC5231] ":count" peut être utilisé :

Exemple :

```
        si spamtest :percent :count "eq" :comparator "i;ascii-numeric" "0"
        {
           fileinto "INBOX.unclassified";
        }
        autrement si spamtest :percent :value "eq" :comparator "i;ascii-numeric" "0"
        {
           fileinto "INBOX.not-spam";
        }
        autrement si spamtest :percent :value "lt" :comparator "i;ascii-numeric" "37"
        {
           fileinto "INBOX.spam-trap";
        }
         autrement
        {
           éliminer ;
        }
```

Cet exemple va avoir pour résultat exactement le même comportement que le précédent.

3.3 Essai virustest

Usage : virustest [COMPARATOR] [MATCH-TYPE] <valeur: chaîne>

Les mises en œuvre de Sieve qui prennent en charge l'essai "virustest" ont un identifiant de "virustest" à utiliser avec le mécanisme de capacité.

L'essai "virustest" s'évalue à vrai si la chaîne de résultat normalisé correspond à la valeur. Le type de correspondance est spécifié par l'argument de correspondance facultatif, qui est par défaut ":is" si il n'est pas spécifié.

La chaîne de résultat normalisé fournie pour le côté gauche de l'essai commence par une valeur numérique dans la gamme de "0" (zéro) à "5" (cinq), dont la signification est résumée ci-dessous :

Valeur de virustest	Interprétation
0	message non testé pour des virus, ou Sieve n'a pas pu déterminer si un essai a été fait.
1	message testé et ne contient pas de virus connu.
2	message testé et contenait un virus connu qui a été remplacé par un contenu inoffensif.
3	message testé et contenait un virus connu qui a été "curé" de sorte qu'il est maintenant inoffensif.
4	message testé et pourrait contenir un virus connu.
5	message testé et contient à coup sûr un virus connu.

La mise en œuvre sous-jacente de Sieve va se transposer en toute vérification de virus faite dans cette gamme numérique, comme approprié. Si le message n'a pas été catégorisé par un outil de vérification de virus, alors le résultat de virustest est "0".

Exemple :

```
require ["virustest", "fileinto", "relational", "comparator-i;ascii-numeric"];
        si virustest :value "eq" :comparator "i;ascii-numeric" "0"
        {
           fileinto "INBOX.unclassified";
        }
        si virustest :value "eq" :comparator "i;ascii-numeric" "4"
        {
           fileinto "INBOX.quarantine";
        }
        autrement si virustest :value "eq" :comparator "i;ascii-numeric" "5"
        {
           éliminer ;
        }
```

Dans cet exemple, tout message qui n'a pas réussi à l'examen d'un outil de vérification de virus va être dirigé sur la boîte aux lettres "INBOX.unclassified". Tout message avec une valeur de résultat normalisé égale à "4" est dirigé sur une boîte aux lettres appelée "INBOX.quarantine" dans le magasin de messages de l'utilisateur. Tout message avec une valeur de résultat normalisé égale à "5" est éliminé (supprimé) et non livré au magasin de messages de l'utilisateur.

4. Considérations sur la sécurité

Les mises en œuvre de Sieve DEVRAIENT s'assurer que les essais "spamtest" et "virustest" rapportent seulement des résultats de vérifications de pourriels et de virus pour les messages qui sont réellement passés par un processus légitime de vérification de pourriel ou virus. En particulier, si de telles vérifications s'appuient sur l'ajout et la vérification ultérieure de champs d'en-tête privés, il est de la responsabilité de la mise en œuvre de s'assurer que ces en-têtes ne peuvent pas être usurpés par l'envoyeur ou un intermédiaire et par là empêcher que la mise en œuvre soit trompée pour retourner de faux résultats de l'essai.

Les administrateurs de serveurs doivent s'assurer que les outils de vérification de virus sont à jour, pour fournir une protection raisonnable aux utilisateurs de l'essai "virustest". Les utilisateurs devraient être conscients du fait que l'essai "virustest" ne donne pas une assurance à 100 % de la suppression de tous les virus, et qu'ils devraient continuer d'être vigilants lorsque ils traitent des messages de contenu et origine inconnus.

À part cela, les extensions "spamtest" et "virustest" ne soulèvent aucun probème de sécurité qui ne soit déjà présent dans le protocole de base [RFC5228], et ces questions sont discutées dans la [RFC5228].

5. Considérations relatives à l'IANA

Les gabarits suivants spécifient l'enregistrement par l'IANA des extensions Sieve spécifiées dans ce document. Les enregistrements pour "spamtest" et "virustest" remplacent ceux de la [RFC3685] :

5.1 Enregistrement de spamtest

Pour : iana@iana.org
Sujet : enregistrement d'une nouvelle extension Sieve.
Nom de capacité : spamtest
Description : fournit un essai pour vérifier la probabilité qu'un message électronique soit un pourriel.
RFC publiée : RFC 5235
Adresse de contact : liste de diffusion Sieve à <ietf-mta-filters@imc.org>

Ces informations ont été ajoutées à la liste des extensions Sieve à http://www.iana.org/assignments/sieve-extensions.

5.2 Enregistrement de virustest

Pour : iana@iana.org
Sujet : enregistrement d'une nouvelle extension Sieve.
Nom de capacité : virustest
Description : fournit un essai pour vérifier la probabilité qu'il y ait un contenu malveillant dans un message électronique.
RFC publiée : RFC 5235
Adresse de contact : liste de diffusion Sieve à <ietf-mta-filters@imc.org>

Ces informations ont été ajoutées à la liste des extensions Sieve à http://www.iana.org/assignments/sieve-extensions.

5.3 Enregistrement de spamtestplus

Pour : iana@iana.org
Sujet : enregistrement d'une nouvelle extension Sieve.
Nom de capacité : spamtestplus
Description : fournit un essai pour vérifier la probabilité qu'un message électronique soit un pourriel, en utilisant une gamme de pourcentages.
RFC publiée : RFC 5235
Adresse de contact : liste de diffusion Sieve à <ietf-mta-filters@imc.org>

Ces informations ont été ajoutées à la liste des extensions Sieve à http://www.iana.org/assignments/sieve-extensions.

6. Références

6.1 Références normatives

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

[RFC4790] C. Newman et autres, "Registre de collation des protocoles d'application de l'Internet", mars 2007. *(P.S.)*

[RFC5228] P. Guenther et autres, "Sieve : un langage de filtrage de messagerie électronique", janvier 2008. (*P.S. ; Remplace* RFC3028, *MàJ par* RFC5229, 5429, 9042*)*

[RFC5231] W. Segmuller, B. Leiba, "Filtrage de messagerie Sieve : extension Relational", janvier 2008. (*Remplace* RFC3431)*(P.S.)*

6.2 Références pour information

[RFC3685] C. Daboo, "Filtrage de messagerie SIEVE : Extensions Spamtest et VirusTest", février 2004. (*Obsolète, voir* RFC5235) *(P.S.)*

Appendice A. Remerciements

Merci à Mark E. Mallett, Tony Hansen, Jutta Degener, Ned Freed, Ashish Gawarikar, Alexey Melnikov, Nigel Swinson, et Aaron Stone de leurs commentaires et corrections.

Appendice B. Changements importants par rapport à la RFC 3685

Voici la liste des changements majeurs par rapport à la précédente spécification [RFC3685], que celle-ci remplace.

1. Un argument ":percent" a été ajouté à l'essai "spamtest", ajoutant une nouvelle gamme numérique de 0 à 100 pour les résultats d'essais.

2. Un "spamtestplus" exige un élément qui a été ajouté pour indiquer la présence de cette extension dans les scripts.

3. Le type de correspondance "count" de la [RFC5231] peut maintenant être utilisé pour déterminer si un message a été ou non testé.

4. Précisé que "essai non fait" signifie aussi "Le système Sieve n'a pas pu déterminer si un essai a été fait".

Adresse de l'auteur

Cyrus Daboo
mél : cyrus@daboo.name

Déclaration complète de droits de reproduction