


26 juil 2008

# Lutter contre le spam avec Postfix

Catégorie : Administration système Tags : GLMF

~~Retrouvez cet article dans : Linux Magazine 89~~

Dans les articles précédents, nous avons abordé la sécurisation interne d'un système de messagerie avec Postfix ainsi que la problématique du routage du courrier. Nous en venons ici à la lutte contre le courrier indésirable.
Cet article peut être lu indépendamment des deux précédents.
Toutefois, il fait appel à des notions déjà présentées précédemment.

## Classification du courrier indésirable

Pour bien appréhender la lutte contre le courrier indésirable, nous allons en établir un classement. Du point du vue de la provenance, on peut établir trois catégories :

- **interne :** courrier indésirable provenant de l'intérieur du réseau. Il peut s'agir du courrier généré par un virus ou tout simplement d'un utilisateur indélicat ;
- **externe :** le traditionnel flot d'attaques par virus ou spam ;
- **fausses notifications :** certains robots utilisent de fausses adresses sources calculées à partir de dictionnaires ou de carnets d'adresses piratés. Si l'adresse source du message non désiré semble provenir d'un de nos domaines, et suivant la configuration du site attaqué, nous pouvons recevoir un avis de non remise.

Les catégories interne et externe peuvent, elles aussi, être subdivisées : le client IP peut être un virus ou un serveur SMTP licite ayant accepté un message indésirable.
En termes d'effet, la classification est assez simple :

- **spam :** les publicités ;
- **exécutables dangereux standards :** la catégorie des virus, chevaux de Troie, et autres codes

malveillants existant sur le marché ;
- **exécutables dangereux spécifiques :** nous classons ici les éventuels codes malveillants qui auraient été écrits spécialement pour nous attaquer dans le cadre par exemple de l'espionnage industriel.

# Insertion d'un dispositif d'examen du contenu dans Postfix

Postfix est avant tout un MTA, il n'a donc pas été écrit pour rechercher les virus ou le spam. En revanche, sa structure modulaire lui permet d'accueillir, dans son flux de message, les logiciels spécialisés dans le filtrage.
Il existe plusieurs façons d'intégrer un dispositif d'examen du contenu dans Postfix. Commençons par la moins bonne :

- Insertion du dispositif en coupure sur le réseau : dans ce cas, le courrier arrive directement sur l'équipement ou le logiciel qui le traite et le renvoie à Postfix. L'inconvénient majeur de ce dispositif est que l'attaquant est directement renseigné sur l'équipement utilisé (par la bannière par exemple). D'autre part, comme tous ces équipements sont plus ou moins vulnérables aux attaques de type déni de service (par débordement de la mémoire disponible par exemple), nous risquons une indisponibilité du service de messagerie. L'attaquant peut alors chercher à nous faire passer en mode dégradé, car, en général, quand on ne comprend pas pourquoi ça ne marche pas, on débranche les équipements de sécurité :-(
- Insertion d'un dispositif dans le système Postfix traitant à la volée le courrier entrant. Postfix est ainsi apte à rejeter le courrier indésirable pendant la livraison ce qui évite la génération des éventuels avis de non-remise, mais risque de ralentir le traitement des messages. Dans un site soumis à une forte charge, cette méthode est déconseillée pour un traitement lourd, car le client connecté risque de ne pas recevoir assez vite la réponse de bonne réception et tentera de multiples fois la même livraison. Il existe deux façons de faire cette intégration : en proxy SMTP ou avec le protocole MILTER [1] de la communauté Sendmail.
- Insertion d'un dispositif après mise en file d'attente : Postfix accepte le courrier sans effectuer ce contrôle, puis présente les messages à un rythme contrôlable au dispositif de filtrage. Cette méthode présente les caractéristiques inverses de la précédente : robustesse, mais nécessité de générer les éventuels avis de non remise.

Postfix dispose également d'un dispositif interne de filtrage assez sommaire : l'examen des en-têtes et du corps du message. Ce dernier présente les limites suivantes :

- Certains sont tentés d'écrire de nombreuses règles de filtrage en utilisant ces capacités : ça ne fera jamais de Postfix un logiciel anti-spam et la rapidité de traitement des messages risque d'en pâtir.
- L'examen du corps qui traite le message se fait ligne par ligne sans décodage : la recherche d'un mot ne peut être effectuée dans une pièce jointe, ni dans une partie de message encodée (en base 64 par exemple).

En revanche, ce dispositif peut faciliter l'intégration d'un dispositif comme nous le verrons par la suite. Ces généralités étant dites, passons à la technique :

## Insertion d'un dispositif en coupure

Dans une telle configuration, Postfix ne peut plus distinguer le courrier entrant du courrier sortant. Le contrôle anti-relais est alors délégué au dispositif amont (sauf s'il est compatible XFORWARD). La configuration de Postfix est alors extrêmement simple, il se contente d'assurer le routage. Si les deux logiciels se trouvent sur le même serveur, on change le port en écoute de Postfix (pour le laisser au dispositif de filtrage) en remplaçant dans le fichier ~~master.cf~~ la ligne :

```
smtp  inet  n  -  -  -  -  smtpd
```

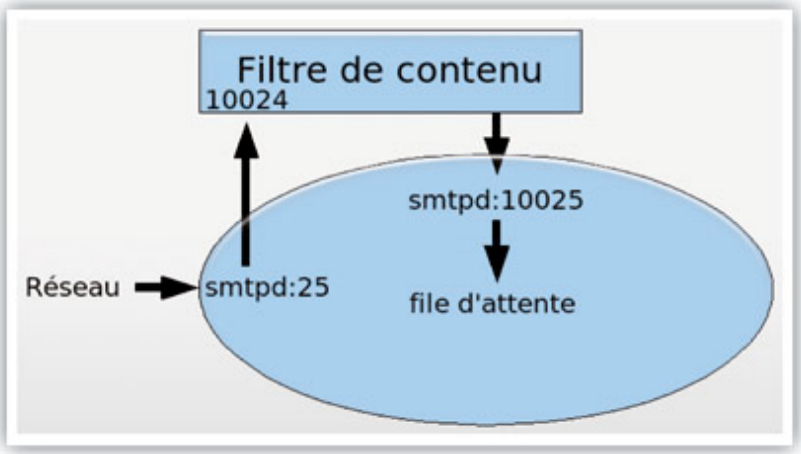
par la ligne :

```
127.0.0.1:11000  inet  n  -  -  -  -  smtpd
```

On met ainsi le serveur ~~smtpd~~ en écoute sur la boucle locale seulement sur le port 11000, port sur lequel le dispositif de filtrage doit renvoyer le courrier.

# Insertion d'un proxy SMTP

Le principe est ici d'insérer le dispositif entre deux démons ~~smtpd~~ de Postfix :




Le premier en écoute sur le port 25 transmet toutes les commandes au dispositif configuré pour renvoyer le courrier au deuxième serveur smtpd en écoute sur le port 10026. Le dispositif de filtrage doit être à même de comprendre toutes les commandes SMTP reçues par Postfix et doit ainsi être compatible ESMTP.
Ce qui donne dans le fichier ~~master.cf~~ :

```
smtp  inet  n  -  n  -  20  smtpd
  -o smtpd_proxy_filter=machine:10025
  -o smtpd_client_connection_count_limit=10
#
# Serveur SMTP après-filtrage. Reçoit le courrier du filtre de
# contenu sur le port 10026.
#
:10026  inet  n  -  n  -  -  smtpd
  -o smtpd_authorized_xforward_hosts=machine
  -o smtpd_client_restrictions=
  -o smtpd_helo_restrictions=
  -o smtpd_sender_restrictions=permit_mynetworks,reject
  -o smtpd_recipient_restrictions=permit_mynetworks,reject
  -o smtpd_data_restrictions=
  -o mynetworks=machine/32
  -o receive_override_options=no_header_body_checks
```

Explications :

- On configure le premier ~~smtpd~~ pour utiliser le proxy smtpd [2], et on limite sa capacité pour éviter un effondrement du dispositif de filtrage.
- On retire au deuxième ~~smtpd~~ toute charge de filtrage et de contrôle (déjà assuré par le premier). Il se contente de n'accepter du courrier qu'en provenance du filtre et on lui ordonne d'accepter les commandes XFORWARD.

# Insertion d'un filtre MILTER

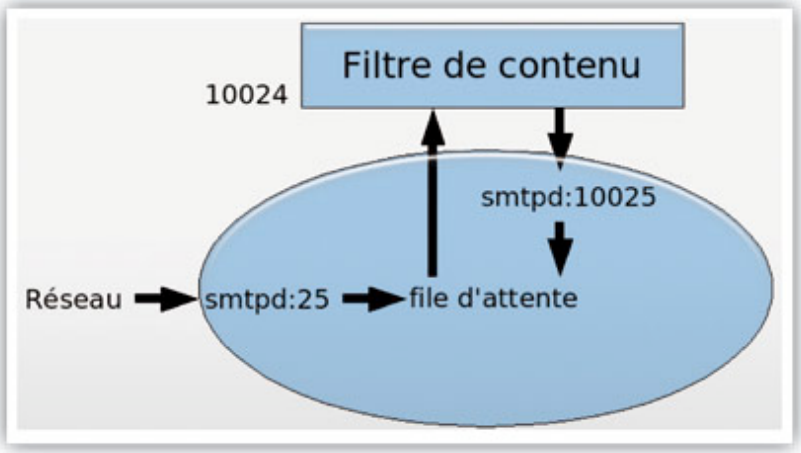
L'emploi du protocole MILTER [1] ne nécessite pas de deuxième démon ~~smtpd~~. Il suffit d'indiquer dans ~~main.cf~~ la liste des filtres MILTER à utiliser :

```
cleanup_milters = inet:localhost:11025, ...
```

Ce mécanisme n'est disponible qu'à partir de la version 2.3 de Postfix.

# Insertion d'un filtre après mise en file d'attente

Ce dispositif est assez similaire au mode proxy si ce n'est qu'on utilise la directive ~~content_filter~~ en lieu et place de la directive ~~smtpd_proxy_filter [2]~~. On peut également utiliser dans cette configuration un filtre qui reçoit et/ou soumet le courrier par d'autres protocoles.




Exemple : notre filtre reçoit les messages par un pipe Unix et les envoie en utilisant la commande ~~sendmail~~ de Postfix :

- Le fichier ~~master.cf~~ contient :

```
filtre  unix  -  n  n  -  10  pipe
  flags=Rq user=filter argv=/rep/script -f ${sender} -- ${recipient}
```

- ☐La directive de filtrage appelle notre filtre :

```
content_filter=filtre:rien.
```

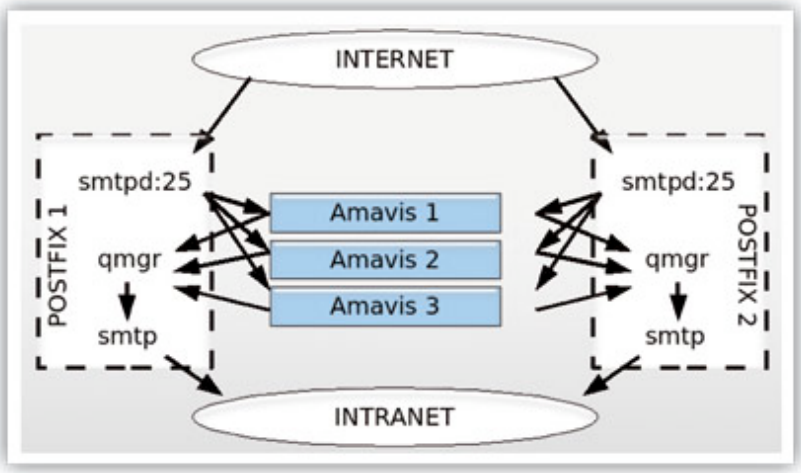
- ☐Le filtre lit le message sur l'entrée standard et appelle ~~sendmail~~ :

```
#!/bin/sh
```

```
# Nettoyage lors en sortant ou lors d'une interruption
trap «rm -f in.$$» 0 1 2 3 15
# Démarrage du processus.
cd /tmp
cat >in.$$
# filtrage à écrire sur le fichier in.$$
# Renvoi
sendmail -G -i «$@» <in.$$
# Renvoi du code de retour de sendmail
exit $?
```

# Robustesse

Quel que soit le mode choisi, un processus de filtrage défaillant peut rendre indisponible le service de messagerie : s'il fonctionne sur la même machine, il peut absorber toute la mémoire disponible ou tout simplement ne plus répondre.
On peut contrôler la charge du filtre en limitant le nombre de livraisons parallèles comme nous l'avons déjà vu, mais ça ne nous préserve pas d'un dysfonctionnement plus grave lié à un virus non supporté par exemple.
En utilisant le mécanisme de répartition de charge basé sur les MX présentés précédemment dans cette série d'articles, on peut créer un système rustique :




- Les serveurs Postfix fonctionnent sur des serveurs distincts des filtres de contenu.
- On insère dans le DNS une entrée MX par filtre de contenu :

```
mx-filtrage  IN  CNAME  filtre1
mx-filtrage  IN  CNAME  filtre2
```

- On crée un transporteur smtp particulier dans ~~master.cf~~ pour ce filtre pour contrôler la charge envoyée :

```
filtre  unix  -  -  -  -  -  smtp
  -o smtp_send_xforward_command=yes
```

et on limite son usage dans ~~main.cf~~ :

- On utilise ce champ dans la directive

```
content_filter (ou smtpd_proxy_filter) : content_filter = filtre:mx-filtrage:10024.
```

- Si le filtre est capable d'utiliser le même mécanisme pour renvoyer le courrier à la deuxième instance du serveur ~~smtpd~~, on crée les entrées MX correspondantes, sinon on fait la même chose avec les enregistrements de type A (round-robbin) :

```
routage  IN  A  postfix1
routage  IN  A  postfix2
```

Le filtre renvoie alors le courrier vers ~~routage:10025~~. L'inconvénient des enregistrements A est que si l'un des Postfix est injoignable, le filtre peut être redirigé lors des tentatives suivantes vers le même serveur. On risque ainsi de perdre du courrier, même si cette probabilité est assez faible. L'emploi d'Amavis [3] amène également une certaine rusticité, puisque ce logiciel sert d'interface entre Postfix, l'anti-spam et les anti-virus. On peut alors également utiliser des anti-virus de secours en cas de défaillance des premiers. Il ne protège toutefois pas des problèmes de saturation de la mémoire. Il s'utilise avec ~~smtpd_proxy_filter~~ ou ~~content_filter~~. Par défaut, il est en écoute sur le port 10024 et renvoie son courrier vers le port 10025.

## Autres mécanismes disponibles dans Postfix

### Délégation de la politique d'accès

Postfix dispose d'un autre mécanisme permettant d'insérer un filtre extérieur : la délégation de politique d'accès [5]. Il ne s'agit pas ici pour le filtre d'examiner le contenu du message, mais simplement, à partir des données de connexion (adresse IP, expéditeur, nom communiqué dans le HELO...), de retourner une action de type table d'accès [4] (ex : ~~action=reject~~). Ce mécanisme est utilisé par exemple pour utiliser les listes grises (voir plus bas). Son emploi est assez simple : on insère ~~check_policy_service inet:host:port~~ dans une restriction. Par exemple :

```
smtpd_recipient_restrictions = permit_mynetworks,
         reject_unauth_destination,
         check_policy_service inet:127.0.0.1:60000
```

### Listes noires en temps réel (RBL)

Postfix peut également intégrer la consultation de listes noires de sites mises à jour en temps réel. Il faut toutefois bien se renseigner sur les mécanismes utilisés pour mettre à jour ces listes avant d'en choisir une pour éviter de rejeter un grand FAI sous prétexte qu'un de ses clients fait l'objet d'une plainte. Comme pour la délégation de politique d'accès, leur emploi s'effectue au travers des restrictions. Exemple :

```
smtpd_recipient_restrictions = permit_mynetworks,
               reject_unauth_destination,
               reject_rbl_client relays.ordb.org,
```

La liste publiée par ORDB n'est pas très efficace, mais n'amène aucune erreur, car le site vérifie que la machine dénoncée est réellement un relais ouvert avant de la mettre en liste noire.

# Lutter contre le courrier indésirable avec Postfix

La lutte contre le courrier indésirable n'est pas un problème facile. Outre la complexité des

technologies utilisées, il est très difficile pour un administrateur d'un grand réseau d'appliquer une politique de filtrage uniforme et qui convienne à tous les utilisateurs. Dans certains cas, il est même inadmissible de prendre le risque de bloquer un message licite. A l'inverse, les utilisateurs acceptent parfois assez difficilement de recevoir 15 publicités par jour en anglais pour des produits pharmaceutiques. Si la technologie anti-spam n'est pas une réponse à tout, certains mécanismes basés sur des règles logiques peuvent contribuer à éliminer le courrier indésirable sans aucun doute.

# Interdiction des exécutables malveillants

Un anti-virus mis à jour en temps réel élimine à lui tout seul la menace que nous avons appelée « exécutables dangereux standards » à ceci près qu'il n'est pas toujours en mesure d'éliminer à temps le dernier sorti des virus.
L'élimination des exécutables malveillants spécifiques ou très récemment sortis ne peut passer que par une interdiction formelle de tout ce qui présente un danger pour le client. Le logiciel Amavis [3] est en mesure d'éliminer les pièces jointes par extension (.exe,...) ou par Content-Type. Cette mesure ne suffit pas à protéger les utilisateurs, car il existe d'autres moyens de lancer du code malveillant (certaines balises HTML, les Javascripts, les XPI...). Pour avoir la politique la plus fine possible, il faut connaître les logiciels clients utilisés et suivre les menaces pour adapter ce filtrage.

# Menace interne

Une mesure simple pour limiter les éventuelles nuisances internes consiste à exiger une authentification interne sur TLS (voir LM N° 85). Ça n'élimine pas tout, mais couplée à l'anti-virus et aux autres éléments de la politique de filtrage, la mesure est efficace.

# Utilisation des listes grises

Dans notre classification, nous avons distingué le courrier externe issu d'un robot ou déjà accepté par un serveur de messagerie. Le mécanisme le plus efficace pour lutter contre les robots consiste à utiliser les listes grises. Le principe en est assez simple. On rejette toute première requête d'un client avec un code d'erreur temporaire (4xx). Si le serveur présente de nouveau son message, on peut supposer que le client est un vrai serveur de messagerie.
Comme nous l'avons indiqué plus haut, le mécanisme le plus approprié dans ce cas est l'emploi de la délégation de politique d'accès.
Sous Debian, l'installation d'un dispositif de liste grise est très simple (comme d'habitude ;-)) [6] :

```
apt-get install postgrey
```

Le démon est alors installé et en écoute sur le port 60000. Il ne reste plus qu'à indiquer à Postfix de l'utiliser comme au paragraphe « délégation de la politique d'accès ».
Le problème avec ces listes est que certains sites supportent assez mal le rejet et ne représentent le message que beaucoup plus tard ou plus du tout. Il faut donc entretenir une liste blanche des serveurs susceptibles.

# Filtrage intégré

On peut faire beaucoup de contrôles avec Postfix en vérifiant la validité de tous les champs, en filtrant les en-têtes,... il faut toutefois bien expérimenter ces filtres pour mesurer leur impact. Le mot clef ~~warn_if_reject~~ [7], qu'on peut utiliser devant n'importe quelle restriction, permet d'enregistrer

le rejet dans les journaux sans qu'il soit effectif et peut alors vous rendre de grands services.

# Examen des commandes EHLO/HELO

En théorie, le nom d'hôte passé dans la commande HELO devrait correspondre au nom DNS du serveur, mais comme indiqué dans la RFC 2505 [8], il n'est pas conseillé d'examiner de manière trop stricte ce nom. La restriction suivante rejette ainsi la plupart des robots, mais malheureusement aussi quelques grands FAI mal administrés :

```
smtpd_helo_restrictions = permit_mynetworks,
                reject_invalid_hostname,
                reject_unknown_hostname
```

A utiliser donc avec précaution. Certains logiciels utilisés pour la délégation de politique d'accès tels ~~policyd~~ [9] proposent des listes noires de noms d'hôtes utilisés dans les commandes HELO.

## Rejet des fausses notifications

L'avis de non-remise contient généralement les en-têtes du message d'origine. Les robots, cherchant généralement à singer le site apparemment source, insèrent dans les en-têtes ou dans le nom communiqué avec la commande HELO des termes semblant indiquer la provenance de notre site. L'astuce proposée ici consiste à rejeter les messages contenant des en-têtes manifestement falsifiés. En supposant que nos serveurs n'utilisent que des noms de type ~~mail#.domaine.com~~, un en-tête faisant référence à une commande ~~HELO domaine.com~~ doit être rejeté. De même, les en-têtes ~~Message-Id~~ générés par nos serveurs ne peuvent contenir ~~@domain.com~~, mais se terminent par ~~@mail#.domaine.com~~ [10] :

```
 # /etc/postfix/main.cf
header_checks = pcre:/etc/postfix/entetes
body_checks
# /etc/postfix/entetes
/^Received: +from +(domaine\.com) +/
  reject forged client name in Received: header: $1
/^Received: +from +[^ ]+ +\(([^ ]+ +[he]+lo=|[he]+lo +)(domaine\.com)\)/
  reject forged client name in Received: header: $2
/^Message-ID:.*@(domaine\.com)/
  reject forged domain name in Message-ID: header: $1
/etc/postfix/body_checks:
/^[> ]*Received: +from +(domaine\.com) /
  reject forged client name in Received: header: $1
/^[> ]*Received: +from +[^ ]+ +\(([^ ]+ +[he]+lo=|[he]+lo +)(domaine\.com)\)/
  reject forged client name in Received: header: $2
/^[> ]*Message-ID:.*@(domaine\.com)/
  reject forged domain name in Message-ID: header: $1
```

## Rejet des adresses de destination aléatoires

Le problème des adresses de destination aléatoires est que les messages peuvent être acceptés par le serveur SMTP et c'est seulement à la livraison finale que le système découvre que le destinataire n'existe pas. Le système génère alors un avis de non-remise qui va générer de nouvelles fausses notifications vers l'extérieur. Pour ne pas cotiser à ce désordre, il est important de rejeter ces messages dès la réception de la commande ~~RCPT TO~~.
Par défaut, Postfix renseigne la directive ~~local_recipient_maps~~ pour n'accepter que les utilisateurs disposant d'un compte Unix ou renseignés dans les alias. En cas d'utilisation d'un autre dispositif

de gestion des boîtes aux lettres, le simple fait de renseigner ce paramètre (par une table LDAP par exemple) sur le serveur recevant le courrier extérieur élimine le problème des adresses aléatoires. Pour les domaines relayés, Postfix dispose d'un mécanisme équivalent, mais évidemment non renseigné par défaut : ~~relay_recipient_maps~~.
Les tables passées en paramètre à ces deux directives peuvent renvoyer n'importe quelle valeur. Postfix vérifie simplement que le destinataire existe. On peut ainsi utiliser des tables conçues pour le routage par exemple.

### Traitement du spam résiduel

Une fois toutes ces mesures en place, il nous reste le plus difficile à éliminer : le spam bien conçu et déjà accepté par une passerelle de messagerie externe. A ce niveau, ce n'est plus l'affaire de Postfix, mais bien d'un logiciel spécialisé. Amavis [3], nous permet une fois de plus de connecter très simplement le très célèbre Spamassassin : tout est prêt dans les fichiers de configuration d'Amavis, il n'y a plus qu'à décommenter la ligne.
Nous ne rentrerons pas ici dans la configuration de Spamassassin qui pourrait, à elle seule, constituer une série d'articles. Signalons juste quelques-uns des mécanismes utilisés par Spamassassin :

- Filtres bayesiens et heuristiques : ils donnent des résultats approximativement sûrs, mais peuvent rejeter du courrier licite si la barre de notation est fixée trop bas. Ces filtres sont en revanche très efficaces sur le poste client (comme le gestionnaire des indésirables de Thunderbird), car la politique est individualisée.
- Listes type razor : consultation d'une base de données pour voir si le message est déjà connu dans la base. Ces listes donnent de très bons résultats, sauf au tout début de la propagation d'un spam.

# Conclusion

La lutte contre le courrier indésirable est trop complexe pour être confiée à un seul mécanisme dans un grand réseau (à moins qu'on en accepte les risques). J'espère vous avoir apporté dans cet article quelques éléments d'éclairage sur les possibilités intrinsèques de Postfix et sa facilité à accueillir d'autres dispositifs. J'utilise personnellement ce mécanisme pour insérer de petits programmes Perl effectuant divers traitements sur les messages, comme des extractions vers des bases de données ou encore des contrôles de cohérence entre les adresses d'enveloppe et celles renseignées dans les en-têtes. Peut-être dans un prochain article ?

**Notes** :

- [1] Quelques explications sur le protocole MILTER et son intégration dans Postfix : http://postfix.traduc.org/index.php/MILTER_README.html
- [2] Les directives ~~content_filter~~ et ~~smtpd_proxy_filter~~ peuvent également être insérées dans le fichier ~~main.cf~~ et non comme paramètre du premier ~~smtpd~~. Dans ce cas, elles devront être redéfinies dans le deuxième ~~smtpd~~ pour éviter un bouclage du courrier en ajoutant les options :
  ~~-o smtpd_proxy_filter=~~
  ~~-o content_filter=~~
- [3] Amavis est disponible sur : http://www.amavis.org/
- [4] Les actions possibles dans une table d'accès sont indiquées dans la page de manuel

access(5) ou sur la page : http://www.postfix.org/access.5.html
- [5] http://postfix.traduc.org/index.php/SMTPD_POLICY_README.html
- [6] Sur Debian Sarge, il faut utiliser les portages du site http://www.backports.org pour obtenir postgrey.
- [7] http://postfix.traduc.org/index.php/postconf.5.html#warn_if_reject
- [8] La RFC 2505 contient des recommandations pour lutter contre le spam au niveau des MTA :
http://rfc.net/rfc2505.html
- [9] Policyd implémente à la fois les listes grises et d'autres mécanismes de lutte contre le spam :
http://policyd.sourceforge.net/
- [10] Par défaut, ~~header_checks~~ s'applique aux en-têtes du message ainsi qu'aux en-têtes des messages en pièces jointes. Ce mécanisme est mis en place par les valeurs par défaut suivantes :
~~nested_header_checks = $header_checks~~
~~mime_header_checks = $header_checks~~

☐Si vous souhaitez accueillir le courrier de personnes transférant de faux messages, il faut indiquer à Postfix que ~~header_checks~~ ne doit pas être utilisé en dehors des en-têtes du message en renseignant ces deux paramètres par d'autres tables ou rien si vous ne souhaitez aucun filtrage sur les en-têtes des pièces jointes.

~~Retrouvez cet article dans : Linux Magazine 89~~

Posté par (La rédaction) | Signature : Xavier Guimard | Article paru dans 

### Laissez une réponse

Vous devez avoir ouvert une session pour écrire un commentaire.

« Précédent Aller au contenu »

Identifiez-vous
Inscription
S'abonner à UNIX Garden

## Articles de 1ère page

- Lutter contre le spam avec Postfix
- Yafray, le moteur de rendu photoréaliste libre, une première approche
- Ajouter des logiciels : la gestion des paquets
- Web : récupérez vos marque-pages
- Une nouvelle disposition de clavier français pour Xorg
- Installation et configuration d'E17

- Migration des données
- Ubuntu, un peu d'histoire...
- H.P.Anvin : M. ISOLinux / SYSLinux
- Les marques déposées et les Logiciels libres


## Il y a actuellement

639 articles/billets en ligne.

Recherche

## Catégories

- Administration réseau
- Administration système
- Agenda-Interview
- Audio-vidéo
- Bureautique
- Comprendre
- Distribution
- Embarqué
- Environnement de bureau
- Graphisme
- Jeux
- Matériel
- News
- Programmation
- Réfléchir
- Sécurité

- Utilitaires
- Web

## Archives

- juillet 2008
- juin 2008
- mai 2008
- avril 2008
- mars 2008
- février 2008
- janvier 2008
- décembre 2007
- novembre 2007
- février 2007

## GNU/Linux Magazine

- GNU/Linux Magazine 107 - Juillet/Août 2008 - Chez votre marchand de journaux
- Edito : GNU/Linux Magazine 107
- GNU/Linux Magazine HS 37 - Juillet/Août 2008 - Chez votre marchand de journaux
- Edito : GNU/Linux Magazine HS 37
- GNU/Linux Magazine 106 - Juin 2008 - Chez votre marchand de journaux !

## GNU/Linux Pratique

- Linux Pratique N°48 -Juillet/Août 2008 - Chez votre marchand de journaux
- Edito : Linux Pratique N°48
- Linux Pratique Essentiel N°2 - Juin/Juillet 2008 - Chez votre marchand de journaux
- Edito : Linux Pratique Essentiel N°2
- Linux Pratique Hors-Série N°15 - Juin / Juillet 2008 - chez votre marchand de journaux.

## MISC Magazine

- Références de l'article « Détection de malware par analyse système » d'Arnaud Pilon paru dans MISC 38
- Références de l'article « La sécurité des communications vocales (3) : techniques numériques » d'Éric Filiol paru dans MISC 38
- Misc 38 : Codes Malicieux, quoi de neuf ? - Juillet/Août 2008 - Chez votre marchand de journaux
- Edito : Misc 38
- Misc 37 : Déni de service - Mai/Juin 2008 - Chez votre marchand de journaux

© 2007 - 2008 UNIX Garden. Tous droits réservés .