

Documentation de Postfix en français

Table of Contents

<u>Documentation de Postfix en français</u>	1
<u>Configuration de Postfix – Éléments de base</u>	3
<u>Introduction</u>	3
<u>Fichiers de configuration de Postfix</u>	4
<u>Quel domaine afficher dans le courrier sortant</u>	4
<u>De quels domaines recevoir le courrier</u>	5
<u>De quels clients relayer le courrier</u>	5
<u>De quelles destination relayer le courrier</u>	6
<u>Quelle méthode de livraison : directe ou indirecte</u>	6
<u>Quels incidents rapporter au postmaster</u>	7
<u>Adresse réseau proxy/NAT</u>	8
<u>Ce que vous devez savoir sur les logs de Postfix</u>	8
<u>Lancer Postfix en environnement "chroot"</u>	9
<u>Mon nom de machine</u>	10
<u>Mon nom de domaine</u>	10
<u>Mes adresses réseau</u>	10
<u>Exemples de configuration standard de Postfix</u>	12
<u>Objet de ce document</u>	12
<u>Postfix sur une machine Internet autonome</u>	12
<u>Postfix sur un client sans rôle</u>	13
<u>Postfix sur un réseau local</u>	13
<u>Postfix sur une passerelle/pare-feu</u>	15
<u>Livrer une partie seulement des courriers localement</u>	16
<u>Utiliser Postfix derrière un pare-feu</u>	17
<u>Configurer Postfix comme serveur MX pour un site distant</u>	18
<u>Utiliser Postfix avec une connexion par modem</u>	19
<u>Postfix sur des machines sans nom réel</u>	20
<u>Solution 1: Postfix version 2.2 et supérieures</u>	20
<u>Solution 2 : Postfix version 2.1 et antérieures</u>	21
<u>Traductions d'adresses</u>	22
<u>Introduction au système de traduction des adresses de Postfix</u>	22
<u>Réécrire ou non, ou labelliser comme invalide</u>	23
<u>Introduction à la traduction d'adresse Postfix</u>	23
<u>Réécriture des adresses à la réception</u>	24
<u>Réécriture à la forme standard</u>	25
<u>Correspondances canoniques</u>	26
<u>Masquage d'adresses</u>	28
<u>Ajout automatique d'un destinataire caché (BCC)</u>	29
<u>Alias virtuels</u>	30
<u>Réécriture des adresses à l'émission</u>	31
<u>Résolution d'adresse de destination</u>	31
<u>Routage du courrier</u>	32
<u>Table des utilisateurs déplacés</u>	32
<u>Remplacement générique pour le courrier SMTP sortant</u>	32
<u>Base de données locale d'alias</u>	33

Table of Contents

Traductions d'adresses

<u>Local per-utilisateur fichiers .forward</u>	34
<u>Récupération locale de toutes les adresses</u>	34
<u>Deboguer vos manipulations d'adresses</u>	34

Hébergement de sites virtuels avec Postfix.....37

<u>Objectifs de ce document</u>	37
<u>Hébergement canonique et hébergement d'autres domaines</u>	37
<u>Fichiers locaux et bases de données en réseau</u>	38
<u>Aussi simple que possible : domaines partagés, comptes UNIX</u>	38
<u>Exemple d'ALIAS virtuel Postfix : domaines séparés, comptes du système UNIX</u>	38
<u>Exemple de BOITES-AUX-LETTRES virtuelle : domaines séparés, comptes non-UNIX</u>	39
<u>Stockage en boîte-aux-lettres non-Postfix : domaines séparés, comptes non-UNIX</u>	41
<u>Domaines de transfert de courrier</u>	43
<u>Listes de diffusion</u>	44
<u>Répondeur automatique</u>	44

Authentification SASL avec Postfix.....46

<u>ATTENTION ATTENTION ATTENTION</u>	46
<u>Comment Postfix utilise les informations d'authentification SASL</u>	46
<u>Quelles implémentations SASL sont supportées</u>	46
<u>Building Postfix with Dovecot SASL support</u>	47
<u>Compiler les bibliothèques Cyrus-SASL</u>	47
<u>Compiler Postfix avec le support de Cyrus-SASL</u>	48
<u>Activer l'authentification SASL dans le serveur SMTP de Postfix</u>	48
<u>Configuration de Dovecot SASL pour le serveur SMTP de Postfix</u>	49
<u>Configuration de Cyrus SASL pour le serveur SMTP de Postfix</u>	49
<u>Tester l'authentification SASL dans le serveur SMTP de Postfix</u>	51
<u>Dépanner SASL</u>	52
<u>Activer l'authentification SASL dans le client SMTP de Postfix</u>	52
<u>Références</u>	53

Support d'IPv6 dans Postfix.....55

<u>Introduction</u>	55
<u>Plateformes supportées</u>	55
<u>Configuration</u>	56
<u>Limitations connues</u>	57
<u>Compatibilité avec le support IPv6 de Postfix <2.2</u>	57
<u>IPv6 pour les plateformes non supportées</u>	58
<u>Références</u>	59

Support TLS de Postfix.....60

<u>ATTENTION</u>	60
<u>Ce que le support TLS de Postfix fait pour vous</u>	60
<u>Comment fonctionne le support TLS de Postfix</u>	60
<u>Compiler Postfix avec le support de TLS</u>	61
<u>Paramètres spécifiques au serveur SMTP</u>	62
<u>Configuration coté serveur du certificat et de la clef privée</u>	62

Table of Contents

Support TLS de Postfix

<u>Enregistrement de l'activité TLS coté serveur</u>	64
<u>Activer TLS dans le serveur SMTP de Postfix</u>	64
<u>Vérification du certificat client</u>	65
<u>Supporter l'authentification sur TLS seulement</u>	66
<u>Cache des sessions TLS coté serveur</u>	66
<u>Contrôle d'accès au serveur</u>	66
<u>Contrôle du chiffrement coté serveur</u>	67
<u>Contrôles divers du serveur</u>	68
<u>Paramètres spécifiques au client SMTP</u>	68
<u>Configuration coté client du certificat et de la clef privée</u>	68
<u>Enregistrement de l'activité TLS coté client</u>	70
<u>Cache des sessions TLS coté client</u>	70
<u>Activer TLS dans le client SMTP de Postfix</u>	71
<u>Exiger le chiffrement TLS</u>	71
<u>Politique TLS par site</u>	72
<u>Fermer un trou de sécurité lié au DNS avec des politiques TLS par site</u>	73
<u>Découvrir les serveurs qui proposent TLS</u>	74
<u>Vérification du certificat du serveur</u>	74
<u>Contrôle du chiffrement coté client</u>	75
<u>Contrôles divers du client</u>	75
<u>Paramètres spécifiques au gestionnaire TLS</u>	75
<u>Documentation rapide</u>	76
<u>Rapporter les problèmes</u>	78
<u>Compatibilité avec le support TLS de Postfix < 2.2</u>	78
<u>Références</u>	79

Installation de Postfix depuis le code source.....80

<u>1 – but de ce document</u>	80
<u>2 – Conventions typographiques</u>	80
<u>3 – Documentation</u>	81
<u>4 – Compiler Postfix sur un système supporté</u>	81
<u>4.1 – Démarrer la compilation</u>	82
<u>4.2 – Quel compilateur utiliser</u>	82
<u>4.3 – Compiler avec des extensions optionnelles</u>	82
<u>4.4 – Surcharger les valeurs des paramètres par défaut</u>	83
<u>4.5 – Support de milliers de processus</u>	83
<u>4.6 – Finalement, compiler Postfix</u>	84
<u>5 – Porter Postfix sur un système non supporté</u>	84
<u>6 – Installer le logiciel après une compilation réussie</u>	84
<u>6.1 – Sauver les binaires Sendmail existants</u>	84
<u>6.2 – Create account and groups</u>	85
<u>6.3 – Installer Postfix</u>	85
<u>6.4 – Configurer Postfix</u>	85
<u>7 – Configurer Postfix seulement pour envoyer du courrier</u>	86
<u>8 – Configurer Postfix pour envoyer et recevoir du courrier par une interface virtuelle</u>	86
<u>9 – Utiliser Postfix au lieu de Sendmail</u>	87
<u>10 – Édition obligatoire des fichiers de configuration</u>	88

Table of Contents

Installation de Postfix depuis le code source

<u>10.1 – Fichiers de configuration de Postfix</u>	88
<u>10.2 – Domaine par défaut pour les adresses non qualifiées</u>	89
<u>10.3 – De quels domaines recevoir le courrier localement</u>	89
<u>10.4 – Adresses des interfaces des proxies/traducteurs d'adresses</u>	89
<u>10.5 – De quels clients locaux relayer le courrier</u>	89
<u>10.6 – Quelles destinations relayées accepter pour les clients étrangers</u>	90
<u>10.7 – Optionel: configurer une machine pour la livraison extérieure</u>	90
<u>10.8 – Créez la base de données d'alias</u>	90
<u>11 – Mettre en cage chroot ou non</u>	91
<u>12 – Soins et alimentation du système Postfix</u>	91

Analyse des goulots d'étranglement.....93

<u>But de ce document</u>	93
<u>Introduction de l'outil qshape</u>	93
<u>Problèmes avec qshape</u>	94
<u>Exemple 1: file d'attente saine</u>	95
<u>Exemple 2: File d'attente retardée pleine de rejet de messages forgés</u>	95
<u>Exemple 3: Congestion de la file d'attente active</u>	96
<u>Exemple 4: Grand volume de messages retardés</u>	97
<u>Information d'arrière-plan : répertoires de files d'attente de Postfix</u>	98
<u>La file d'attente "maildrop"</u>	98
<u>La file d'attente "hold"</u>	99
<u>La file d'attente "entrante" (incoming)</u>	100
<u>La file d'attente "active"</u>	101
<u>La file d'attente "retardée" (deferred)</u>	102
<u>Références</u>	103

Optimisation des performances de Postfix.....104

<u>But de l'optimisation des performances de Postfix</u>	104
<u>Éléments généraux de performance de la réception</u>	105
<u>Faire plus de travail avec vos processus serveurs SMTP</u>	105
<u>Ralentir les clients SMTP qui commettent beaucoup d'erreurs</u>	105
<u>Mesures contre les clients qui ouvrent trop de connexions</u>	106
<u>Éléments généraux de performance de la livraison</u>	107
<u>Optimiser le nombre de livraisons simultanées</u>	107
<u>Optimiser le nombre de destinataires par livraison</u>	108
<u>Optimiser la fréquence de tentatives de livraison du courrier retardé</u>	109
<u>Optimiser le nombre de processus Postfix</u>	110
<u>Optimiser le nombre de fichiers ou de sockets ouverts</u>	110

Howto déboguage Postfix.....112

<u>But de ce document</u>	112
<u>Recherche des signes manifestes de problème</u>	112
<u>Déboguer Postfix de l'intérieur</u>	113
<u>Désactiver la mise en cage dans master.cf</u>	113
<u>Journaux verbeux pour des connexions SMTP spécifiques</u>	114
<u>Enregistrer une session SMTP avec un sniffer réseau</u>	114

Table of Contents

Howto déboguage Postfix

<u>Rendre les programmes démons de Postfix plus bavards</u>	114
<u>Tracer manuellement un processus démon de Postfix</u>	114
<u>Tracer automatiquement un processus démon de Postfix</u>	115
<u>Lancer des programmes démons avec le debugger interactif <code>xxgdb</code></u>	115
<u>Lancer des programmes démons avec un debugger non-interactif</u>	116
<u>Comportement déraisonnable</u>	116
<u>Rapporter les problèmes à la liste <code>postfix-users@postfix.org</code></u>	117

Inspection du contenu par Postfix.....119

Arrêt des notifications indésirables.....120

<u>Introduction</u>	120
<u>Que sont les notifications indésirables ?</u>	120
<u>Comment bloquer les notifications à destination d'adresses aléatoires ?</u>	120
<u>Comment bloquer les notifications à destination d'adresses réelles ?</u>	120
<u>Bloquer les notifications avec un HELO renseigné</u>	121
<u>Bloquer les notifications contenant un expéditeur</u>	122
<u>Bloquer les notifications avec d'autres informations</u>	122
<u>Bloquer les notifications des antivirus</u>	123

Inspection du contenu intégrée à Postfix.....124

<u>Introduction à l'inspection du contenu intégrée à Postfix</u>	124
<u>Quel courrier est assujéti à l'examen des en-têtes/du corps</u>	125
<u>Limites de l'examen des en-têtes/du corps</u>	125
<u>Prévenir le blocage du rapport quotidien de livraison</u>	126
<u>Configurer l'examen des en-têtes/du corps seulement pour le courrier venant de l'extérieur</u>	127
<u>Configurer l'examen des en-têtes/du corps seulement pour le courrier de certains domaines</u>	127

Filtrage du contenu après mise en file d'attente.....129

<u>Introduction</u>	129
<u>Principes d'opération</u>	130
<u>Exemple de filtrage simple de contenu</u>	130
<u>Performances du filtrage simple de contenu</u>	132
<u>Limites du filtrage simple de contenu</u>	132
<u>Désactiver le filtrage simple de contenu</u>	133
<u>Exemple de filtrage avancé de contenu</u>	133
<u>Filtrage avancé de contenu : demander que tout le courrier soit filtré</u>	133
<u>Filtrage avancé de contenu : envoyer les messages non filtrés au filtre de contenu</u>	134
<u>Filtrage avancé de contenu : lancer le filtre de contenu</u>	134
<u>Filtrage avancé de contenu : ré-injecter les messages dans Postfix</u>	135
<u>Performance du filtrage avancé de contenu</u>	135
<u>Désactiver le filtrage avancé de contenu</u>	136
<u>Filtrer le courrier des utilisateurs extérieurs seulement</u>	136
<u>Filtres différents par domaine</u>	137
<u>Actions FILTER dans les tables d'accès ou d'en-tête/contenu</u>	137

Table of Contents

<u>Filtrage de contenu avant mise en file d'attente avec Postfix</u>	139
<u>ATTENTION ATTENTION ATTENTION</u>	139
<u>La fonctionnalité de filtrage de contenu avant mise en file d'attente de Postfix</u>	139
<u>Principes de l'opération</u>	139
<u>Pour et contre le filtrage de contenu avant mise en file d'attente</u>	140
<u>Configurer le dispositif proxy SMTP de Postfix</u>	140
<u>Paramètres de configuration</u>	142
<u>Comment Postfix dialogue avec le filtre avant mise en file d'attente</u>	142
<u>Support Militer de Postfix avant mise en file d'attente</u>	143
<u>Introduction</u>	143
<u>Compiler les applications Militer</u>	143
<u>Lancer les applications Militer</u>	144
<u>Configurer Postfix</u>	144
<u>Applications Militer</u>	144
<u>Interprétation des erreurs Militer</u>	145
<u>Version du protocole Militer</u>	145
<u>Timeouts du protocole Militer</u>	146
<u>Macro émulation Sendmail</u>	146
<u>Contournement des problèmes</u>	147
<u>Limites</u>	148
<u>Contrôle d'accès et de relais SMTP avec Postfix</u>	150
<u>Introduction</u>	150
<u>Contrôle de relais, de pourriel et politique par utilisateur</u>	150
<u>Restrictions à appliquer à tous les messages SMTP</u>	151
<u>Sélectivité avec les listes de restriction d'accès</u>	151
<u>Evaluation différée des listes de restriction d'accès SMTP</u>	152
<u>Utilisation dangereuse du paramètre smtpd_recipient_restrictions</u>	153
<u>Tester les règles d'accès SMTP</u>	154
<u>Délégation de la politique d'accès SMTP avec Postfix</u>	155
<u>But de la délégation de la politique d'accès SMTP</u>	155
<u>Description du protocole</u>	155
<u>Configuration de la politique client/serveur</u>	157
<u>Exemple: serveur de politique liste grise</u>	158
<u>Mettre en liste grise le courrier des domaines fréquemment forgés</u>	159
<u>Mettre en liste grise tout le courrier</u>	160
<u>Routine de maintenance des listes grises</u>	160
<u>Exemple de serveur Perl de liste grise</u>	161
<u>Vérification des adresses par Postfix</u>	162
<u>ATTENTION ATTENTION ATTENTION</u>	162
<u>Quelles vérifications d'adresse Postfix peut-il faire pour vous</u>	162
<u>Comment marche la vérification d'adresse</u>	162
<u>Limites de la vérification d'adresse</u>	163
<u>Vérification des adresses de destination</u>	164
<u>Vérification des adresses d'expédition pour les messages de domaines fréquemment forgés</u>	164

Table of Contents

<u>Vérification des adresses par Postfix</u>	
<u>Vérification des adresses d'expédition pour tous le courrier</u>	165
<u>Base de données de vérification d'adresse</u>	165
<u>Gestion des base de données de vérification d'adresse</u>	166
<u>Contrôle du routage des sondages de vérification d'adresse</u>	166
<u>Exemples de routage forcé du sondage</u>	167
<u>Limites du routage forcé du sondage</u>	167
<u>Contrôle d'accès par client, utilisateur, etc.</u>	168
<u>Classes de restriction de Postfix</u>	168
<u>Protéger les listes de distribution internes</u>	169
<u>Restreindre les utilisateurs pouvant envoyer du courrier vers sites extérieurs</u>	170
<u>Postfix ETRN Howto</u>	171
<u>Purpose of the Postfix fast ETRN service</u>	171
<u>Using the Postfix fast ETRN service</u>	171
<u>How Postfix fast ETRN works</u>	172
<u>Postfix fast ETRN service limitations</u>	172
<u>Configuring the Postfix fast ETRN service</u>	172
<u>Configuring a domain for ETRN service only</u>	173
<u>Testing the Postfix fast ETRN service</u>	174
<u>Postfix and UUCP</u>	176
<u>Using UUCP over TCP</u>	176
<u>Setting up a Postfix Internet to UUCP gateway</u>	176
<u>Setting up a Postfix LAN to UUCP gateway</u>	177
<u>Introduction aux tables de correspondances</u>	178
<u>Introduction</u>	178
<u>Le modèle de table de correspondance de Postfix</u>	178
<u>Listes et tables de Postfix</u>	179
<u>Préparer Postfix pour les consultations LDAP ou SQL</u>	179
<u>Maintenir les fichiers tables de correspondances de Postfix</u>	179
<u>Mettre à jour les fichiers BD Berkeley en sûreté</u>	180
<u>Types de table de correspondances de Postfix</u>	181
<u>Howto CDB de Postfix</u>	183
<u>Introduction</u>	183
<u>Compiler Postfix avec CDB</u>	183
<u>Howto bases de données Berkeley</u>	185
<u>Introduction</u>	185
<u>Comment compiler Postfix sur des systèmes sans librairie Berkeley DB</u>	185
<u>Compiler Postfix sur les systèmes BSD avec de multiples versions de Berkeley DB</u>	186
<u>Compiler Postfix sur les systèmes Linux avec de multiples versions de Berkeley DB</u>	186
<u>Optimiser les performances</u>	187
<u>Problèmes en cas d'absence de la librairie pthread</u>	187

Table of Contents

<u>Postfix LDAP Howto</u>	188
<u>Support de LDAP dans Postfix</u>	188
<u>Compiler Postfix avec le support LDAP</u>	188
<u>Configurer les correspondances par consultation LDAP</u>	189
<u>Exemple: alias locaux</u>	189
<u>Exemple: domaines/adresses virtuelles</u>	190
<u>Autres utilisations des consultations LDAP</u>	190
<u>Notes et éléments à prendre en compte</u>	190
<u>Retours d'expérience</u>	191
<u>Références</u>	191
<u>Howto MySQL Postfix</u>	193
<u>Introduction</u>	193
<u>Compiler Postfix avec le support de MySQL</u>	193
<u>Utiliser des tables MySQL</u>	193
<u>Exemple: alias locaux</u>	194
<u>Notes complémentaires</u>	194
<u>Références</u>	194
<u>Support PCRE de Postfix</u>	195
<u>Support des expressions rationnelles PCRE (Perl Compatible Regular Expressions)</u>	195
<u>Compiler Postfix avec le support PCRE</u>	195
<u>Éléments à connaître</u>	195
<u>Howto PostgreSQL avec Postfix</u>	197
<u>Introduction</u>	197
<u>Compiler Postfix avec le support PostgreSQL</u>	197
<u>Configurer les tables de correspondances PostgreSQL</u>	197
<u>Exemple: local aliases</u>	197
<u>Utiliser des bases de données miroir</u>	198
<u>Credits</u>	198
<u>Postfix VERP Howto</u>	199
<u>Postfix VERP support</u>	199
<u>Paramètres de configuration VERP de Postfix</u>	199
<u>Utiliser VERP avec les gestionnaires de liste Majordomo etc.</u>	200
<u>Le support VERP dans le serveur SMTP de Postfix</u>	201
<u>Le support VERP dans la commande sendmail de Postfix</u>	201
<u>Le support VERP dans le serveur OMOP de Postfix</u>	202
<u>Postfix et Linux</u>	203
<u>Éléments de configuration des bases Berkeley DB</u>	203
<u>Éléments de configuration de Procmail</u>	203
<u>Performances de Syslogd</u>	203
<u>Postfix et NFS</u>	204

Table of Contents

<u>Postfix and Ultrix.....</u>	205
<u>Postfix on Ultrix.....</u>	205
<u>Postfix + Maildrop Howto.....</u>	206
<u>Introduction.....</u>	206
<u>Livraison directe sans utiliser l'agent local de livraison.....</u>	206
<u>Livraison indirecte via l'agent local de livraison.....</u>	207
<u>Références.....</u>	208
<u>Présentation de l'architecture de Postfix.....</u>	209
<u>Introduction.....</u>	209
<u>Comment Postfix reçoit le courrier.....</u>	209
<u>Comment Postfix livre le courrier.....</u>	210
<u>Ce qui se passe en coulisse.....</u>	211
<u>Commandes de Postfix.....</u>	214
<u>Rejeter les destinataires locaux inconnus avec Postfix.....</u>	216
<u>Introduction.....</u>	216
<u>Configurer local_recipient_maps dans main.cf.....</u>	216
<u>Quand devez-vous changer la paramètre local_recipient_maps dans main.cf.....</u>	217
<u>Format de la table des destinataires locaux.....</u>	217
<u>Classes d'adresses Postfix.....</u>	219
<u>Introduction.....</u>	219
<u>A quoi servent les classes d'adresses ?.....</u>	219
<u>Quelles classes d'adresses Postfix implémente-t-il ?.....</u>	219
<u>Améliorations par rapport à Postfix 1.1.....</u>	221
<u>Incompatibilités avec Postfix 1.1.....</u>	221
<u>Postfix Connection Cache.....</u>	223
<u>Introduction.....</u>	223
<u>What SMTP connection caching can do for you.....</u>	223
<u>Connection cache implementation.....</u>	223
<u>Connection cache configuration.....</u>	224
<u>Connection cache safety mechanisms.....</u>	225
<u>Connection cache limitations.....</u>	225
<u>Connection cache statistics.....</u>	225
<u>Support DSN de Postfix.....</u>	227
<u>Introduction.....</u>	227
<u>Restreindre le champ des notifications de "succès".....</u>	227
<u>Interface de la ligne de commande sendmail de Postfix.....</u>	228
<u>Compatibilité avec le support VERP de Postfix.....</u>	228
<u>Guidelines for Package Builders.....</u>	229
<u>Purpose of this document.....</u>	229
<u>General distributions: please provide a small default main.cf file.....</u>	229
<u>General distributions: please include README or HTML files.....</u>	229

Table of Contents

<u>Guidelines for Package Builders</u>	
<u>Postfix Installation parameters</u>	229
<u>Preparing a pre-built package for distribution to other systems</u>	229
<u>Begin Security Alert</u>	230
<u>End Security Alert</u>	230
<u>Installing a pre-built Postfix package</u>	230
<u>Ordonnancement de la file d'attente</u>	231
<u>Objectif de ce document</u>	231
<u>Pourquoi avoir remplacé l'ancien gestionnaire des files d'attente</u>	231
<u>Comment fonctionne le gestionnaire des files d'attente</u>	231
<u>Howto XCLIENT</u>	233
<u>But de l'extension SMTP XCLIENT</u>	233
<u>XCLIENT Command syntax</u>	233
<u>Exemples XCLIENT</u>	234
<u>Sécurité</u>	235
<u>SMTP connection caching</u>	235
<u>Howto XFORWARD</u>	236
<u>But de l'extension SMTP XFORWARD</u>	236
<u>Syntaxe de la commande XFORWARD</u>	236
<u>XFORWARD Exemple</u>	237
<u>Sécurité</u>	238
<u>Cache des connexions SMTP</u>	238
<u>Paramètres de Configuration de Postfix</u>	239
<u>Format du fichier main.cf de Postfix</u>	239
<u>Pages de manuel de Postfix</u>	337
<u>Informations pour les nouveaux utilisateurs de Postfix</u>	337
<u>Organisation des pages de manuel de Postfix</u>	337
<u>Commandes</u>	337
<u>Configuration de Postfix</u>	338
<u>Construction des tables</u>	338
<u>Types de tables</u>	338
<u>Processus démons</u>	338

Documentation de Postfix en français

Ce site propose une traduction de la [documentation de Postfix](#). N'hésitez pas à me signaler les erreurs ou meilleures traductions que vous trouverez (x.guimard@free.fr)

Configuration générale

- [Configuration de base](#)
- [Exemples de configurations standards](#)
- [Réécriture d'adresse](#)
- [Hébergement de sites virtuels](#)
- [Authentification SASL](#)
- [Support d'IPv6](#)
- [Authentification et chiffrement TLS](#)
- [Installation depuis le code source](#)

Résolution des problèmes

- [Analyse des goulots d'étranglement](#)
- [Optimisation des performance](#)
- [Stratégies de débogage](#)
- Messages d'erreur (*)

Inspection du contenu

- [Introduction à l'inspection du contenu](#)
- [Arrêt des notifications indésirables](#)
- [Inspection du contenu intégrée](#)
- [Filtrage du contenu après mise en file d'attente](#)
- [Filtrage du contenu avant mise en file d'attente](#)
- [Applications MILTER avant mise en file d'attente](#)

Contrôle d'accès et de relais SMTP

- [Introduction au contrôle d'accès/de relais](#)
- [Délégation de la politique d'accès](#)
- [Vérification des adresses](#)
- [Accès par client, utilisateur,...](#)
- [Support ETRN](#)
- [LAN connecté via UUCP](#)

Tables de correspondances (bases de données)

- [Introduction aux tables de correspondances](#)
- [Howto CDB](#)
- [Howto BD Berkeley](#)
- [Howto LDAP](#)
- [Howto MySQL](#)
- [Howto PCRE](#)
- [Howto PostgreSQL](#)

Support des listes de diffusion

- Support qmail/ezmlm (*)
- Support VERP

Environnements spécifiques

- Problèmes spécifiques à Linux
- Problèmes spécifiques à NFS
- Support Ultrix

Autres agents de livraison du courrier

- Cyrus (*)
- Maldrop
- LMTP (*)

Autres sujets

- Présentation de l'architecture
- Tous les paramètres de main.cf
- Toutes les pages de manuel
- Rejet des destinataires locaux inconnus
- Classes d'adresses
- Howto cache des connexions
- Support DSN de Postfix
- Guide pour les développeurs de package
- Ordonnanceur de la file d'attente
- Commande XCLIENT
- Commande XFORWARD

(*) Ces documents seront disponibles ultérieurement sur <http://www.postfix.org/> et ses sites miroirs.

Configuration de Postfix –

Éléments de base

Introduction

Postfix a plusieurs centaines de paramètres de configuration qui sont contrôlés par l'intermédiaire du fichier **main.cf**. Heureusement, ils ont des valeurs par défaut. Dans la plupart des cas, vous devez configurer seulement deux ou trois paramètres avant de pouvoir employer le système de courrier:

- Fichiers de configuration de Postfix

On supposera ci-dessous que vous avez déjà installé Postfix sur votre système, soit en compilant le code source (comme décrit dans le fichier INSTALL), soit en installant une version déjà compilée.

Ce document parcourt la configuration de base de Postfix. Les informations pour des utilisations particulières telles les clusters de messagerie, les firewalls ou clients reliés par modem peuvent être trouvées dans le fichier STANDARD CONFIGURATION README mais ne consultez pas cette page avant d'avoir compris les éléments exposés ici.

Les premiers paramètres dignes d'intérêt indiquent l'identité de la machine et son rôle dans le réseau :

- Quel domaine afficher dans le courrier sortant
- De quels domaines recevoir le courrier
- De quels clients relayer le courrier
- De quelles destination relayer le courrier
- Quelle méthode de livraison : directe ou indirecte

Les valeurs par défaut pour beaucoup d'autres paramètres de configuration sont juste dérivées de ces derniers.

Le prochain paramètre intéressant commande la quantité de courrier envoyée au postmaster local :

- Quels événements doivent être rapportés à l'administrateur

Soyez sûr de placer le suivant correctement si vous êtes derrière un proxy ou un traducteur d'adresses réseau, et vous utilisez serveur de MX backup pour un autre domaine :

- Adresses réseau de Proxy/NAT

Les processus démons de Postfix fonctionnent en arrière plan et loguent les problèmes ainsi que leur activité normale via Syslog. Vous devriez faire attention aux éléments suivants :

- Ce que vous devez savoir sur les logs de Postfix

Si votre machine doit être sécurisée, vous pourriez vouloir lancer Postfix dans un environnement **chroot** :

- Lancer Postfix en environnement chroot

Si vous utilisez Postfix sur une interface réseau virtuelle, ou si d'autres serveurs de mail fonctionnent sur votre machine sur des interfaces virtuelles, vous devriez regarder les paramètres ci-dessous :

- Mon nom de machine
- Mon nom de domaine
- Mes adresses de réseau

Fichiers de configuration de Postfix

Par défaut, les fichiers de configuration de Postfix se trouvent dans le répertoire `/etc/postfix`. Les deux plus importants sont `main.cf` et `master.cf` ; ces fichiers doivent appartenir à root. Donner à quelqu'un d'autre les droits d'écriture sur ces deux fichiers (ou sur leurs répertoires parents) revient à lui donner des privilèges root.

Un minimum de paramètres doivent être configurés dans `/etc/postfix/main.cf`. Les paramètres ressemblent à des variables shell avec deux différences importantes : la première est que Postfix ne sait pas interpréter les apostrophes comme un shell Unix.

Pour renseigner un paramètre :

```
/etc/postfix/main.cf:
    parameter = value
```

et pour l'utiliser, il suffit de le faire précéder par un `$` :

```
/etc/postfix/main.cf:
    other_parameter = $parameter
```

Vous pouvez utiliser `$parameter` avant que sa valeur soit renseignée (c'est la seconde différence avec un shell Unix). Le langage de configuration de Postfix utilise une évaluation paresseuse et ne regarde la valeur d'un paramètre que lorsqu'il est utilisé.

Postfix utilise des bases de données entre autres pour le contrôle d'accès et les réécritures d'adresses. La page [DATABASE README](#) présente le fonctionnement de Postfix avec des bases Berkeley, LDAP, SQL et d'autres types. Ci-dessous un exemple d'invocation d'une base :

```
/etc/postfix/main.cf:
    virtual_alias_maps = hash:/etc/postfix/virtual
```

A chaque changement des fichiers `main.cf` ou `master.cf`, lancez la commande suivante en tant que root pour prendre en considération ces changements :

```
# postfix reload
```

Quel domaine afficher dans le courrier sortant

Le paramètre myorigin indique le domaine qui apparaît dans le courrier envoyé à partir de cette machine. La valeur par défaut est le nom de machine, \$myhostname, qui vaut par défaut le nom de la machine. À moins que vous gériez un site vraiment petit, vous voudrez probablement changer cela en \$mydomain, dont la valeur par défaut est le domaine parent de la machine.

Documentation de Postfix en français

Pour la cohérence entre les adresses d'expédition et de réception, myorigin indique également le domaine par défaut qui est automatiquement ajouté aux adresses de destination non qualifiées.

Exemples (utilisez seulement une ligne parmi les suivantes) :

```
/etc/postfix/main.cf
myorigin = $myhostname (défaut : envoie le courrier comme "user@$myhostname")
myorigin = $mydomain (probablement souhaitable : "user@$mydomain")
```

De quels domaines recevoir le courrier

Le paramètre mydestination indique les domaines pour lesquels cette machine délivrera le courrier localement, au lieu de le transmettre à une autre machine. La valeur par défaut est de recevoir le courrier à destination de la machine elle-même. Regardez la page VIRTUAL_README pour configurer les domaines hébergés par Postfix.

Vous pouvez indiquer zéro ou plusieurs nom de domaine, */des/fichiers* et/ou des tables de correspondance type:name (telles hash:, btree:, nis:, ldap:, ou mysql:), séparées par des espaces et/ou des virgules. */un/fichier* est remplacé par son contenu; type:name demande qu'une consultation de table soit faite et détermine simplement l'existence : le résultat de la consultation est ignoré.

IMPORTANT : Si votre machine est un serveur de mail pour son domaine entier, vous devez énumérer \$mydomain.

Exemple 1 : valeur par défaut.

```
/etc/postfix/main.cf :
mydestination = $myhostname localhost.$mydomain localhost
```

Exemple 2 : serveur de mail d'un domaine entier.

```
/etc/postfix/main.cf :
mydestination = $myhostname localhost.$mydomain localhost $mydomain
```

Exemple 3 : machine avec plusieurs enregistrements DNS de type A.

```
/etc/postfix/main.cf :
mydestination = $myhostname localhost.$mydomain localhost
www.$mydomain ftp.$mydomain
```

Attention : afin d'éviter des boucles de distribution du courrier, vous devez énumérer tous les noms d'hôtes de la machine, incluant \$myhostname, et localhost.\$mydomain.

De quels clients relayer le courrier

Par défaut, Postfix relaie le courrier des clients des réseaux autorisés et des domaines autorisés. Les réseaux autorisés sont définis par le paramètre mynetworks. La valeur par défaut autorise tous les clients des sous-réseaux IP auxquels la machine est reliée.

IMPORTANT : Si votre machine est connectée sur un WAN, la valeur par défaut de mynetworks risque d'être trop laxiste.

Exemples (utilisez l'un d'entre eux) :

```
/etc/postfix/main.cf :  
  mynetworks_style = subnet (défaut : autorise les sous-réseaux raccordés)  
  mynetworks_style = host   (sécurisé : n'autorise que la machine locale)  
  mynetworks = 127.0.0.0/8   (sécurisé : n'autorise que la machine locale)  
  mynetworks = 127.0.0.0/8 168.100.189.2/32
```

Vous pouvez spécifier les réseaux autorisés dans le fichier main.cf ou vous pouvez laisser Postfix le faire pour vous (comportement par défaut). Le résultat dépend de la valeur du paramètre mynetworks_style.

- Ecrivez "mynetworks_style = host" lorsque Postfix ne doit router que le courrier de sa machine.
- Ecrivez "mynetworks_style = subnet" (défaut) lorsque Postfix doit router le courrier des clients SMTP connectés sur le même sous-réseau. Sur Linux, cela ne fonctionne qu'avec les interfaces mentionnées avec la commande "ifconfig".
- Ecrivez "mynetworks_style = class" lorsque Postfix doit router le courrier des clients SMTP de la même classe d'adresse IP (A/B/C) que celle de la machine. N'utilisez pas cette option sur une connexion par modem : cela revient à autoriser tout le réseau de votre fournisseur d'accès. Autrement, renseignez explicitement mynetworks comme décrit ci-dessus.

Si vous renseignez mynetworks, Postfix ignore le paramètre mynetworks_style. Pour indiquer une liste de réseaux autorisés, écrivez les réseaux sous la forme CIDR (réseau/masque) ; par exemple :

```
/etc/postfix/main.cf :  
  mynetworks = 168.100.189.0/28, 127.0.0.0/8
```

Vous pouvez également indiquer le chemin absolu d'un fichier au lieu de lister les réseaux dans le fichier main.cf.

De quelles destination relayer le courrier

Par défaut, Postfix relaie le courrier étranger (provenant de clients hors réseaux autorisés) seulement vers les destinations autorisées. Les destinations extérieures autorisées sont définies avec le paramètre de configuration relay_domains. Par défaut, Postfix autorise tous les domaines (et sous-domaines) listés dans le paramètre mydestination.

Exemples (n'utilisez qu'un seul d'entre eux)

```
/etc/postfix/main.cf :  
  relay_domains = $mydestination (défaut)  
  relay_domains =                (sécurisé : ne relaie aucun courrier venant d'étrangers)  
  relay_domains = $mydomain      (relaie le courrier vers mon domaine et ses sous-domaines)
```

Quelle méthode de livraison : directe ou indirecte

Par défaut, Postfix tente de délivrer le courrier directement sur Internet. Suivant votre environnement, ce n'est pas toujours possible ou souhaitable. Par exemple, votre système peut être éteint en dehors des heures ouvrées, il peut être derrière un firewall, ou bien encore être connecté via un fournisseur d'accès qui n'autorise pas la livraison directe du courrier. Dans ces cas, vous devez configurer Postfix pour effectuer les livraisons via un relais.

Exemples (n'utilisez qu'un seul d'entre eux)

De quelles destination relayer le courrier

```
/etc/postfix/main.cf :  
  relayhost = (défaut : livraison directe)  
  relayhost = $mydomain (livraison via la passerelle de messagerie de mon domain  
  relayhost = [mail.$mydomain] (livraison via la passerelle mail.$mydomain)  
  relayhost = [mail.isp.tld] (livraison via la passerelle du fournisseur d'accès)
```

L'utilisation des [] évite la consultation des champs MX du DNS. Ne vous inquiétez pas si vous ne savez pas de quoi il s'agit. Assurez-vous seulement d'ajouter les [] autour du commutateur de messagerie fournit par votre FAI, sinon le courrier pourrait ne pas être livré.

La page [STANDARD CONFIGURATION README](#) montre plusieurs exemples pour les réseaux protégés par firewall ou raccordés par un modem.

Quels incidents rapporter au postmaster

Il est souhaitable d'installer un alias postmaster dans la table [aliases\(5\)](#) pointant vers le nom d'une personne. Cet alias doit exister, de sorte que les gens puissent signaler des problèmes de distribution de courrier. Lorsque vous modifiez la table [aliases\(5\)](#), vérifiez que vous redirigez également le courrier du super-utilisateur.

```
/etc/aliases:  
  postmaster: Vous  
  root: Vous
```

Lancez la commande "newaliases" après avoir changé le fichier /etc/aliases. Au lieu de /etc/aliases, votre fichier d'alias peut être situé ailleurs. Utilisez la commande "postconf [alias_maps](#)" pour le trouver.

Le système Postfix lui-même signale également des problèmes à l'alias postmaster. Vous pouvez ne pas être intéressé par tous les types d'événements, aussi ce mécanisme de report d'incidents est configurable. La valeur par défaut signale seulement les problèmes sérieux (ressource, logiciel) au postmaster :

Défaut :

```
/etc/postfix/main.cf  
  notify\_classes = resource, software
```

Ci dessous la signification des classes :

bounce

Informe le postmaster du courrier non livrable. Si le courrier est non livrable, un avis de non-livraison simple est envoyé, avec une copie du message qui n'a pas été livré, ou envoie une retranscription de la session SMTP en cas de rejet du message. Pour des raisons d'intimité, la copie envoyée au postmaster d'un avis de non-livraison simple est tronquée après les en-têtes de message originaux. Voyez également le paragraphe [luser_relay](#) et "2bounce" (ci-dessous). La notification est envoyée à l'adresse indiquée au paramètre de configuration [bounce_notice_recipient](#) (défaut : postmaster).

2bounce

Si un avis de non-livraison simple est lui-même non livrable, le postmaster reçoit un double de l'avis de non-livraison avec une copie de l'avis de non-livraison simple (non tronqué). La notification est envoyée à l'adresse indiquée au paramètre de configuration [2bounce_notice_recipient](#) (défaut : postmaster).

delay

Informe le postmaster des courriers retardés. Dans ce cas, le postmaster reçoit les en-têtes de message seulement. La notification est envoyée à l'adresse indiquée au paramètre de configuration delay_notice_recipient (défaut : postmaster).

policy

Informe le postmaster des demandes clients qui ont été rejetées en raison de la politique de restriction UCE (anti-spam). Le postmaster reçoit une transcription de la session SMTP entière. La notification est envoyée à l'adresse indiquée au paramètre de configuration error_notice_recipient (défaut : postmaster).

protocol

Informe le postmaster des erreurs de protocole (côté client ou serveur) ou des tentatives d'un client d'exécuter des commandes non implémentées. Le postmaster reçoit une transcription de la session SMTP entière. La notification est envoyée à l'adresse indiquée au paramètre de configuration error_notice_recipient (défaut : postmaster).

resource

Informe le postmaster des courriers non délivrés en raison d'un problème de ressource (par exemple, la file d'attente écrivant les erreurs). La notification est envoyée à l'adresse indiquée au paramètre de configuration error_notice_recipient (défaut : postmaster).

software

Informe le postmaster des courriers non délivrés en raison de problèmes logiciels. La notification est envoyée à l'adresse indiquée au paramètre de configuration error_notice_recipient (défaut : postmaster).

Adresse réseau proxy/NAT

Certains serveurs sont connectés à Internet via un traducteur d'adresse (NAT) ou proxy. Cela signifie que les clients Internet se connectent sur l'adresse du traducteur ou proxy au lieu de se connecter sur l'adresse du serveur de courrier. Le traducteur ou proxy transfère la connexion sur l'adresse réseau du serveur de courrier, mais Postfix ne le sait pas.

Si vous utilisez Postfix derrière un traducteur ou proxy, vous devez renseigner le paramètre proxy_interfaces en lui indiquant toutes les adresses externes des traducteurs ou proxies. Vous pouvez utiliser des noms de machines au lieu d'adresses réseaux.

IMPORTANT : Vous devez indiquer les adresses de vos proxy/NAT lorsque votre système est une machine de secours (MX backup) pour d'autres domaines, autrement les courriers risquent de boucler si le serveur MX principal ne fonctionne pas.

Exemple : machine derrière un traducteur et faisant fonctionner un MX backup

```
/etc/postfix/main.cf
proxy_interfaces = 1.2.3.4 (l'adresse externe du traducteur/proxy)
```

Ce que vous devez savoir sur les logs de Postfix

Les démons Postfix fonctionnent en arrière plan et journalisent les problèmes et l'activité normale via le démon syslog. Le processus syslogd trie les événements par classe et sévérité et les ajoute aux fichiers journaux. Les classes, niveaux et noms de fichiers journaux sont généralement indiqués dans le fichier

Documentation de Postfix en français

/etc/syslog.conf. Au minimum, il doit contenir quelque chose comme :

```
/etc/syslog.conf
mail.err          /dev/console
mail.debug        /var/log/maillog
```

Après avoir modifié le fichier syslog.conf, envoyez un signal HUP au processus syslogd.

IMPORTANT : beaucoup d'implémentations de syslogd ne créent pas les fichiers. Vous devez les créer avant de (re)lancer syslogd.

IMPORTANT : sur Linux, vous devez faire précéder le fichier du signe "-" (ex: -/var/log/maillog), autrement le processus syslogd utilisera plus de ressources que Postfix.

Normalement, le nombre de problèmes restera faible, mais une bonne idée consiste à lancer toutes les nuits avant la rotation des journaux :

```
# postfix check
# egrep '(reject|warning|error|fatal|panic):' /fichier/journal
```

- La première ligne (postfix check) invite Postfix à rapporter les problèmes de permission/appartenance.
- La seconde ligne examine les problèmes rapportés par le logiciel de courrier et renvoie ceux concernant les relais et requêtes malformées. Ceci peut produire beaucoup de lignes. Vous pouvez appliquer d'autres filtres pour éliminer les informations inintéressantes.

La page [DEBUG README](#) montre la signification des "warnings" et autres labels utilisés dans les journaux générés par Postfix.

Lancer Postfix en environnement "chroot"

Les démons de Postfix peuvent être configurés (via le fichier master.cf) pour fonctionner dans une cage chroot. Le processus fonctionne avec un minimum de privilèges et avec un accès au système de fichier limité à la file d'attente (/var/spool/postfix). Ceci fournit une barrière significative contre les intrusions. La barrière n'est pas impénétrable (le chroot limite seulement l'accès au système de fichier), mais chaque petit pas compte.

A l'exception des démons de Postfix qui délivrent le courrier localement ou qui exécutent des commandes extérieures à Postfix, tous les démons peuvent être mis en cage chroot.

Les sites exigeant un haut niveau de sécurité considéreront sans doute que tous les démons accessibles par réseau doivent être en cage : les processus [smtp\(8\)](#) et [smtpd\(8\)](#), et peut-être le client [lmtp\(8\)](#). Tous les démons du serveur de courrier du domaine de l'auteur (porcupine.org) qui peuvent l'être fonctionnent en cage.

Le fichier /etc/postfix/master.cf par défaut ne configure aucun démon en chroot. Pour activer cette fonctionnalité, éditez le fichier /etc/postfix/master.cf et suivez les instructions incluses dans le fichier. Lorsque vous avez terminé, lancez "postfix reload" pour valider les changements.

Notez qu'un démon en cage chroot résout les noms de fichiers relativement au répertoire de la file d'attente (/var/spool/postfix). Pour réussir une mise en cage, beaucoup de systèmes UNIX nécessitent la création de certains fichiers ou inodes. Le répertoire "examples/chroot-setup" des sources de Postfix recèle différents

scripts pouvant vous aider sur différents systèmes d'exploitation.

En plus, vous devrez certainement configurer syslogd pour écouter une socket dans la cage. Ci-dessous un exemple de lignes de commande activant ceci sur différents systèmes :

FreeBSD : `syslogd -l /var/spool/postfix/var/run/log`

Linux, OpenBSD : `syslogd -a /var/spool/postfix/dev/log`

Mon nom de machine

Le paramètre myhostname indique le nom de la machine exploitant le système Postfix sous forme qualifiée (machine.domaine). \$myhostname est utilisé comme valeur par défaut dans beaucoup d'autres paramètres de configuration de Postfix.

Par défaut, myhostname contient le nom de la machine. Si votre nom de machine n'a pas la forme machine.domaine, ou si vous utilisez Postfix sur une interface virtuelle, vous devrez indiquer le nom de domaine que le système de courrier devrait employer.

Autrement si vous renseignez le paramètre mydomain, Postfix utilisera cette valeur pour générer la valeur du paramètre myhostname au bon format.

Exemples : (utilisez seulement l'un d'entre eux)

```
/etc/postfix/main.cf
myhostname = host.local.domain (le nom d'hôte n'est un nom qualifié)
myhostname = host.virtual.domain (interface virtuelle)
myhostname = virtual.domain (interface virtuelle)
```

Mon nom de domaine

Le paramètre mydomain indique le domaine de rattachement de \$myhostname. Par défaut il est dérivé de \$myhostname amputé de la première partie (sauf si le résultat donne un nom de domaine racine).

Inversement, si vous spécifiez mydomain dans main.cf, Postfix utilisera sa valeur pour générer une valeur qualifiée pour le paramètre myhostname.

Exemples (utilisez seulement l'un d'entre eux) :

```
/etc/postfix/main.cf
mydomain = local.domain
mydomain = virtual.domain (interface virtuelle)
```

Mes adresses réseau

Le paramètre inet_interfaces indique toutes les adresses d'interface réseau sur lesquelles le système Postfix doit écouter ; le courrier adressé à "utilisateur@[adresse de réseau] sera délivré localement, comme s'il était adressé à un domaine énuméré dans \$mydestination.

Vous pouvez outrepasser le paramètre inet_interfaces dans le fichier master.cf en faisant précéder un nom de

serveur par une adresse IP.

Par défaut Postfix écoute sur toutes les interfaces actives. Si vous utilisez des serveurs de mail sur des interfaces virtuelles, vous devrez indiquer sur quelles interfaces écouter.

IMPORTANT : si vous utilisez un MTA sur des adresses virtuelles, vous devez explicitement renseigner le paramètre inet_interfaces pour l'interface de la machine pour le MTA non-virtuel qui reçoit le courrier pour la machine elle-même : ce MTA ne devrait jamais écouter sur les interfaces virtuelles ou vous risquez une boucle de messagerie.

Exemples (valeur par défaut) :

```
/etc/postfix/main.cf
inet_interfaces = all
```

Exemple : machine faisant fonctionner un ou plusieurs serveurs virtuels. Pour chaque instance de Postfix, utilisez seulement l'un d'entre eux.

```
/etc/postfix/main.cf
inet_interfaces = virtual.host.tld          (Postfix virtuel)
inet_interfaces = $myhostname localhost... (Postfix non-virtuel)
```

Note: vous devez arrêter et redémarrer Postfix lorsque ces paramètres changent.

Exemples de configuration

standard de Postfix

Objet de ce document

Ce document présente diverses configurations typiques de Postfix. Il vous sera utile après que vous ayez suivi les étapes de la configuration basique décrites à la page [BASIC CONFIGURATION README](#). Ne l'utilisez pas si vous n'avez pas déjà un Postfix fonctionnant pour l'envoi ou la réception locale.

La première partie montre quelques configurations standard permettant de résoudre certains problèmes spécifiques.

- [Postfix sur une machine Internet autonome](#)
- [Postfix sur un client sans rôle](#)
- [Postfix sur un réseau local](#)
- [Postfix sur une passerelle/pare-feu](#)

La deuxième partie vient en complément pour des machines dans un environnement particulier.

- [Livrer une partie seulement des courriers localement](#)
- [Utiliser Postfix derrière un pare-feu](#)
- [Configurer Postfix comme serveur MX pour un site distant](#)
- [Utiliser Postfix avec un modem](#)
- [Postfix sur des machines sans nom réel](#)

Postfix sur une machine Internet autonome

Postfix devrait fonctionner sans modifier la configuration sur une machine de ce type avec un accès direct à Internet. C'est ainsi qu'il s'installe lorsque vous téléchargez les sources via <http://www.postfix.org/>.

Vous pouvez utiliser la commande "**postconf -n**" pour voir quels sont les paramètres modifiés par votre main.cf. Peu de paramètres doivent être configurés pour installer une machine autonome après les recommandations de la page [BASIC CONFIGURATION README](#) :

```
/etc/postfix/main.cf:
# Optional: envoie les courriers avec l'adresse utilisateur@nom-de-domaine au lieu de v
#myorigin = $mydomain

# Optional: renseigne l'adresse externe du pare-feu/proxy.
#proxy_interfaces = 1.2.3.4

# Ne pas relayer les courriers des autres machines.
mynetworks_style = host
relay_domains =
```

Voyez aussi le paragraphe "[Postfix sur des machines sans nom Internet réel](#)" s'il s'applique à votre configuration.

Postfix sur un client sans rôle

Un client sans rôle est une machine qui peut seulement envoyer du courrier. Il ne reçoit pas de courrier du réseau et n'en livre aucun localement. Ces clients utilisent généralement les protocoles POP, IMAP ou un partage NFS pour accéder aux boîtes-aux-lettres.

Dans cet exemple, nous supposons que le nom de domaine est "exemple.com" et que la machine s'appelle "nullclient.exemple.com". Comme précédemment, les exemples ne montrent que les paramètres n'utilisant pas la valeur par défaut.

```
1 /etc/postfix/main.cf:
2   myorigin = $mydomain
3   relayhost = $mydomain
4   inet_interfaces = 127.0.0.1
5   local_transport = error:local delivery is disabled
6
7 /etc/postfix/master.cf:
8   Commentez la ligne correspondant à l'agent local
```

Explications :

- Ligne 2 : Utiliser des adresses d'expédition "user@exemple.com" (au lieu de "user@nullclient.exemple.com"), s'il n'est aucun besoin d'utiliser "user@nullclient.exemple.com".
- Ligne 3 : Transférer tout le courrier vers le serveur responsable du domaine "exemple.com". Ceci évite au client de tenter de livrer du courrier vers une destination qu'il ne peut joindre directement.
- Ligne 4 : Ne pas accepter de courrier en dehors de la boucle locale.
- Lignes 5 à 8 : Désactiver la livraison locale du courrier. Tous les courriers sont routés vers le serveur indiqué à la ligne 3.

Postfix sur un réseau local

Ce paragraphe décrit un environnement comprenant un serveur de courrier et plusieurs machines recevant et envoyant du courrier. Toutes les machines sont configurées pour envoyer le courrier en utilisant des adresses sous la forme "user@exemple.com" et en recevoir à destination d'adresses sous la forme "user@hostname.exemple.com". Le serveur principal reçoit également du courrier à destination de "user@exemple.com". Nous appellerons cette machine "mailhost.exemple.com".

Une conséquence de cette configuration est que le courrier à destination de root et des autres comptes système est envoyé au serveur principal. Reportez-vous au paragraphe "[Livrer une partie seulement des courriers localement](#)" ci-dessous pour voir les solutions envisageables.

Comme précédemment, les exemples ne montrent que les paramètres n'utilisant pas la valeur par défaut.

Tout d'abord voyons la configuration – plus simple – des autres machines. Elles envoient le courrier avec des adresse sous la forme "user@exemple.com" et le reçoivent à destination de "user@hostname.exemple.com".

```
1 /etc/postfix/main.cf:
2   myorigin = $mydomain
3   mynetworks = 127.0.0.0/8 10.0.0.0/24
```


Documentation de Postfix en français

```
4      relay_domains =  
5      # Optionel: transfère tout le courrier au serveur principal  
6      #relayhost = $mydomain
```

Explications :

- Ligne 2 : Envoie le courrier sous la forme "user@exemple.com".
- Ligne 3 : Indique le réseau autorisé.
- Ligne 4 : Cette machine ne relaie pas le courrier des réseaux non autorisés.
- Ligne 6 : Cette ligne est nécessaire si l'accès à Internet n'est pas direct. Voyez ci-dessous "[Postfix derrière un pare-feu](#)".

Maintenant voyons la configuration du serveur de courrier. Cette machine envoie le courrier avec une adresse "user@exemple.com" et accepte celui à destination de "user@hostname.exemple.com" et "user@exemple.com".

```
1 DNS:  
2      exemple.com      IN      MX  10 mailhost.exemple.com.  
3  
4 /etc/postfix/main.cf:  
5      myorigin = $mydomain  
6      mydestination = $myhostname localhost.$mydomain localhost $mydomain  
7      mynetworks = 127.0.0.0/8 10.0.0.0/24  
8      relay_domains =  
9      # Optionel: transfère le courrier extérieur au pare-feu  
10     #relayhost = [pare-feu.exemple.com]
```

Explications :

- Ligne 2 : Envoie le courrier du domaine "exemple.com" vers la machine mailhost.exemple.com. N'oubliez pas le "." en fin de ligne.
- Ligne 5 : Envoie le courrier avec un adresse "user@exemple.com".
- Ligne 6 : Cette machine est la destination du courrier à destination de "exemple.com" et de ses machines itself.
- Ligne 7 : Indique le réseau interne.
- Ligne 8 : Cette machine ne relaie pas le courrier des réseaux externes.
- Ligne 10 : C'est utile seulement lorsque le serveur de courrier doit transférer le courrier extérieur via un serveur de mail sur un pare-feu. Les [] indiquent à Postfix de ne pas effectuer de recherche MX.

Dans un tel environnement, les utilisateurs ont accès à leur courrier par un ou plusieurs des protocoles suivants :

- Accès via NFS ou équivalent.
- Accès via POP ou IMAP.
- Boîte-aux-lettres sur la machine habituelle de l'utilisateur.

Dans ce dernier cas, chaque utilisateur a un alias sur le serveur qui transfère le courrier sur sa machine habituelle :

```
/etc/aliases :  
joe :      joe@joes.preferred.machine  
jane :     jane@janes.preferred.machine
```

Sur certains systèmes, la base des alias n'est pas `/etc/aliases`. Pour la trouver, lancez la commande "**postconf alias maps**".

Lancez la commande "**newaliases**" à chaque changement du fichier d'alias.

Postfix sur une passerelle/pare-feu

L'idée ici est de faire fonctionner Postfix sur une passerelle (ou un pare-feu) qui transfère le courrier à destination de "exemple.com" vers une machine interne mais rejete le courrier à destination de "quelque-chose.exemple.com". Problème : avec "relay_domains = exemple.com", le pare-feu accepte également le courrier pour "quelque-chose.exemple.com".

Note : cet exemple nécessite Postfix version 2.0 ou supérieure. Pour connaître la version de Postfix, exécutez la commande "**postconf mail version**".

Cette exemple est présenté en plusieurs parties. La première interdit la livraison locale sur le pare-feu, le rendant plus difficile à corrompre.

```
1 /etc/postfix/main.cf :
2     myorigin = exemple.com
3     mydestination =
4     local_recipient_maps =
5     local_transport = error :local mail delivery is disabled
6
7 /etc/postfix/master.cf :
8     Commentez la ligne correspondant à l'agent local
```

Explications :

- Ligne 2 : Envoie le courrier de cette machine comme "utilisateur@exemple.com", s'il n'existe aucune raison de l'envoyer comme "utilisateur@pare-feu.exemple.com".
- Lignes 3–8 : Désactive la livraison locale sur la machine pare-feu.

Pour des raisons techniques, le pare-feu peut devoir recevoir le courrier à destination de `postmaster@[adresse IP du pare-feu]`. La deuxième configuration proposée ajoute le support de cette fonctionnalité, et en supplément nous ajoutons la boîte-aux-lettres `abuse@[adresse IP du pare-feu]`. Tout le courrier à destination de ces deux adresses est envoyé vers une adresse interne.

```
1 /etc/postfix/main.cf :
2     virtual_alias_maps = hash :/etc/postfix/virtual
3
4 /etc/postfix/virtual :
5     postmaster      postmaster@exemple.com
6     abuse           abuse@exemple.com
```

Explication :

- Puisque mydestination est vide, (voir le premier exemple), seule l'adresse correspondant exactement à \$inet_interfaces ou à \$proxy_interfaces sont considérées locales. Ainsi "localpart@[a.d.d.r]" correspond à "localpart" dans canonical(5) et virtual(5). Ceci impose de définir l'adresse IP du pare-feu dans le fichier de configuration de Postfix.

La dernière partie de cet exemple paramètre le transfert du courrier, ce qui est le but essentiel.

Documentation de Postfix en français

```
1 /etc/postfix/main.cf :
2   mynetworks = 127.0.0.0/8 12.34.56.0/24
3   relay_domains = exemple.com
4   parent_domain_matches_subdomains =
5       debug_peer_list smtpd_access_maps
6   smtpd_recipient_restrictions =
7       permit_mynetworks reject_unauth_destination
8
9   relay_recipient_maps = hash :/etc/postfix/relay_recipients
10  transport_maps = hash :/etc/postfix/transport
11
12 /etc/postfix/relay_recipients :
13   user1@exemple.com    x
14   user2@exemple.com    x
15   . . .
16
17 /etc/postfix/transport :
18   exemple.com    smtp :[serveur-interne.exemple.com]
```

Explication :

- Lignes 1 à 7 : Accepter le courrier provenant du réseau interne (\$mynetworks), et accepter le courrier extérieur à destination de "user@exemple.com" mais pas "user@anything.exemple.com" (lignes 4 et 5).
- Lignes 9 et 12 à 14 : Définissent la liste des adresses valides du domaine "exemple.com" qui peuvent recevoir du courrier depuis Internet. Ceci évite d'encombrer les files d'attente avec du courrier non livrable. Si vous ne pouvez maintenir une liste des destinataires valides, vous pouvez indiquer "relay_recipient_maps =" (valeur vide), ou écrire "@exemple.com x" dans la table relay_recipients.
- Lignes 10, 17–18 : Transfère le courrier du domaine "exemple.com" à une machine interne. Les [] indiquent à Postfix de ne pas effectuer de recherche MX.

Indiquez **dbm** au lieu de **hash** si votre machine utilise des fichiers **dbm** au lieu de **db**. Pour voir les types de tables de correspondances supportées par Postfix, utilisez la commande "**postconf -m**".

Lancez la commande "**postmap /etc/postfix/relay_recipients**" à chaque changement de la table relay_recipients.

De même, lancez la commande "**postmap /etc/postfix/transport**" à chaque changement de la table transport.

Dans certaines installations, il peut y avoir des instances séparées de Postfix traitant le courrier entrant et sortant sur un firewall multi-sites. L'instance entrante de Postfix a un serveur SMTP écoutant sur l'interface externe du firewall et l'instance sortante sur l'interface interne. Dans une telle configuration il est tentant de configurer \$inet_interfaces dans chaque instance avec seulement l'adresse de l'interface correspondante.

Dans la plupart des cas, utiliser inet_interfaces en ce sens ne fonctionnera pas, car comme expliqué au paragraphe \$inet_interfaces du manuel de référence, l'agent de livraison smtp(8) utilisera également l'adresse de l'interface spécifiée comme adresse source des connexions sortantes et ne pourra pas joindre les domaines de "l'autre côté" du pare-feu. Les symptômes caractérisant ce dysfonctionnement sont que le firewall ne peut se connecter aux machines dans cette situation. Voyez la documentation du paramètre inet_interfaces.

Livrer une partie seulement des courriers localement.

En utilisant des adresses "utilisateur@exemple.com" (au lieu de "utilisateur@machine.exemple.com") le

courrier de "root" et des autres comptes système est envoyé vers le serveur de courrier central. Pour livrer ces courriers localement, vous pouvez utiliser les alias virtuels comme suit :

```
1 /etc/postfix/main.cf :
2     virtual_alias_maps = hash :/etc/postfix/virtual
3
4 /etc/postfix/virtual :
5     root      root@localhost
6     . . .
```

Explications :

- Ligne 5 : Comme indiqué dans la page de manuel virtual(5), le nom "root" correspond à "root@site" lorsque "site" vaut \$myorigin, lorsque "site" est dans la liste \$mydestination, ou lorsqu'il correspond à \$inet_interfaces ou \$proxy_interfaces.

Utiliser Postfix derrière un pare-feu

La plus simple façon d'utiliser Postfix derrière un pare-feu est d'envoyer tout le courrier vers une machine passerelle et de laisser cette machine gérer le routage interne/externe. Des exemples de cette configuration sont vu au paragraphe réseau local ci-dessus. Une approche plus sophistiquée est d'envoyer seulement le courrier extérieur vers cette passerelle et de livrer directement le courrier local. C'est ce qu'utilise Wietse (le créateur de Postfix).

Note : cet exemple nécessite Postfix version 2.0 ou supérieure. Pour connaître la version de Postfix, lancez la commande "postconf mail_version".

L'exemple suivant montre un complément de configuration. Vous devez le combiner avec une configuration basique telles que présentées dans la première moitié de ce document.

```
1 /etc/postfix/main.cf :
2     transport_maps = hash :/etc/postfix/transport
3     relayhost =
4     # Optionnel pour une machine pas toujours active
5     #fallback_relay = [gateway.exemple.com]
6
7 /etc/postfix/transport :
8     # Internal delivery.
9     exemple.com      :
10    .exemple.com      :
11    # External delivery.
12    *                  smtp :[gateway.exemple.com]
```

Explications :

- Lignes 2 et 7 à 12 : Demande que le courrier de l'intranet soit délivré directement et que le courrier extérieur soit transféré à la passerelle. Bien entendu, cet exemple suppose que le site utilise des enregistrements DNS MX dans l'intranet. Les [] indiquent à Postfix de ne pas effectuer de recherche MX.
- Ligne 3 : IMPORTANT : ne renseignez pas le paramètre relayhost dans le fichier main.cf.
- Ligne 5 : Ceci évite d'encombrer les files d'attente lorsque des machines sont éteintes : Postfix tente de délivrer le courrier directement et transfère le courrier non livrable à une passerelle.

Indiquez **dbm** au lieu de **hash** si votre machine utilise des fichiers **dbm** au lieu de **db**. Pour voir les types de tables de correspondances supportées par Postfix, utilisez la commande "**postconf -m**".

Lancez la commande "**postmap /etc/postfix/transport**" à chaque changement de la table transport.

Configurer Postfix comme serveur MX pour un site distant

Ce paragraphe montre un complément de configuration. Vous devez combiner l'exemple avec une configuration basique telle que présenté dans la première moitié de ce document.

Lorsque votre serveur est une machine MX SECONDAIRE pour un site distant, c'est tout qu'il vous faut :

```
1 DNS :
2     le.domaine.a.sauvegarder      IN      MX 100 your.machine.tld.
3
4 /etc/postfix/main.cf :
5     relay_domains = . . . le.domaine.a.sauvegarder
6     smtpd_recipient_restrictions =
7         permit_mynetworks reject_unauth_destination
8
9     # Vous devez indiquer l'adresse extérieure du traducteur/proxy.
10    #proxy_interfaces = 1.2.3.4
11
12    relay_recipient_maps = hash :/etc/postfix/relay_recipients
13
14 /etc/postfix/relay_recipients :
15     user1@le.domaine.a.sauvegarder    x
16     user2@le.domaine.a.sauvegarder    x
17     . . .
```

Lorsque votre système est le serveur MX PRIMAIRE pour un site distant, vous devez indiquer en plus de l'exemple précédent :

```
18 /etc/postfix/main.cf :
19     transport_maps = hash :/etc/postfix/transport
20
21 /etc/postfix/transport :
22     le.domaine.a.sauvegarder      relay :[their.mail.host.tld]
```

Notes important :

- Ne listez pas le.domaine.a.sauvegarder dans mydestination.
- Ne listez pas le.domaine.a.sauvegarder dans virtual_alias_domains.
- Ne listez pas le.domaine.a.sauvegarder dans virtual_mailbox_domains.
- Lignes 1 à 7 : Transfert le courrier d'Internet à destination du domaine "le.domaine.a.sauvegarder" vers le serveur MX primaire de ce domaine.
- Ligne 10 : Ceci est un plus si Postfix reçoit le courrier via un traducteur ou un proxy montrant une autre adresse au monde que la machine locale.
- Lignes 12–16 : Définit la liste des adresses valides du domaine "le.domaine.a.sauvegarder". Ceci évite à votre file d'attente de stocker des courriers non livrables. Si vous ne pouvez maintenir une telle liste, vous devez indiquer "relay_recipient_maps =" (c'est à dire une valeur vide), ou vous devez indiquer une correspondance générique "@exemple.com x" dans la table relay_recipients.
- Ligne 22 : Les [] indiquent à Postfix de ne pas effectuer de recherche MX.

Indiquez **dbm** au lieu de **hash** si votre machine utilise des fichiers **dbm** au lieu de **db**. Pour voir les types de tables de correspondances supportées par Postfix, utilisez la commande "**postconf -m**".

Lancez la commande "**postmap /etc/postfix/transport**" à chaque changement de la table transport.

NOTE : N'utilisez pas la fonctionnalité fallback relay lorsque vous relayer du courrier pour un MX backup ou primaire. Le courrier bouclera entre le serveur MX et la machine fallback relay lorsque la destination ne pourra être jointe.

- Dans main.cf indiquez "relay transport = relay",
- dans master.cf indiquez "-o fallback relay =" à la fin de la ligne concernant relay.
- Dans la table transport, indiquez "relay:saut suivant..." comme partie droite pour les entrées correspondants à des MX principaux ou de secours.

Ce sont les valeurs par défaut sur les versions 2.2 et supérieures de Postfix.

Utiliser Postfix avec une connexion par modem

Ce paragraphe concerne les connexions modem déconnectées la majeure partie du temps. Pour les connexions modem reliées 24h/24 7j/7, voyez le paragraphe Postfix sur un réseau local ci-dessus.

Ce paragraphe montre un complément de configuration. Vous devez combiner l'exemple avec une configuration basique telle que présenté dans la première moitié de ce document.

Si vous n'avez pas votre propre nom de machine (comme avec une adresse IP dynamique), étudiez alors le paragraphe "Postfix sur des machines sans nom réel".

- Transférez tous vos messages vers votre fournisseur d'accès.

Si votre machine est déconnectée la majeure partie du temps, il y a peu d'opportunités pour délivrer le courrier extérieur. Il est préférable de donner le courrier à une machine connectée en permanence. Dans l'exemple suivant, les [] indiquent à Postfix de ne pas effectuer de recherche MX.

```
/etc/postfix/main.cf :  
    relayhost = [smtprelay.fai.com]
```

- Désactivez la livraison spontanée du courrier (si vous n'avez pas de connexion automatique).

En temps normal, Postfix délivre le courrier à sa convenance. Si votre machine nécessite une connexion automatique, cela déclenchera un appel téléphonique à chaque nouveau courrier et chaque fois que Postfix tentera de livrer un courrier retardé. Pour prévenir ce comportement, désactivez la livraison spontanée.

```
/etc/postfix/main.cf :  
    defer transports = smtp
```

- Désactivez les vérifications DNS pour les clients (réseau local connecté par modem).

```
/etc/postfix/main.cf :  
    disable dns lookups = yes
```

- Videz la file d'attente à chaque fois que le lien Internet est établi.

Insérez la ligne suivante dans vos scripts PPP ou SLIP :

Documentation de Postfix en français

`/usr/sbin/sendmail -q` (chaque fois que li lien Internet est actif)

L'emplacement exact de la commande `sendmail` dépend su système. Utilisez la commande "`postconf sendmail_path`" pour connaître son emplacement.

Pour savoir si la file d'attente est vidée, utilisez un script du genre :

```
#!/bin/sh

# Lance la livraison.
/usr/sbin/sendmail -q

# Laisse quelque temps à Postfix pour démarrer.
sleep 10

# Boucle qui attend que la file soit vidée.
while mailq | grep '^[^ ]*\*' >/dev/null
do
    sleep 10
done
```

Si vous avez désactivé la livraison du courrier SMTP spontanée, vous devez également lancer "`sendmail -q`" à chaque fois que le lien est actif. Ainsi le nouveau courrier est posté.

Postfix sur des machines sans nom réel

Ce paragraphe concerne les machine qui n'ont pas de nom Internet. Généralement, il s'agit de machines obtenant leur adresse dynamiquement par DHCP ou modem. Postfix vous laissera envoyer et recevoir du courrier entre comptes sur la même machine avec un nom fantaisiste. Toutefois, vous ne pouvez pas utilisez ce nom lorsque vous envoyez du courrier sur Internet car personne ne pourra vous répondre. Dans les faits, beaucoup de sites refusent le courrier de domaines inexistants

Note : les explications suivantes dépendent de la version de Postfix. Pour connaître la version que vous utilisez, lancez la commande "`postconf mail_version`".

Solution 1: Postfix version 2.2 et supérieures

Postfix 2.2 utilise le remplacement d'adresses generic(5) pour changer les adresses locales fantaisistes par des adresses Internet valides. Ces remplacements ne concernent QUE le courrier qui quitte la machine et pas celui échangé entre utilisateurs de la même machine.

L'exemple suivant présente une configuration additionnelle. Vous devez la combiner avec une configuration de base telle que présentée dans la première partie de ce document.

```
1 /etc/postfix/main.cf:
2     smtp_generic_maps = hash:/etc/postfix/generic
3
4 /etc/postfix/generic:
5     user1@localdomain.local      comptel@mon.fai
6     user2@localdomain.local      compte2@mon.autre.fai
7     @localdomain.local          compte3+local@mon.fai
```

Lorsque le courrier est envoyé à une machine extérieure via SMTP :

- la ligne 5 remplace *user1@localdomain.local* par l'adresse fournie par son FAI,
- la ligne 6 remplace *user2@localdomain.local* par l'adresse fournie par son FAI,
- la ligne 7 remplace les autres adresses locales par la troisième adresse mais y ajoute une extension d'adresse *+local* (cat exemple suppose que le FAI supporte ce style d'extensions).

Indiquez **dbm** au lieu de **hash** si votre machine utilise des fichiers **dbm** au lieu de **db**. Pour voir les types de tables de correspondances supportées par Postfix, utilisez la commande "**postconf -m**".

Lancez la commande "**postmap /etc/postfix/generic**" à chaque changement de la table générique.

Solution 2 : Postfix version 2.1 et antérieures

La solution avec les versions plus anciennes de Postfix est d'utiliser autant que possible des adresses Internet valides et de laisser Postfix faire la correspondance entre ces adresses valides et les adresses locales. Ainsi vous pouvez envoyer du courrier sur Internet et vers les adresses locales non valides sur Internet

L'exemple suivant montre un complément de configuration qui doit être combiné avec l'un de ceux présentés dans la première moitié de ce document.

```
1 /etc/postfix/main.cf :
2     myhostname = hostname.localdomain
3     mydomain = localdomain
4
5     canonical_maps = hash :/etc/postfix/canonical
6
7     virtual_alias_maps = hash :/etc/postfix/virtual
8
9 /etc/postfix/canonical :
10    your-login-name      your-account@your-isp.com
11
12 /etc/postfix/virtual :
13    your-account@your-isp.com      your-login-name
```

Explications :

- Lignes 2 et 3 : Indiquez votre domaine local ici. N'utilisez pas de domaines existant réellement sur Internet. Voyez la [RFC 2606](#) pour obtenir des exemples de domaines dont on garantit qu'ils n'appartiennent à personne.
- Lignes 5, 9 et 10 : Ceci crée la correspondance entre "votre-login@hostname.localdomain" et "votre-compte@votre-fai.com". Ces lignes sont recommandées.
- Lignes 7, 12 et 13 : Livre le courrier destiné à "votre-compte@votre-fai.com" localement. Ces lignes ne sont pas obligatoires, mais utiles.

Traductions d'adresses

Introduction au système de traduction des adresses de Postfix

La réécriture des adresses fait partie intégrante du système de courrier Postfix. Postfix réécrit les adresses dans beaucoup de situations. Certaines sont surtout cosmétiques et d'autres indispensables pour livrer correctement le courrier à destination. Ces réécritures sont :

- Compléter une adresse partielle. Par exemple, transformer "utilisateur" en "utilisateur@exemple.com" ou "utilisateur@machine" en "utilisateur@machine.exemple.com".
- Remplacer une adresse par une adresse équivalente. Par exemple, remplacer "utilisateur@exemple.com" par "prenom.nom@exemple.com" lors de l'envoi de courrier et faire la traduction inverse à la réception.
- Remplacer une adresse interne par une adresse externe. Par exemple, remplacer "utilisateur@localdomain.local" par "compte-fai@mon.fai" lorsqu'on envoie un courrier depuis une machine personnelle vers Internet.
- Remplacer une adresse par plusieurs. Par exemple, remplacer l'adresse d'un alias par les adresses listées sous cet alias.
- Déterminer quand et où livrer un message pour une adresse spécifique. Par exemple, livrer le courrier à destination de "utilisateur@exemple.com" avec l'agent de livraison [smtp\(8\)](#), vers les machines enregistrées comme serveur de courrier dans le DNS pour le domaine "exemple.com".

Bien que Postfix n'est pas de langage de réécriture d'adresses, il peut surprendre par la puissance de son système de table de correspondance. Typiquement, Postfix utilise ces tables pour faire correspondre les adresses une à une ou groupe par groupe et peut en particulier utiliser des expressions rationnelles pour établir la correspondance d'adresses vers une ou plusieurs. Ces tables prédéfinies peut prendre la forme d'un fichier local ou d'une base de données NIS, LDAP ou SQL. La page [DATABASE_README](#) présente les tables de correspondances Postfix.

Sujets couverts par ce document :

- [Réécrire ou non, ou labelliser comme invalide](#)
- [Introduction à la traduction d'adresse Postfix](#)
- [Réécriture à la réception](#)
 - ◆ [Réécriture à la forme standard](#)
 - ◆ [Correspondances canoniques](#)
 - ◆ [Masquage d'adresse](#)
 - ◆ [Ajout automatique d'un destinataire caché \(BCC\)](#)
 - ◆ [Alias virtuels](#)
- [Réécriture à l'émission](#)
 - ◆ [Résolution d'adresse de destination](#)
 - ◆ [Routage du courrier](#)
 - ◆ [Table des utilisateurs déplacés](#)
- [Réécriture pour une livraison extérieure](#)

- ♦ Remplacement générique pour le courrier SMTP sortant
- Réécriture pour la livraison locale
 - ♦ Base de données locale d'alias
 - ♦ Fichiers utilisateurs .forward
 - ♦ Récupération locale de toutes les adresses (courrier perdu)
- Deboguer vos manipulations d'adresses

Réécrire ou non, ou labelliser comme invalide

Les versions 2.1 et antérieures de Postfix réécrivent toujours les adresses dans les en-têtes de message et ajoutent le domaine propriété de Postfix aux adresses incomplètes. Alors que la réécriture des en-têtes de message est OK pour les messages d'origine locale, elle n'est pas souhaitable pour les messages extérieurs :

- La modification des en-têtes est mal vue des standards,
- Ajouter le domaine propriété de Postfix produit des résultats incorrects pour les adresses extérieures incomplètes,
- Ajouter le domaine propriété de Postfix donne parfois l'apparence qu'un spam est issu d'utilisateurs locaux.

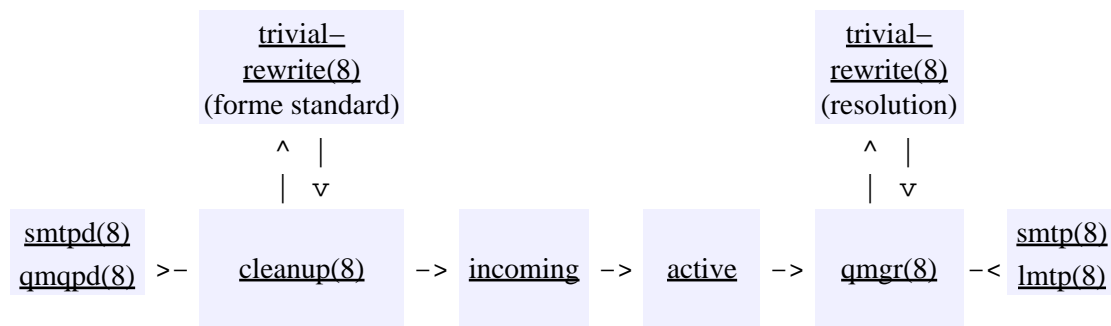
La version 2.2 de Postfix vous donne la possibilité de choisir de réécrire ou non les en-têtes de message des clients SMTP distants, ou de labelliser les adresses incomplètes de tels en-têtes comme invalides.

Fonctionnement :

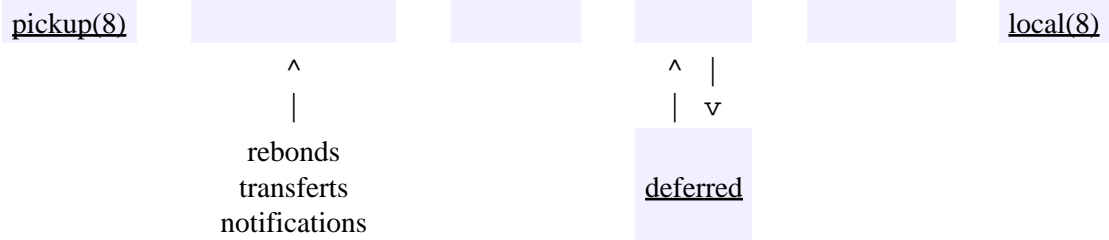
- Postfix examine systématiquement les en-têtes des messages provenant des clients SMTP locaux ou de la commande sendmail, et ajoute son propre domaine aux adresses incomplètes. Le paramètre local_header_rewrite_clients contrôle les clients SMTP que Postfix doit considérer comme locaux (par défaut, seulement les adresses de réseau des interfaces).
- Postfix ne réécrit pas les en-têtes de message des clients SMTP distants lorsque la valeur du paramètre remote_header_rewrite_domain est vide (valeur par défaut).
- Dans les autres cas, Postfix ajoute la valeur remote_header_rewrite_domain aux adresses incomplètes dans les en-têtes de message des clients SMTP distants. Cette fonctionnalité peut être utilisée pour ajouter un domaine réservé tel "domaine invalide", ainsi les adresses incomplètes ne peuvent être confondues avec des adresses locales.

Introduction à la traduction d'adresse Postfix

Le schéma ci-dessous montre les éléments de Postfix intervenant plus particulièrement dans les traductions d'adresses. Voyez la page OVERVIEW pour avoir un aperçu de l'architecture complète de Postfix. Les noms suivis d'un numéro sont les programmes démons de Postfix et les autres désignent les files d'attente ou les sources internes des messages.



Documentation de Postfix en français



Le tableau ci-dessous résume toutes les manipulations d'adresses opérées par Postfix. Si vous lisez ce document pour la première fois, rendez-vous au paragraphe Address rewriting when mail est received Après l'avoir lu, ce tableau vous aidera à trouver rapidement ce que vous cherchez.

Manipulations d'adresses	Portée	Démon	Paramètre d'activation global	Paramètre de
<u>Réécrire les adresses à la forme standard</u>	tout le courrier	<u>trivial-rewrite(8)</u>	<u>append at myorigin,</u> <u>append dot mydomain,</u> <u>swap bangpath,</u> <u>allow percent hack</u>	<u>local header re</u> <u>remote header</u>
<u>Correspondance canonique</u>	tout le courrier	<u>cleanup(8)</u>	<u>canonical maps</u>	<u>receive override</u> <u>local header re</u> <u>remote header</u>
<u>Masquage d'adresse</u>	tout le courrier	<u>cleanup(8)</u>	<u>masquerade domains</u>	<u>receive override</u> <u>local header re</u> <u>remote header</u>
<u>Ajout automatique d'un destinataire caché (BCC)</u>	nouveau message	<u>cleanup(8)</u>	<u>always bcc, sender bcc maps,</u> <u>recipient bcc maps</u>	<u>receive override</u>
<u>Alias virtuel</u>	tout le courrier	<u>cleanup(8)</u>	<u>virtual alias maps</u>	<u>receive override</u>
<u>Résolution d'une adresse de destination</u>	tout le courrier	<u>trivial-rewrite(8)</u>	aucun	aucun
<u>Routage du courrier</u>	tout le courrier	<u>trivial-rewrite(8)</u>	<u>transport maps</u>	aucun
<u>Table des utilisateurs déplacés</u>	tout le courrier	<u>trivial-rewrite(8)</u>	<u>relocated maps</u>	aucun
<u>Table de correspondance générique</u>	courrier SMTP sortant	<u>smtp(8)</u>	<u>smtp generic maps</u>	aucun
<u>Base locale des alias</u>	Courrier local seulement	<u>local(8)</u>	<u>alias maps</u>	aucun
<u>Fichiers utilisateurs forward</u>	Courrier local seulement	<u>local(8)</u>	<u>forward path</u>	aucun
<u>Récupération locale de toutes les adresses (courrier perdu)</u>	Courrier local seulement	<u>local(8)</u>	<u>user relay</u>	aucun

Réécriture des adresses à la réception

Le serveur cleanup(8) reçoit le courrier provenant de l'extérieur et de Postfix lui-même tels les courriers à transférer, le courrier non-livrable renvoyé à l'expéditeur et les notifications à postmaster.

Le serveur cleanup(8) modifie l'expéditeur, le destinataire et le contenu du message dans la forme standard avant de l'envoyer dans la file d'attente incoming. Il modifie l'expéditeur et le destinataire dans les en-têtes et l'enveloppe du message, ajoute les en-têtes manquants tels From: ou Date: qui sont impératifs dans les courriers standards et supprime les en-têtes de message tels Bcc: qui doivent disparaître. Le serveur cleanup(8) délègue les opérations de traduction les plus complexes au serveur trivial-rewrite(8) tel que décrit plus bas.

Les manipulations d'adresses à ce niveau sont :

- Réécriture à la forme standard
- Correspondances canoniques
- Masquage d'adresses
- Ajout automatique d'un destinataire caché (Bcc)
- Alias virtuels

Réécriture à la forme standard

Avant que le démon cleanup(8) ne remplace une adresse en utilisant une table de correspondance, il réécrit les adresses à la forme standard "utilisateur@nom.de.domaine.qualifié", en les envoyant au démon trivial-rewrite(8). Le but ici est de réduire le nombre d'entrées dans les tables de correspondances

Le démon trivial-rewrite(8) de Postfix implémente nativement les manipulations d'adresses suivantes :

Transforme "@machineA,@machineB:utilisateur@site" en "utilisateur@site"

La forme ci-dessus s'appelle une adresse de route et indique que le courrier à destination de "utilisateur@site" doit être délivré via les machines A et B. Son utilisation est totalement obsolète. Postfix ne sait pas traiter les adresses de route autrement qu'en ne retenant que l'adresse finale.

NOTE : les versions 2.2 et supérieures de Postfix ne réécrivent les en-têtes de message des clients SMTP distants que si le client correspond au paramètre local_header_rewrite_clients, ou si le paramètre remote_header_rewrite_domain contient une valeur non vide. Pour revenir au comportement des versions antérieures à la 2.2, indiquez "local_header_rewrite_clients = static:all".

Transforme "site!utilisateur" en "utilisateur@site"

Cette fonctionnalité est contrôlée par le paramètre booléen swap_bangpath (défaut: yes). Son but est de transformer les adresses UUCP en adresses classiques. C'est utile uniquement si vous recevez du courrier via UUCP.

NOTE : les versions 2.2 et supérieures de Postfix ne réécrivent les en-têtes de message des clients SMTP distants que si le client correspond au paramètre local_header_rewrite_clients, ou si le paramètre remote_header_rewrite_domain contient une valeur non vide. Pour revenir au comportement des versions antérieures à la 2.2, indiquez "local_header_rewrite_clients = static:all".

Transforme "utilisateur%domain" en "user@domain"

Cette fonctionnalité est contrôlée par le paramètre booléen allow_percent_hack (défaut: yes). Typiquement, c'est utilisé pour traiter les monstruosités du style "utilisateur%domaine@autre.domaine".

NOTE : les versions 2.2 et supérieures de Postfix ne réécrivent les en-têtes de

message des clients SMTP distants que si le client correspond au paramètre local_header_rewrite_clients, ou si le paramètre remote_header_rewrite_domain contient une valeur non vide. Pour revenir au comportement des versions antérieures à la 2.2, indiquez "local_header_rewrite_clients = static:all".

Transforme "utilisateur" en "utilisateur@\$myorigin"

Cette fonctionnalité est contrôlée par le paramètre booléen append_at_myorigin (défaut: yes). Il est préférable de ne jamais la désactiver car beaucoup de composants de Postfix nécessitent d'avoir des adresses sous la forme "utilisateur@domaine".

NOTE : les versions 2.2 et supérieures de Postfix ne réécrivent les en-têtes de message des clients SMTP distants que si le client correspond au paramètre local_header_rewrite_clients, ou si le paramètre remote_header_rewrite_domain contient une valeur non vide. Pour revenir au comportement des versions antérieures à la 2.2, indiquez "local_header_rewrite_clients = static:all".

Si votre machine n'est pas le serveur de courrier principal du domaine \$myorigin et que vous souhaitez avoir quelques utilisateurs livrés localement sans passer par le serveur principal, créez une entrée la table des alias virtuels qui redirige "utilisateur@\$myorigin" vers "utilisateur@\$myhostname". Voyez aussi le paragraphe "livrer certains utilisateurs localement" de la page STANDARD CONFIGURATION README.

Transforme "utilisateur@machine" en "user@machine.\$mydomain"

Cette fonctionnalité est contrôlée par le paramètre booléen append_dot_mydomain (défaut: yes). Le but ici est d'obtenir un traitement homogène des différentes formes du nom de la même machine.

NOTE : les versions 2.2 et supérieures de Postfix ne réécrivent les en-têtes de message des clients SMTP distants que si le client correspond au paramètre local_header_rewrite_clients, ou si le paramètre remote_header_rewrite_domain contient une valeur non vide. Pour revenir au comportement des versions antérieures à la 2.2, indiquez "local_header_rewrite_clients = static:all".

Certains jugeront incorrect de réécrire "machine" en "machine.domain". C'est pourquoi cette fonctionnalité est désactivable. D'autres trouveront convenable d'ajouter automatiquement le nom de domaine propriété de Postfix.

Transforme "utilisateur@site." en "utilisateur@site" (sans point à la fin).

Le suffixe '.' est supprimé. Toutefois, une adresse se terminant par plusieurs points sera considérée invalide et rejetée.

NOTE : les versions 2.2 et supérieures de Postfix ne réécrivent les en-têtes de message des clients SMTP distants que si le client correspond au paramètre local_header_rewrite_clients, ou si le paramètre remote_header_rewrite_domain contient une valeur non vide. Pour revenir au comportement des versions antérieures à la 2.2, indiquez "local_header_rewrite_clients = static:all".

Correspondances canoniques

Le démon cleanup(8) utilise la table canonical(5) pour réécrire toutes les adresses dans l'enveloppe et les en-têtes du message. Par défaut, toutes les adresses des en-têtes et de l'enveloppe sont réécrites ; ceci est contrôlé par le paramètre de configuration canonical_classes.

Documentation de Postfix en français

NOTE : les versions 2.2 et supérieures de Postfix ne réécrivent les en-têtes de message des clients SMTP distants que si le client correspond au paramètre local_header_rewrite_clients, ou si le paramètre remote_header_rewrite_domain contient une valeur non vide. Pour revenir au comportement des versions antérieures à la 2.2, indiquez "local_header_rewrite_clients = static:all".

Ceci est effectué pour les adresses locales et distantes. Ces correspondances sont pratiques pour remplacer les noms de login en adresses sous la forme "prénom.nom" ou pour nettoyer les adresses invalides produites par les systèmes de messagerie.

Cette fonctionnalité est désactivée par défaut. Pour l'activer, renseignez le paramètre canonical_maps dans le fichier main.cf et indiquez une ou plusieurs tables de correspondances séparées par des espaces ou des virgules.

Exemple :

```
/etc/postfix/main.cf :
    canonical_maps = hash:/etc/postfix/canonical

/etc/postfix/canonical:
    wietse          Wietse.Venema
```

Pour les correspondances statiques comme montré ci-dessus, les tables de correspondances de type hash:, ldap:, mysql: or pgsql: suffisent. Pour les correspondances dynamiques, vous pouvez utiliser les tables d'expressions rationnelles. Ceci suppose que vous soyez familiarisé avec les concepts exposés dans les pages regexp_table(5), pcre_table(5) et canonical(5).

En supplément des correspondances canoniques qui sont appliquées à l'expéditeur et au destinataire, vous pouvez spécifier séparément les correspondances à appliquer aux adresses d'expédition et de destination.

Exemple :

```
/etc/postfix/main.cf :
    sender_canonical_maps = hash:/etc/postfix/sender_canonical
    recipient_canonical_maps = hash:/etc/postfix/recipient_canonical
```

Ces correspondances sont appliquées avant les correspondances communes. Les paramètres sender_canonical_classes et recipient_canonical_classes contrôlent quelles adresses sont sujettes aux correspondances sender_canonical_maps et recipient_canonical_maps, respectivement.

Les correspondances spécifiques aux adresses d'expédition sont pratiques pour réécrire les adresses incorrectes en gardant la possibilité de les utiliser sans créer de boucle de message.

Les correspondances canoniques peuvent être désactivées sélectivement pour le courrier reçu par smtpd(8), qmqpd(8), ou pickup(8), en écrasant les paramètres du fichier main.cf dans le fichier master.cf. Cette fonctionnalité est disponible à partir de la version 2.1 de Postfix.

Exemple :

```
/etc/postfix/master.cf :
:10026      inet      n      -      n      -      -      smtpd
-o receive_override_options=no_address_mapping
```

Note : n'insérez pas d'espace après le "=" ici.

Masquage d'adresses

Le masquage d'adresses est une méthode pour cacher les machines à l'intérieur d'un domaine derrière sa passerelle de messagerie et donner l'impression que le courrier provient de la machine elle-même au lieu des machines internes.

NOTE : les versions 2.2 et supérieures de Postfix ne réécrivent les en-têtes de message des clients SMTP distants que si le client correspond au paramètre local_header_rewrite_clients, ou si le paramètre remote_header_rewrite_domain contient une valeur non vide. Pour revenir au comportement des versions antérieures à la 2.2, indiquez "local_header_rewrite_clients = static:all".

Cette fonctionnalité est désactivée par défaut et est implémentée dans le serveur cleanup(8). Pour l'activer, renseignez le paramètre masquerade_domains du fichier main.cf et indiquer un ou plusieurs noms de domaines séparés par des espaces ou des virgules. Lorsque Postfix tente de masquer un domaine, il parcourt la liste de gauche à droite et utilise la première expression qui correspond.

Exemple :

```
/etc/postfix/main.cf :
masquerade_domains = foo.exemple.com exemple.com
```

transforme "n'importe.quoi.foo.exemple.com" en "foo.exemple.com", et "quelque.chose.d'autre.exemple.com" en "exemple.com".

Un nom de domaine préfixé par "!" signifie qu'il ne faut pas masquer ce domaine ou ses sous-domaines :

```
/etc/postfix/main.cf :
masquerade_domains = !foo.exemple.com exemple.com
```

ne transforme pas "n'importe.quoi.foo.exemple.com", mais transforme "quelque.chose.d'autre.exemple.com" en "exemple.com".

Le paramètre de configuration masquerade_exceptions indique quels noms d'utilisateurs ne doivent pas être soumis aux règles de masquage. Mentionnez un ou plusieurs noms d'utilisateur séparés par des espaces ou des virgules.

Exemple :

```
/etc/postfix/main.cf :
masquerade_exceptions = root
```

Par défaut, Postfix ne fait pas d'exceptions.

Point subtil : par défaut, le masquage d'adresse n'est appliqué qu'aux en-têtes de message et à l'adresse d'expédition de l'enveloppe, mais pas aux adresses de destination de l'enveloppe. Ceci vous permet d'utiliser le masquage d'adresse sur une machine passerelle en gardant la possibilité d'envoyer de l'extérieur vers un utilisateur sur une machine particulière.

Pour masquer également les adresses de destination de l'enveloppe, indiquez (version 1.1 et supérieures) :

```
/etc/postfix/main.cf :
masquerade_classes = envelope_sender, envelope_recipient,
```

`header_sender, header_recipient`

Si vous réécrivez ainsi les adresses de destination de l'enveloppe, Postfix ne saura plus expédier le courrier vers les machines internes.

Le masquage d'adresse peut être désactivé sélectivement pour le courrier reçu par smtpd(8), qmqpd(8), ou pickup(8), en écrasant les paramètres du fichier main.cf dans le fichier master.cf. Cette fonctionnalité est disponible sur les versions 2.1 et supérieures de Postfix.

Exemple :

```
/etc/postfix/master.cf :  
:10026      inet  n      -      n      -      -      smtpd  
-o receive_override_options=no_address_mapping
```

Note : n'insérez pas d'espaces après le signe "=" ici.

Ajout automatique d'un destinataire caché (BCC)

Après avoir appliqué les correspondances canoniques et le masquage d'adresse, le démon cleanup(8) peut générer un champ BCC (blind carbon-copy) pour insérer un destinataire caché. Postfix fournit trois mécanismes :

always bcc = *adresse*

Délivre une copie de tous les messages à l'adresse indiquée. Dans les versions antérieures à Postfix 2.1, cette fonctionnalité est implémentée dans smtpd(8), qmqpd(8) ou pickup(8).

sender bcc maps = *type:table*

Recherche un destinataire BCC dans la table de correspondance "type:table" en utilisant l'adresse d'expédition de l'enveloppe. Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

recipient bcc maps = *type:table*

Recherche un destinataire BCC dans la table de correspondance "type:table" en utilisant l'adresse de destination de l'enveloppe. Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

Note : les destinataires BCC sont seulement générées pour les nouveaux messages. Pour éviter les boucles de message, ils ne sont pas non plus générés pour les messages générés que Postfix transfère en interne ni pour les messages qu'il génère.

L'ajout automatique d'un destinataire caché (y compris *always bcc*) peut être désactivé sélectivement pour le courrier reçu par smtpd(8), qmqpd(8) ou pickup(8), en écrasant les paramètres du fichier main.cf dans le fichier master.cf. Cette fonctionnalité est disponible sur les versions 2.1 et supérieures de Postfix.

Exemple :

```
/etc/postfix/master.cf :  
:10026      inet  n      -      n      -      -      smtpd  
-o receive_override_options=no_address_mapping
```

Note : n'insérez pas d'espaces après le signe "=" ici.

Alias virtuels

Avant d'écrire les destinataires dans la file d'attente, le démon [cleanup\(8\)](#) utilise la table d'alias optionnelle [virtual\(5\)](#) pour rediriger le courrier. Les correspondances n'affectent que l'adresse de destination de l'enveloppe et n'a aucun effet sur l'adresse d'expédition ni sur les adresses des en-têtes. Ces correspondances par alias virtuels sont pratiques pour rediriger le courrier des [alias de domaines virtuels](#) vers les boîtes-aux-lettres d'utilisateurs réels et pour rediriger le courrier des domaines disparus. Les alias virtuels peuvent également être utilisés pour transformer les "prénom.nom" en noms de login UNIX bien que les [alias](#) locaux semblent plus appropriés. Voyez la page [VIRTUAL README](#) pour un aperçu des méthodes d'hébergement de domaines virtuels avec Postfix.

Les alias virtuels sont désactivés par défaut. Pour activer cette fonctionnalité, renseignez le paramètre [virtual_alias_maps](#) dans le fichier [main.cf](#) en indiquant une ou plusieurs tables de correspondances, séparées par des espaces ou des virgules

Exemple :

```
/etc/postfix/main.cf :
    virtual\_alias\_maps = hash:/etc/postfix/virtual

/etc/postfix/virtual:
    Wietse.Venema      wietse
```

Les adresses trouvées sont soumises à une autre itération pour rechercher un alias virtuel, mais elles ne sont pas soumises aux correspondances canoniques pour éviter les boucles de message.

Pour les correspondances statiques comme présentées ci-dessus, les tables de correspondances de type [hash:](#), [ldap:](#), [mysql:](#) ou [pgsql:](#) sont suffisantes. Pour les correspondances dynamiques, vous pouvez utiliser les tables d'expressions rationnelles. Ceci suppose que vous soyez familiarisé avec les idées exprimées dans les pages [regexp_table\(5\)](#), [pcre_table\(5\)](#) et [virtual\(5\)](#).

Note : les copies cachées automatiques ne sont produites que pour les nouveaux messages. Pour éviter les boucles de messages, elles ne sont pas générées pour les messages transférés depuis l'intérieur et pour ceux générés par Postfix lui-même.

Les correspondances par alias virtuels peuvent être désactivés séparément pour les courriers reçus par [smtpd\(8\)](#), [qmqpd\(8\)](#), ou [pickup\(8\)](#), en écrasant les paramètres du fichier [main.cf](#) dans le fichier [master.cf](#). Cette fonctionnalité est disponible à partir de la version 2.1 de Postfix.

Exemple :

```
/etc/postfix/master.cf :
:10026      inet      n      -      n      -      -      smtpd
-o receive\_override\_options=no_address_mapping
```

Note : n'insérez pas d'espaces après le signe "=" ici.

A partir d'ici, le message est prêt à être stocké dans la [file d'attente incoming](#) de Postfix.

Réécriture des adresses à l'émission

Le gestionnaire des files d'attente de Postfix trie le courrier suivant sa destination et le donne à l'agent de livraison local(8), smtp(8), ou lmtp(8). Comme le serveur cleanup(8), il délègue les manipulations d'adresse les plus complexes au serveur trivial-rewrite(8).

Les manipulations d'adresse à ce niveau sont :

- Résolution d'adresse de destination
- Routage du courrier
- Table des utilisateurs déplacés

Chaque agent de livraison de Postfix tente de livrer le courrier à destination en encapsulant l'expéditeur, le destinataire et le contenu message suivant les règles du protocole SMTP, LMTP, etc. Lorsqu'un message ne peut être livré, il est retourné à l'expéditeur ou déplacé dans la file d'attente retardée et testé plus tard.

Manipulations d'adresses lorsque le courrier est livré via l'agent de livraison smtp(8) :

- Remplacement générique pour le courrier SMTP sortant

Manipulations d'adresse lorsque le courrier est délivré via l'agent de livraison local(8) :

- Base de données locale d'alias
- Fichiers utilisateurs .forward
- Récupération locale de toutes les adresses (courrier perdu)

La suite de ce document présente chaque étape de la manipulation en détail avec des exemples spécifiques qui pointent vers d'autres pages avec exemples.

Résolution d'adresse de destination

Le gestionnaire de file d'attente qmgr(8) de Postfix prélève les nouveaux messages depuis la file d'attente d'arrivée (incoming) et les messages anciens depuis la file d'attente retardée (deferred), et demande au démon de réécriture et résolution des adresses trivial-rewrite(8) où il doit être livré.

Depuis la version 2.0, Postfix distingue quatre classes majeures d'adresses. Chaque classe a sa propre liste de noms de domaine et sa propre méthode de livraison par défaut comme indiqué dans le tableau ci-dessous. Reportez-vous à la page ADDRESS CLASS README pour plus de détail. Avant la version 2.0, Postfix ne distingue que le courrier local du reste.

Liste de domaines de destination	Méthode de livraison par défaut	Disponibilité
<u>\$mydestination</u> , <u>\$inet_interfaces</u> , <u>\$proxy_interfaces</u>	<u>\$local_transport</u>	Postfix 1.0
<u>\$virtual_mailbox_domains</u>	<u>\$virtual_transport</u>	Postfix 2.0
<u>\$relay_domains</u>	<u>\$relay_transport</u>	Postfix 2.0
none	<u>\$défaut_transport</u>	Postfix 1.0

Routage du courrier

Une fois que le démon trivial-rewrite(8) a déterminé une méthode de livraison par défaut, il recherche la table optionnelle transport(5) pour obtenir des informations qui surchargent la destination du message et/ou la méthode de livraison. Typiquement, on utilise la table transport(5) pour envoyer du courrier à un système non connecté à Internet, ou pour utiliser une configuration de client SMTP particulière pour des destinations ayant des conditions spéciales. Reportez-vous aux pages STANDARD CONFIGURATION README et UUCP README et aux exemples de la page transport(5).

Les tables de correspondance de routage sont désactivées par défaut. Pour les activer, renseignez le paramètre transport_maps du fichier main.cf en indiquant une ou plusieurs tables de correspondance séparées par des espaces ou des virgules.

Exemple :

```
/etc/postfix/main.cf :  
    transport_maps = hash:/etc/postfix/transport
```

Table des utilisateurs déplacés

Ensuite, le démon de réécriture et de résolution des adresses trivial-rewrite(8) examine chaque destinataire au regard de la base de données relocated(5). Cette table fournit les informations pour joindre les utilisateurs qui n'ont plus de compte ou pour obtenir la conduite à tenir pour les domaines qui n'existent plus. Lorsqu'un message est envoyé à une adresse listée dans cette table, le message est retourné à son expéditeur avec un message d'explication.

La base de données relocated(5) est recherchée après la table de correspondance transport(5) au cas où la table transport(5) remplacerait un destinataire.

L'examen des utilisateurs déplacés est désactivé par défaut. Pour l'activer, renseignez le paramètre relocated_maps du fichier main.cf et indiquez une ou plusieurs tables de correspondance séparées par des espaces ou des virgules.

Exemple :

```
/etc/postfix/main.cf :  
    relocated_maps = hash:/etc/postfix/relocated  
  
/etc/postfix/relocated:  
    utilisateur@exemple.com      autre-utilisateur@ailleurs.com
```

A partir de la version 2 de Postfix, le courrier pour les utilisateurs déplacés sont rejetés par le serveur SMTP pour le motif "user has moved to autreutilisateur@ailleurs.com". Les versions plus anciennes reçoivent d'abord le message puis le retournent à l'expéditeur comme non-livrable avec le même motif.

Remplacement générique pour le courrier SMTP sortant

Certaines machines n'ont pas de nom de domaine valide sur Internet, et utilisent à la place un nom tel *localdomain.local*. Ceci peut poser problème lorsque vous envoyez un message sur Internet car beaucoup de serveurs rejettent les adresses de messagerie sans nom de domaine valide.

Avec le paramètre smtp_generic_maps, vous pouvez indiquer des tables de correspondances generiques qui remplacent les adresses de messagerie locales par des adresses Internet valides lorsque le courrier quitte la machine via SMTP. Les correspondances generiques remplacent les adresses de l'enveloppe et des en-têtes et n'agissent pas récursivement. Elles n'agissent pas lorsque vous envoyez du courrier entre adresses de la machine locale.

Cette fonctionnalité est disponible sur les versions 2.2 et supérieures de Postfix.

Exemple:

```
/etc/postfix/main.cf:
  smtp_generic_maps = hash:/etc/postfix/generic

/etc/postfix/generic:
  user1@localdomain.local      comptel@mon.fai
  user2@localdomain.local      compte2@mon.autre.fai
  @localdomain.local           compte3+local@mon.fai
```

Lorsque le courrier est envoyé à l'extérieur via SMTP, ceci remplace *user1@localdomain.local* par *comptel@mon.fai*, *user2@localdomain.local* par *compte2@mon.autre.fai*, et les autres adresses locales par *compte3@mon.fai* avec une extension d'adresse *+local* (cet exemple suppose que le FAI supporte les extensions d'adresse du type "+").

Base de données locale d'alias

Lorsque le courrier est livré localement, l'agent de livraison local(8) examine chaque destinataire au travers de la base de donnée "aliases(5)". Les traductions n'affectent que les adresses dans les en-têtes. Ces alias locaux sont typiquement employés pour implémenter les listes de distributions ou pour rediriger le courrier des alias standards tels postmaster vers de réels utilisateurs. Cette table peut également servir à remplacer les "prénom.nom" par des noms de login.

L'utilisation des alias locaux est activé par défaut. La configuration par défaut dépend de l'environnement de votre système d'exploitation mais est généralement l'un des suivants :

```
/etc/postfix/main.cf :
  alias_maps = hash:/etc/aliases
  alias_maps = dbm:/etc/aliases, nis:mail.aliases
```

Le chemin de la base de données des alias est contrôlé par le paramètre de configuration alias_database. Cette valeur dépend du système mais est généralement l'un des suivants :

```
/etc/postfix/main.cf :
  alias_database = hash:/etc/aliases (4.4BSD, LINUX)
  alias_database = dbm:/etc/aliases (4.3BSD, SYSV<4)
  alias_database = dbm:/etc/mail/aliases (SYSV4)
```

Un fichier aliases(5) peut indiquer qu'un message doit être livré dans un fichier local ou passé à une commande via son entrée standard. Pour des raisons de sécurité, la livraison vers des fichiers et des commandes est exécutée avec les droits du propriétaire de la base de données des alias Le userid par défaut privs est utilisé pour livrer aux commandes ou fichiers aux alias appartenant à root.

Local per-utilisateur fichiers .forward

Avec la livraison via l'agent [local\(8\)](#), les utilisateurs peuvent contrôler cette livraison en spécifiant des destinations dans le fichier .forward de leur répertoire par défaut (home directories). La syntaxe de ces fichiers est la même que celle des [alias\(5\)](#) locaux oté de la partie gauche des alias (clef de correspondance et colonne).

Récupération locale de toutes les adresses

Lorsque l'agent [local\(8\)](#) de livraison découvre qu'un destinataire n'existe pas, le message est normalement retourné à l'expéditeur avec la mention "user unknown". Il est parfois souhaitable de transférer le courrier des adresses inexistantes vers une autre machine. Dans ce but, vous pouvez indiquer une destination avec le paramètre de configuration [luser_relay](#).

Autre possibilité, le courrier des adresses inexistantes peut être délégué à un autre transporteur de courrier indiqué dans le paramètre de configuration [fallback_transport](#). Pour plus de détails, reportez-vous à la page de manuel de l'agent local de livraison [local\(8\)](#).

Note : si vous utilisez le paramètre [luser_relay](#) pour recevoir le courrier ne correspondant pas à un compte UNIX, vous devez alors indiquer :

```
/etc/postfix/main.cf :  
    local\_recipient\_maps =
```

(c'est à dire vide) dans le fichier [main.cf](#) , sinon le serveur SMTP de Postfix rejettera le courrier ne correspondant pas à un compte UNIX avec la mention "User unknown in local destinataire table". Reportez-vous à la page [LOCAL_RECIPIENT_README](#) pour plus d'informations.

[luser_relay](#) peut indiquer une adresse. Il est soumis aux expansions de "\$name". Exemples :

\$user@autre.machine

L'utilisateur nu, sans extension d'adresse, est préfixée à "@autre.machine". Par exemple, les messages pour "utilisateur+foo" sont envoyés à "utilisateur@autre.machine".

\$local@autre.machine

L'adresse originale de destination, y compris les extensions, est préfixée à "@autre.machine". Par exemple, les messages pour "utilisateur+foo" sont envoyés à "utilisateur+foo@autre.machine".

sysadmin+\$user

L'utilisateur nu, sans extension d'adresse, est ajoutée à "sysadmin". Par exemple, les messages pour "utilisateur+foo" sont envoyés à "sysadmin+utilisateur".

sysadmin+\$local

L'adresse originale de destination, y compris les extensions, est ajoutée à "sysadmin". Par exemple, les messages pour "utilisateur+foo" sont envoyés à "sysadmin+utilisateur+foo".

Deboguer vos manipulations d'adresses

Avec les versions supérieures ou égales à la version 2.1, vous pouvez demander à Postfix de produire des messages de notification de livraison pour le débogage. Ces rapports ne montrent pas seulement les adresses expéditeur/detinataire après réécriture, ils montrent également les informations sur la livraison aux

Documentation de Postfix en français

boîtes-aux-lettres ou aux commandes non-Postfix, les réponses des autres serveurs SMTP, etc.

Postfix peut produire deux types de rapports de livraison de message pour le débogage.

- What-if: rapporte ce qui se passerait, mais ne livre pas le message. Ce mode opératoire est appelé par :

```
$ /usr/sbin/sendmail -bv address...
```

Le rapport du status de la livraison sera envoyé à <votre nom de login>.

- What happened: délivre le courrier et le rapport et/ou les erreurs en incluant les réponses des autres serveurs SMTP. Ce mode opératoire est appelé par :

```
$ /usr/sbin/sendmail -v address...
```

Le rapport du status de la livraison sera envoyé à <votre nom de login>.

Ces rapports contiennent les informations qui sont générées par les agents de livraison de postfix. Comme les processus fonctionnent comme démon et n'interagissent pas directement avec les utilisateurs, le résultat est envoyé par message à l'expéditeur du message de test. Le format de ces rapports est pratiquement identique à ceux des notifications de non-livraison ordinaires.

Par exemple est reproduit ci-dessous le rapport produit par la commande "sendmail -bv postfix-user@postfix.org". La première partie de ce rapport contient le texte lisible. Dans ce cas, le courrier serait livré via mail.cloud9.net avec une réponse "250 Ok". Les autres rapports doivent montrer la livraison à la boîte-aux-lettres ou vers une commande non-Postfix.

```
Content-Description: Notification
Content-Type: text/plain
```

This is the Postfix program at host spike.porcupine.org.

Enclosed is the mail delivery report that you requested.

The Postfix program

```
<postfix-users@postfix.org>: delivery via mail.cloud9.net[168.100.1.4]: 250 Ok
```

La seconde partie du rapport est au format lisible par les machines et inclut les informations suivantes :

- L'adresse de l'expéditeur dans l'enveloppe (wietse@porcupine.org).
- L'adresse de réception dans l'enveloppe (postfix-users@postfix.org). Si l'adresse de destination a été changée par Postfix, il inclut alors l'adresse originale de destination.
- Le status de livraison.

Certains détails sont préliminaires et changeront comme Postfix implémente les standards DSN (notification de status de livraison).

```
Content-Description: Delivery report
Content-Type: message/delivery-status
```

```
Reporting-MTA: dns; spike.porcupine.org
X-Postfix-Queue-ID: 84863BC0E5
X-Postfix-Sender: rfc822; wietse@porcupine.org
Arrival-Date: Tue, 13 Apr 2004 19:27:43 -0400 (EDT)
```

```
Final-Recipient: rfc822; postfix-uses@postfix.org
```

Documentation de Postfix en français

```
Action: deliverable
Status: 2.0.0
Diagnostic-Code: X-Postfix; delivery via mail.cloud9.net[168.100.1.4]: 250 Ok
```

La troisième partie de rapport contient le message que Postfix aurait du livrer incluant les en-têtes From: et To:, afin que vous puissiez voir tous les effets de réécritures d'adresses. Le message soumis à la commande "sendmail -bv" n'a pas de corps, donc aucun n'est montré dans l'exemple ci-dessous.

```
Content-Description: Message
Content-Type: message/rfc822

Received: by spike.porcupine.org (Postfix, from userid 1001)
        id 84863BC0E5; Tue, 13 Apr 2004 19:27:43 -0400 (EDT)
Subject: probe
To: postfix-users@postfix.org
Message-Id: <20040413232743.84863BC0E5@spike.porcupine.org>
Date: Tue, 13 Apr 2004 19:27:43 -0400 (EDT)
From: wietse@porcupine.org (Wietse Venema)
```

Hébergement de sites virtuels avec

Postfix

Objectifs de ce document

Ce document ne concerne que les versions 2.0 ou supérieures de Postfix.

Ce document donne une vue d'ensemble de la façon dont Postfix peut être employé pour accueillir des domaines Internet multiples, dont la livraison finale se fait sur la machine elle-même et/ou pour transférer le courrier vers d'autres destinations.

Il ne décrit pas seulement les mécanismes de livraison de Postfix mais donne des liens pour utiliser des logiciels de livraison étrangers à Postfix.

Les sujets suivants sont abordés :

- Hébergement canonique et hébergement d'autres domaines
- Fichiers locaux et bases de données en réseau
- Aussi simple que possible : domaines partagés, comptes du système UNIX
- Exemple d'ALIAS virtuel de Postfix : domaines séparés, comptes du système UNIX
- Exemple de BOITES-AUX-LETTRES virtuelle : domaines séparés, comptes non-UNIX
- Stockage en boîte-aux-lettres non-Postfix : domaines séparés, comptes non-UNIX
- Domaines de transfert de courrier
- Listes de diffusion
- Répondeur automatique

Hébergement canonique et hébergement d'autres domaines

La plupart des systèmes Postfix sont la **destination finale** de seulement quelques noms de domaine. Ceci inclut le nom de machine, [l'adresse IP] de la machine et parfois le nom du domaine parent. Le reste de ce document se référera à ces domaines comme domaines canoniques. Ils correspondent généralement à la classe d'adresses "domaine local" de Postfix tel que décrit à la page ADDRESS CLASS README.

En plus des domaines canoniques, Postfix peut être configuré pour être la **destination finale** de plusieurs autres domaines. Ces domaines sont appelés "hébergés", car ils ne sont pas directement associés au nom de la machine. Ces domaines hébergés correspondent généralement à la classe d'adresses "domaine virtuel d'alias" de postfix et/ou à la classe d'adresses domaine virtuel de boîtes-aux-lettres comme défini à la page ADDRESS CLASS README.

Mais attendez, il y a mieux!. Postfix peut être configuré pour être le serveur MX de secours d'autres domaines. Dans ce cas, Postfix n'est **pas la destination finale** de ces domaines. Il conserve le courrier lorsque le serveur MX principal ne fonctionne pas, et transfère le courrier dès qu'il fonctionne de nouveau. Cette fonctionnalité est utilisée avec la classe d'adresses domaine relayé comme définie à la page ADDRESS CLASS README.

Finalement, Postfix peut être configuré comme machine de retransmission du courrier à travers Internet. Evidemment, Postfix n'est pas la destination finale de ces messages. Cette fonction est disponible pour les clients et/ou utilisateurs autorisés et est implémentée par la classe d'adresses domaine par défaut définie à la page ADDRESS CLASS README.

Fichiers locaux et bases de données en réseau

L'exemple ci-dessous utilise des tables de correspondances issues de fichiers locaux tels des bases DBM ou Berkeley. Elles sont faciles à déboguer avec la commande **postmap** :

```
Exemple: postmap -q info@exemple.com hash:/etc/postfix/virtual
```

Reportez-vous à la page LDAP README, MYSQL README et PGSQL README pour étudier le remplacement des fichiers locaux par ces bases. Le lecteur est fortement incité à faire fonctionner son système avec des fichiers locaux avant de migrer vers des bases de données réseau et à utiliser la commande **postmap** pour vérifier que ces bases de données produisent exactement le même résultat que les fichiers locaux.

```
Exemple: postmap -q info@exemple.com ldap:/etc/postfix/virtual.cf
```

Aussi simple que possible : domaines partagés, comptes UNIX

La plus simple méthode pour ajouter un domaine supplémentaire est de l'ajouter aux domaines listés dans le paramètre de configuration mydestination et d'ajouter les noms d'utilisateur dans le fichier des mots de passe UNIX.

Cette approche ne fait aucune distinction entre les domaines canoniques et hébergés. Chaque utilisateur reçoit le courrier de tous les domaines.

Dans les exemples suivants, nous utiliserons "exemple.com" comme domaine hébergé par la machine locale.

```
/etc/postfix/main.cf:  
mydestination = $myhostname localhost.$mydomain ... exemple.com
```

Les limites de cette approche sont :

- Un manque total de cloisonnement : le courrier de info@mon.nom.de.machine est délivré au même compte UNIX que celui de info@exemple.com.
- En gérant les utilisateurs au travers du fichier des mots de passe UNIX, l'administration d'un grand nombre d'utilisateurs est difficile.

L'exemple qui suit fournit une solution pour ces deux limites.

Exemple d'ALIAS virtuel Postfix : domaines séparés, comptes du système UNIX

Avec l'approche décrite dans ce paragraphe, chaque domaine hébergé peut avoir ses propres informations, adresses électroniques, etc. Toutefois, il utilise toujours les comptes du système UNIX pour ses livraisons

locales.

Avec les domaines d'alias virtuels, chaque adresse hébergée est un alias d'un compte du système UNIX ou d'une adresse extérieure. L'exemple suivant montre comment utiliser ce mécanisme pour le domaine exemple.com.

```

1 /etc/postfix/main.cf:
2   virtual_alias_domains = exemple.com ...autres domaines hébergés...
3   virtual_alias_maps = hash:/etc/postfix/virtual
4
5 /etc/postfix/virtual:
6   postmaster@exemple.com postmaster
7   info@exemple.com      joe
8   sales@exemple.com     jane
9   # Décommentez l'entrée suivante pour implémenter une adresse de collecte
10  # @exemple.com        jim
11  ...alias virtuel pour d'autres domaines...
```

Notes :

- Ligne 2 : le paramètre virtual_alias_domains indique à Postfix que le domaine exemple.com est un domaine d'alias virtuels. Si vous l'oubliez, Postfix rejettera le courrier (relais interdit) ou ne saura pas le livrer (le courrier pour exemple.com sera renvoyé à la machine elle-même).

Ne listez JAMAIS un domaine d'alias virtuels dans la liste des domaines mydestination!

- Lignes 3–8 : le fichier /etc/postfix/virtual contient les alias virtuels. Avec cet exemple, le courrier de postmaster@exemple.com est livré au postmaster local alors que celui de sales@exemple.com est envoyé au compte UNIX jane. Le courrier de toutes les autres adresses du domaine exemple.com est rejeté avec le message d'erreur "User unknown".
- Ligne 10 : l'entrée commentée (texte après #) montre comment implémenter une adresse de collecte qui reçoit tout le courrier des adresses du domaine exemple.com non listées dans le fichier d'alias virtuels. Ce n'est pas sans risque. Les spammers essaient d'envoyer du courrier semblant venir et à destination de n'importe quel nom possible. Une adresse de collecte est susceptible de recevoir de nombreux messages de spam ou de notification de messages envoyés avec une adresse n–importe–quoi@exemple.com.

Lancez la commande "**postmap /etc/postfix/virtual**" après modification du fichier virtual, puis lancez la commande "**postfix reload**" après avoir modifié le fichier main.cf.

Note : les alias virtuels peuvent correspondre à une adresse locale, à une adresse extérieure ou au deux. Ils ne doivent pas nécessairement correspondre à des comptes du système UNIX de votre machine.

Pour plus de détails sur les fichiers d'alias et en particulier pour les destinataires multiples, reportez-vous à la page de manuel virtual(5).

Les alias virtuels résolvent un problème : ils permettent à chaque domaine d'avoir ses propres adresses de courrier. Mais il en reste un : chaque adresse virtuelle correspond à un compte UNIX. A chaque nouvelle adresse, vous augmentez les comptes du système UNIX.

Exemple de BOITES–AUX–LETTRES virtuelle : domaines séparés, comptes non–UNIX

Documentation de Postfix en français

Lorsqu'un système accumule les domaines et les utilisateurs, il devient moins souhaitable de créer pour chaque utilisateur un compte sur le système UNIX.

Avec l'agent de livraison de courrier virtual(8) de Postfix, chaque adresse de destination peut avoir sa propre boîte-aux-lettres virtuelle. Contrairement aux domaines d'alias virtuels, les domaines de boîtes-aux-lettres virtuelles ne nécessitent pas d'avoir une correspondance pour chaque adresse de destination, et les propriétaires d'une boîte aux lettres n'ont pas besoin de disposer d'un compte du système UNIX.

L'agent de livraison de courrier virtual(8) de Postfix examine le chemin de la boîte-aux-lettre de l'utilisateur, l'uid et le gid via des tables séparées suivant l'adresse de destination. Le répertoire de livraison est déterminé en terminant le chemin de la boîte-aux-lettres par "/".

Si vous trouvez saugrenue l'idée d'utiliser plusieurs tables, souvenez-vous que vous pouvez stocker les informations dans une base SQL. Si vous prenez ce chemin, lisez le paragraphe "fichiers locaux et bases de données" de cette page.

Ci-dessous un exemple d'un domaine virtuel de boîtes-aux-lettres "exemple.com" :

```
1 /etc/postfix/main.cf:
2   virtual_mailbox_domains = exemple.com ...autres domaines...
3   virtual_mailbox_base = /var/mail/vhosts
4   virtual_mailbox_maps = hash:/etc/postfix/vmailbox
5   virtual_minimum_uid = 100
6   virtual_uid_maps = static:5000
7   virtual_gid_maps = static:5000
8   virtual_alias_maps = hash:/etc/postfix/virtual
9
10 /etc/postfix/vmailbox:
11   info@exemple.com      exemple.com/info
12   sales@exemple.com     exemple.com/sales/
13   # Décommentez la ligne ci-dessous pour implémenter une adresse de collecte.
14   # @exemple.com        exemple.com/catchall
15   ...virtual mailboxes for more domains...
16
17 /etc/postfix/virtual:
18   postmaster@exemple.com postmaster
```

Notes:

- Ligne 2 : Le paramètre virtual_mailbox_domains indique à Postfix que exemple.com est un domaine de boîtes-aux-lettres virtuelles. Si vous l'oubliez, Postfix rejettera le courrier (relais interdit) ou ne sera pas en mesure de le livrer (le courrier de exmple.com bouclera).

Ne listez JAMAIS un domaine de boîtes-aux-lettres virtuelles dans les domaines "mydestination"!

N'inscrivez JAMAIS un domaine de boîtes-aux-lettres virtuelles dans la liste des domaines d'ALIAS virtuels!

- Ligne 3 : Le paramètre virtual_mailbox_base indique le répertoire de base pour toutes les boîtes aux lettres virtuelles. C'est un mécanisme évitant les erreurs : le courrier ne peut être livré n'importe où.
- Lines 4, 10–15: Le paramètre virtual_mailbox_maps indique la table des correspondances entre les adresses virtuelles et les boîtes-aux-lettres (ou les répertoires). Dans cet exemple, le courrier de info@exemple.com est envoyé dans le fichier /var/mail/vhosts/exemple.com/info et celui de sales@exemple.com est envoyé dans le répertoire de boîtes /var/mail/vhosts/exemple.com/sales/.

- Ligne 5 : Le paramètre virtual_minimum_uid indique la limite basse de l'UID du propriétaire de la boîte-aux-lettres ou du répertoire. C'est un mécanisme de sécurité évitant les erreurs. Il évite d'écrire sur des fichiers sensibles.
- Lignes 6, 7: Les paramètres virtual_uid_maps et virtual_gid_maps indiquent que les boîtes-aux-lettres virtuelles appartiennent à un UID et un GID fixé à 5000. Si ce n'est pas ce que vous souhaitez, indiquez une table de correspondance entre adresses de destination et uid.
- Ligne 14 : La ligne commentée (texte après #) montre comment implémenter une adresse de collecte. Soyez prêts à recevoir un grand nombre de spam et de notifications concernant du spam correspondant à n-importe-quoi@exemple.com.

N'inscrivez JAMAIS une BOÎTE-AUX-LETTRES de collecte virtuelle dans le fichier des ALIAS virtuels!!

- Lignes 8, 17, 18: Comme vous pouvez le constater, il est possible de mixer les alias virtuels avec des boîtes-aux-lettres virtuelles. Nous utilisons cette fonctionnalité pour rediriger le courrier du postmaster de exemple.com au postmaster local. Vous pouvez utiliser ce même mécanisme pour rediriger une adresse vers une adresse externe.
- Ligne 18 : Cet exemple suppose que \$myorigin est listé dans le paramètre mydestination du fichier main.cf. Dans le cas contraire, indiquez explicitement le domaine dans l'adresse coté droit sinon, le courrier n'ira pas au bon domaine.

Lancez la commande "**postmap /etc/postfix/virtual**" après avoir modifié le fichier virtual, lancez la commande "**postmap /etc/postfix/vmailbox**" après modification du fichier vmmailbox puis lancez "**postfix reload**" après avoir modifié le fichier main.cf.

Note : le courrier livré est ajouté avec les privilèges des UID/GID indiqués par les paramètres virtual_uid_maps et virtual_gid_maps. Les versions 2.0 et supérieures de Postfix ne créent pas les répertoires de boîtes-aux-lettres dans des répertoires positionnés en écriture pour tout le monde vous devez les créer par avance. Postfix peut être en mesure de créer les boîtes-aux-lettres lui-même suivant les permissions d'écriture sur le répertoire parent, mais il est préférable de les créer avant.

Pour plus de détail à propos de l'agent de livraison virtuel, reportez-vous à la page de manuel virtual(8).

Stockage en boîte-aux-lettres non-Postfix : domaines séparés, comptes non-UNIX

il s'agit d'une variante de l'exemple de boîtes-aux-lettres virtuelles. Comme précédemment, chaque domaine hébergé peut avoir ses propres boîtes-aux-lettres.

Comme un logiciel extérieur à Postfix sera utilisé pour la livraison finale, certains concepts de Postfix sont nécessaire pour la bonne cohésion de l'ensemble. Pour plus de détails, reportez-vous au paragraphe sur la classe "domaine de boîtes-aux-lettres virtuelles" de la page ADDRESS CLASS README.

Ce paragraphe décrit le point-de-vue de Postfix. Consultez les pages LMTP README et MAILDROP README pour plus d'information sur Cyrus le rejet du courrier.

Dans l'exemple ci-dessous, le courrier du domaine hébergé exemple.com est envoyé à un agent extérieur à Postfix :

```
1 /etc/postfix/main.cf:
2   virtual_transport = ...voir ci-dessous...
```

Documentation de Postfix en français

```
3  virtual mailbox domains = exemple.com ...autres domaines...
4  virtual mailbox maps = hash:/etc/postfix/vmailbox
5  virtual alias maps = hash:/etc/postfix/virtual
6
7  /etc/postfix/vmailbox:
8      info@exemple.com    whatever
9      sales@exemple.com   whatever
10     # Décommentez la ligne ci-dessous pour implémenter une adresse de collecte.
11     # Configurez le stockage de la boîte-aux-lettres pour accepter toutes les adresses.
12     # @exemple.com      whatever
13     ...boîtes-aux-lettres virtuelles pour d'autres domaines...
14
15 /etc/postfix/virtual:
16     postmaster@exemple.com postmaster
```

Notes :

- Ligne 2 : Avec la livraison avec un gestionnaire de boîtes-aux-lettres non-Postfix pour des domaines hébergés, le paramètre virtual transport indique généralement le client LMTP de Postfix, ou le nom de l'entrée du fichier master.cf qui lance les logiciels extérieurs à Postfix via l'agent de livraison pipe. Quelques exemples (utilisez seulement l'un d'entre eux) :

```
virtual transport = lmtp:unix:/path/name (utilise une socket UNIX)
virtual transport = lmtp:hostname:port   (utilise une socket TCP)
virtual transport = maildrop:              (utilise pipe(8) pour commander)
```

Un exemple de méthode de livraison maildrop est déjà présenté dans le fichier master.cf par défaut. Reportez-vous à la page [MAILDROP README](#) pour plus d'explications.

- Ligne 3 : Le paramètre virtual mailbox domains indique à Postfix que le domaine exemple.com doit être livré en utilisant la méthode définie dans le paramètre virtual transport présenté ci-dessus. Si vous l'oubliez, Postfix rejettera le courrier (relais interdit) ou sera incapable de le livrer (boucle de message).

Ne listez JAMAIS un domaine de boîtes-aux-lettres virtuelles dans la liste des domaines mydestination!

Ne listez JAMAIS un domaine de boîtes-aux-lettres virtuelles dans les domaines d'ALIAS virtuels!

- Lines 4, 7–13: Le paramètre virtual mailbox maps indique une table de correspondance avec toutes les adresses valides. Le résultat de la consultation est ignoré par Postfix. Dans l'exemple ci-dessus, info@exemple.com et sales@exemple.com sont inscrits dans cette liste et le courrier des autres adresses est rejeté avec la mention "User unknown". Si vous souhaitez utiliser LDAP, MySQL ou PostgreSQL au lieu des fichiers locaux, lisez le paragraphe "[fichiers locaux et bases de données](#)" au début de cette page!
- Ligne 12 : La ligne commentée (texte après #) vous montre comment créer une adresse de collecte. Ici aussi, le résultat de la consultation est ignoré par Postfix.

N'inserez JAMAIS l'adresse d'une BOÎTE-AUX-LETTRES virtuelle dans le fichier des ALIAS!!

Note : Si vous indiquez une adresse dans virtual mailbox maps, vous devez quand même configurer le système de livraison non-Postfix pour recevoir le courrier de toutes les adresses de ce domaine.

- Lignes 5, 15, 16 : comme vous l'avez ci-dessus, il est possible de mixer les alias virtuels et les boîtes-aux-lettres virtuelles. Nous utilisons cette fonctionnalité pour rediriger le courrier du postmaster de exemple.com vers le postmaster local. Vous pouvez utiliser le même mécanisme pour rediriger n'importe quelle adresse vers une adresse locale ou extérieure.

- Ligne 16 : Cet exemple suppose que `$myorigin` est listé sous le paramètre `mydestination` dans le fichier `main.cf`. Si ce n'est pas le cas, renseignez explicitement le nom de domaine dans la partie droite de la table des alias virtuels, sinon le courrier n'ira pas au bon domaine.

Lancez la commande "**postmap /etc/postfix/virtual**" après avoir modifié le fichier virtual, lancez "**postmap /etc/postfix/vmailbox**" après avoir modifié le fichier vmmailbox et lancez la commande "**postfix reload**" après avoir modifié le fichier `main.cf`.

Domaines de transfert de courrier

Certains fournisseurs hébergent des domaines qui n'ont pas (ou peu) de boîtes-aux-lettres locales. Le but principal est de transférer le courrier de ces domaines ailleurs. L'exemple ci-dessous montre comment paramétrer le domaine comme un domaine de transfert :

```
1 /etc/postfix/main.cf:
2   virtual_alias_domains = exemple.com ...autre domaines hébergés...
3   virtual_alias_maps = hash:/etc/postfix/virtual
4
5 /etc/postfix/virtual:
6   postmaster@exemple.com postmaster
7   joe@exemple.com        joe@quelque-part
8   jane@exemple.com       jane@quelque-part-ailleurs
9   # Décommentez la ligne ci-dessous pour implémenter une adresse de collecte
10  # @exemple.com          jim@encore-un-autre-site
11  ...alias virtuels pour d'autres domaines...
```

Notes :

- Ligne 2 : Le paramètre `virtual_alias_domains` indique à Postfix que `exemple.com` est un domaine d'alias virtuels. Si vous l'omettez, Postfix rejettera le courrier avec la mention "relay access denied" ou sera incapable de le livrer (boucle de messages).

Ne listez JAMAIS un domaine d'alias virtuels dans la liste des domaines `mydestination`!

- Lignes 3–11 : Le fichier `/etc/postfix/virtual` contient les alias virtuels. Dans cet exemple, le courrier de `postmaster@exemple.com` est envoyé au postmaster local alors que celui de `joe@exemple.com` est envoyé à l'adresse extérieure `joe@quelque-part`, et celui de `jane@exemple.com` vers `jane@quelque-part-ailleurs`. Les messages des autres adresses sont rejetées avec la mention "User unknown".
- Ligne 10 : L'entrée commentée (texte après #) montre comment implémenter une adresse de collecte qui reçoit le courrier de toutes les adresses du domaine `exemple.com` non listées dans le fichier des alias virtuels. Ce n'est pas sans risque. Les spammers essaient d'envoyer du courrier semblant venir et à destination de n'importe quel nom possible. Une adresse de collecte est susceptible de recevoir de nombreux messages de spam ou de notification de messages envoyés avec une adresse `n-importe-quoi@exemple.com`.

Lancez la commande "**postmap /etc/postfix/virtual**" après avoir modifié le fichier virtual et lancez la commande "**postfix reload**" après avoir modifié le fichier `main.cf`.

Pour plus de détails sur les fichiers d'alias virtuels et en particulier pour les adressages multiples, reportez-vous à la page de manuel [virtual\(5\)](#).

Listes de diffusion

Les exemples proposés ci–avant ont montré comment redirigé le courrier du postmaster du domaine virtuel vers le postmaster local. Vous pouvez utiliser la même méthode pour toutes les adresses vers une adresse locale ou extérieure.

Il y a une limitation majeure : les alias et boîtes–aux–lettres virtuelles ne peuvent livrer directement une liste de diffusion comme majordomo. La solution est d'utiliser de créer des alias virtuels redirigeant vers l'agent de livraison local :

```
/etc/postfix/main.cf:
    virtual alias maps = hash:/etc/postfix/virtual

/etc/postfix/virtual:
    listname-request@example.com listname-request
    listname@example.com         listname
    owner-listname@example.com   owner-listname

/etc/aliases:
    listname: "|/chemin/vers/majordomo ... "
    owner-listname: ...
    listname-request: ...
```

Cet exemple suppose que \$myorigin est mentionné dans le paramètre mydestination du fichier main.cf. Si ce n'est pas le cas, renseignez explicitement le nom de domaine dans la partie droite de la table, sinon le courrier n'ira pas au bon domaine.

Pour plus d'information sur l'agent de livraison local, reportez–vous à la page de manuel local(8).

Pourquoi cet exemple emploie–t–il un alias virtuel maladroit au lieu d'utiliser un transport plus élégant ? La principale raison est que le courrier de la liste de diffusion serait rejeté avec la mention "User unknown". Faire fonctionner le transport nécessiterait des entrées statiques dans les tables.

- Dans le cas d'un domaine d'alias virtuels, il faudra une correspondance pour chaque liste de diffusion.
- Dans le cas d'un domaine de boîtes–aux–lettres virtuelles, il faudra une boîte virtuelle pour chaque liste de diffusion.

Répondeur automatique

Pour mettre en uvre un répondeur automatique pour des destinataires virtuels en continuant de livrer le courrier normalement, utilisez une table d'alias virtuels :

```
/etc/postfix/main.cf:
    virtual alias maps = hash:/etc/postfix/virtual

/etc/postfix/virtual:
    user@domain.tld user@domain.tld, user@domain.tld@autoreply.mydomain.tld
```

Ceci livre le courrier au destinataire et envoie une copie du message à l'adresse qui génère des réponses automatiques. Cette adresse peut correspondre à une autre machine ou être traité localement en mettant en uvre une entrée dans la table de transport qui livre via un pipe le courrier à destination de autoreply.mydomain.tld à un script qui répond à l'expéditeur.

Ne listez PAS autoreply.mydomain.tld dans mydestination!

```
/etc/postfix/main.cf:
    transport_maps = hash:/etc/postfix/transport

/etc/postfix/transport:
    autoreply.mydomain.tld  autoreply:

/etc/postfix/master.cf:
# =====
# service type  private unpriv  chroot  wakeup  maxproc command
#               (yes)   (yes)   (yes)   (never) (100)
# =====
autoreply unix  -      n      n      -      -      pipe
               flags= user=nobody argv=/path/to/autoreply $sender $mailbox
```

Ceci appelle /path/to/autoreply avec sur la ligne de commande l'adresse du destinataire et l'adresse du destinataire sous la forme utilisateur@domaine.

Pour plus d'information, reportez-vous à la page de manuel pipe(8) et aux commentaires du fichier master.cf.

Authentification SASL avec Postfix

ATTENTION ATTENTION ATTENTION

Les gens inquiets d'installer Postfix peuvent espérer que Postfix est plus sûr que les autres serveurs de messagerie. La librairie Cyrus SASL est un code très long et en l'utilisant avec le client et le serveur SMTP, Postfix devient aussi sûr que les autres systèmes qui l'utilisent.

Comment Postfix utilise les informations d'authentification SASL

Le support SASL de Postfix ([RFC 2554](#)) peut être utilisé pour authentifier les clients SMTP distants sur le serveur SMTP de Postfix et pour authentifier le client SMTP de Postfix auprès des serveurs SMTP distants.

Lorsqu'un message arrive, Postfix enregistre le nom d'utilisateur fournit, la méthode d'authentification et l'adresse de l'expéditeur dans le fichier de logs et éventuellement autorise l'accès via la restriction anti-spam [permit_sasl_authenticated](#).

Postfix n'enregistre pas les informations d'authentification SASL dans les en-têtes du message et ne les passe pas dans les commandes SMTP lorsqu'il transfère le courrier car cela ne regarde pas les autres serveurs. Ceux qui doivent y accéder les trouverons dans le fichier de logs. Un jour, les en-têtes de message de Postfix pourront être configurés pour enregistrer le nom d'utilisateur SASL sans avoir à éditer le code C.

Ce document couvre les sujets suivants :

- [Quelles implémentations SASL sont supportées](#)
- [Compiler Postfix avec le support SASL Dovecot](#)
- [Compiler les librairies Cyrus-SASL](#)
- [Compiler Postfix avec le support de Cyrus-SASL](#)
- [Activer l'authentification SASL dans le serveur SMTP de Postfix](#)
- [Configuration de Dovecot SASL pour le serveur SMTP de Postfix](#)
- [Configuration de Cyrus SASL pour le serveur SMTP de Postfix](#)
- [Tester l'authentification SASL dans le serveur SMTP de Postfix](#)
- [Dépanner SASL](#)
- [Activer l'authentification SASL dans le client SMTP de Postfix](#)
- [Références](#)

Lors d'un envoi de message, Postfix peut examiner le nom de machine du serveur ou le domaine de destination (partie droite de l'adresse) au regard d'une table et si un couple login/mot-de-passe est trouvé, il l'utilisera pour s'authentifier au serveur.

Quelles implémentations SASL sont supportées

Ce document décrit le fonctionnement de Postfix avec les implémentations SASL suivantes :

- Cyrus SASL version 1 (client et serveur)
- Cyrus SASL version 2 (client et serveur)
- Protocole Dovecot version 1 (serveur uniquement, Postfix versions 2.3 et supérieures)

Postfix version 2.3 introduit un mécanisme de plug-in qui fournit le support de multiples implémentations SASL. Pour connaître les implémentations compilées dans Postfix, utilisez les commandes suivantes :

```
% postconf -a (SASL support in the SMTP server)
% postconf -A (SASL support in the SMTP+LMTP client)
```

Bien entendu, ces commandes ne sont pas disponibles dans les versions antérieures de Postfix.

Building Postfix with Dovecot SASL support

Le support SASL Dovecot est disponible dans Postfix versions 2.3 et supérieures. Le code source de Dovecot est disponible via <http://www.dovecot.org/>. A l'heure de l'écriture de ces lignes, seul la partie serveur du support SASL est disponible, ainsi, vous ne pouvez l'utiliser pour vous authentifier auprès du serveur de votre fournisseur de réseau. Dovecot utilise son propre processus démon pour l'authentification. Ceci maintient simple le processus de compilation de Postfix car il n'y a pas lieu de lui lier d'autres bibliothèques.

Pour générer les nécessaires Makefiles, lancez ce qui suit dans le répertoire racine de Postfix :

```
% make makefiles CCARGS='-DUSE_SASL_AUTH -DDEF_SASL_SERVER=\"dovecot\"'
```

Après cela, lancez "make" comme indiqué à la page [INSTALL](#).

Notes :

- Le paramètre "-DDEF_SASL_SERVER" n'est pas nécessaire ; il rend simplement la configuration de Postfix un peu plus simple car vous n'avez pas à indiquer le type de plug-in SASL utilisé dans le fichier [main.cf](#).
- Si vous voulez supporter LDAP ou TLS, vous devrez intégrer leurs CCARGS et AUXLIBS dans la ligne de commande ci-dessus.

Compiler les bibliothèques Cyrus-SASL

Postfix semble fonctionner avec cyrus-sasl-1.5.5 ou cyrus-sasl-2.1.1, disponibles à l'adresse suivante :

<ftp://ftp.andrew.cmu.edu/pub/cyrus-mail/>.

IMPORTANT : si vous installez les bibliothèques Cyrus SASL par défaut, vous devrez créer le lien symbolique `/usr/lib/sasl -> /usr/local/lib/sasl` pour la version 1.5.5 ou `/usr/lib/sasl2 -> /usr/local/lib/sasl2` pour la version 2.1.1.

Forcément, la version 5 de Microsoft Internet Explorer nécessite une méthode d'authentification SASL non-standard. Pour l'activer, spécifiez ``./configure --enable-login''.

Compiler Postfix avec le support de Cyrus–SASL

Pour compiler Postfix avec le support de l'authentification SASL, ce qui suit suppose que les fichiers include de Cyrus SASL sont situés dans le répertoire /usr/local/include et les bibliothèques dans le répertoire /usr/local/lib.

Sur certains systèmes ceci construit les définitions nécessaires au Makefile :

(pour Cyrus SASL version 1.5.5):

```
% make tidy # si vous utilisez des sources ayant déjà été compilées
% make makefiles CCARGS="-DUSE_SASL_AUTH -DUSE_CYRUS_SASL \
-I/usr/local/include" AUXLIBS="-L/usr/local/lib -lsasl"
```

(pour Cyrus SASL version 2.1.1):

```
% make tidy # si vous utilisez des sources ayant déjà été compilées
% make makefiles CCARGS="-DUSE_SASL_AUTH -DUSE_CYRUS_SASL \
-I/usr/local/include/sasl" AUXLIBS="-L/usr/local/lib -lsasl2"
```

Sur Solaris 2.x vous devez indiquer les informations pour lier les bibliothèques sinon ld.so ne trouvera pas les bibliothèques partagées SASL

(pour Cyrus SASL version 1.5.5):

```
% make tidy # si vous utilisez des sources ayant déjà été compilées
% make makefiles CCARGS="-DUSE_SASL_AUTH -DUSE_CYRUS_SASL \
-I/usr/local/include" AUXLIBS="-L/usr/local/lib \
-R/usr/local/lib -lsasl"
```

(pour Cyrus SASL version 2.1.1):

```
% make tidy # si vous utilisez des sources ayant déjà été compilées
% make makefiles CCARGS="-DUSE_SASL_AUTH -DUSE_CYRUS_SASL \
-I/usr/local/include/sasl" AUXLIBS="-L/usr/local/lib \
-R/usr/local/lib -lsasl2"
```

Activer l'authentification SASL dans le serveur SMTP de Postfix

Pour activer le support SASL dans le serveur SMTP :

```
/etc/postfix/main.cf:
    smtpd_sasl_auth_enable = yes
```

Pour autoriser le relais de courrier des utilisateurs authentifiés :

```
/etc/postfix/main.cf:
    smtpd_recipient_restrictions =
        permit_mynetworks permit_sasl_authenticated ...
```

Pour reporter les noms de login SASL dans les en-têtes de message Received: (Postfix versions 2.3 et supérieures):

```
/etc/postfix/main.cf:
    smtpd_sasl_authenticated_header = yes
```

Note : les noms de login SASL seront ainsi connus du monde entier.

Les anciens clients SMTP Microsoft utilisent une version non standard de la syntaxe du protocole AUTH et s'attendent à ce que le serveur SMTP réponde au EHLO avec "250 AUTH=stuff" au lieu de "250 AUTH stuff". Pour accueillir de tels clients (en complément des clients conformes), utilisez ce qui suit :

```
/etc/postfix/main.cf:  
  broken_sasl_auth_clients = yes
```

Configuration de Dovecot SASL pour le serveur SMTP de Postfix

Le support SASL de Dovecot est disponible sur les versions 2.3 et supérieures de Postfix. Du côté de Postfix, vous devez indiquer l'emplacement de la socket du démon d'authentification Dovecot. Nous utilisons un chemin relatif au répertoire des files d'attente de Postfix, ainsi le mécanisme fonctionne que Postfix fonctionne en cage (chroot) ou non :

```
/etc/postfix/main.cf:  
  smtpd_sasl_type = dovecot  
  smtpd_sasl_path = private/auth
```

Du côté de Dovecot, vous devez également indiquer l'emplacement de la socket du démon d'authentification. Dans ce cas nous indiquons un chemin absolu. Dans l'exemple ci-dessous, nous supposons que le répertoire des files d'attente de Postfix est /var/spool/postfix/.

```
/chemin/vers/dovecot.conf:  
  auth default {  
    mechanisms = plain login  
    passdb pam {  
    }  
    userdb passwd {  
    }  
    socket listen {  
      client {  
        path = /var/spool/postfix/private/auth  
        mode = 0660  
        user = postfix  
        group = postfix  
      }  
    }  
  }
```

Reportez-vous à la documentation de Dovecot pour la configuration et le fonctionnement du serveur d'authentification Dovecot.

Configuration de Cyrus SASL pour le serveur SMTP de Postfix

Vous devez indiquer dans le fichier /usr/local/lib/sasl/smtpd.conf (SASL version 1.5.5) ou /usr/local/lib/sasl2/smtpd.conf (SASL version 2.1.1) comment le serveur doit valider les mots de passe des clients.

Note : certaines distributions de Postfix sont modifiées et cherchent le fichier smtpd.conf dans /etc/postfix.

Note : certaines distributions de Cyrus SASL recherchent le fichier smtpd.conf dans /etc/sasl2.

- Pour effectuer les authentifications au travers du fichier des mots de passe UNIX :

(Cyrus SASL version 1.5.5)

```
/usr/local/lib/sasl/smtpd.conf:  
pwcheck_method: pwcheck
```

(Cyrus SASL version 2.1.1)

```
/usr/local/lib/sasl2/smtpd.conf:  
pwcheck_method: pwcheck
```

Le nom du fichier du répertoire /usr/local/lib/sasl (Cyrus SASL version 1.5.5) ou /usr/local/lib/sasl2 (Cyrus SASL version 2.1.1) utilisé par la librairie SASL pour la configuration peut être indiqué avec :

```
/etc/postfix/main.cf:  
smtpd_sasl_application_name = smtpd
```

Le démon pwcheck est inclus dans les sources de cyrus-sasl.

IMPORTANT : les processus de Postfix doivent avoir les droits lecture+exécution sur le répertoire /var/pwcheck sinon l'authentification risque d'échouer.

- Autre possibilité avec les versions Cyrus SASL 1.5.26 et supérieures (y compris 2.1.1), essayez :

(Cyrus SASL version 1.5.26)

```
/usr/local/lib/sasl/smtpd.conf:  
pwcheck_method: saslauthd
```

(Cyrus SASL version 2.1.1)

```
/usr/local/lib/sasl2/smtpd.conf:  
pwcheck_method: saslauthd
```

Le démon saslauthd est également inclus dans les sources de cyrus-sasl. Il est plus flexible que pwcheck en ce sens qu'il peut effectuer les authentifications en utilisant PAM et diverses autres sources. Pour utiliser PAM, lancez saslauthd avec "-a pam".

- Pour authentifier avec la base propre de Cyrus SASL :

(Cyrus SASL version 1.5.5)

```
/usr/local/lib/sasl/smtpd.conf:  
pwcheck_method: sasldb
```

(Cyrus SASL version 2.1.1)

```
/usr/local/lib/sasl2/smtpd.conf:  
pwcheck_method: auxprop
```

Ceci utilisera le fichier de mots de passe de SASL (défaut: /etc/sasldb dans la version 1.5.5, ou /etc/sasldb2 dans la version 2.1.1), qui est maintenu avec la commande saslpasswd ou saslpasswd2 (parties des logiciels Cyrus SASL). Sur certains systèmes mal supportés, la commande saslpasswd doit être lancée plusieurs fois avant d'arrêter de se plaindre. Le serveur SMTP de Postfix doit avoir accès au fichier sasldb – vous devriez jouer avec les permissions des groupes. Avec le mécanisme d'authentification OTP, le serveur doit également avoir un accès en écriture sur le fichier /etc/sasldb2 or /etc/sasldb (ou sur la base de données SQL si utilisée).

IMPORTANT: Pour que sasldb fonctionne, assurez-vous que le domaine SASL (royaume) corresponde à un nom de domaine pleinement qualifié.

EXEMPLE:

(Cyrus SASL version 1.5.5)

```
% saslpasswd -c -u `postconf -h myhostname` exampleuser
```

(Cyrus SASL version 2.1.1)

```
% saslpasswd2 -c -u `postconf -h myhostname` exampleuser
```

Vous pouvez connaître le royaume des utilisateurs dans `sasldb` avec `sasldblistusers` (Cyrus SASL version 1.5.5) ou `sasldblistusers2` (Cyrus SASL version 2.1.1).

Du côté de Postfix, vous ne pouvez avoir qu'un royaume par instance `smtpd`, et seuls les utilisateurs appartenant à ce royaume pourront s'authentifier. La variable Postfix `smtpd_sasl_local_domain` contrôle le royaume utilisé par `smtpd` :

```
/etc/postfix/main.cf:  
smtpd_sasl_local_domain = $myhostname
```

IMPORTANT : tout les utilisateurs doivent pouvoir s'authentifier en utilisant TOUS les mécanismes annoncés par Postfix sinon la négociation pourrait aboutir à un mécanisme non supporté et l'authentification échouera. Par exemple si vous configurez SASL pour utiliser `saslauthd` pour les authentifications via PAM (pluggable authentication modules), seuls les mécanismes PLAIN et LOGIN sont supportés et ont une chance de réussir alors que la librairie SASL supporte d'autres mécanismes tels DIGEST-MD5. Ceci arrive car ces mécanismes sont conçus sous forme de plugins et la librairie SASL n'a aucun moyen de savoir que votre source d'authentification est PAM. Ainsi vous devrez limiter la liste des mécanismes annoncés par Postfix.

- Avec les anciennes versions de Cyrus SASL, vous devez supprimer les librairies correspondantes du répertoire des plug-in SASL (à refaire à chaque fois que le système est mis à jour).
- Avec Cyrus SASL versions 2.1.1 et supérieures :

```
/usr/local/lib/sasl2/smtpd.conf:  
mech_list: plain login
```

Pour les mêmes raisons vous devrez limiter la liste des plugins utilisés par l'authentification.

- Avec la version 1.5.5 de SASL, votre seul choix est d'effacer les librairies correspondantes du répertoire `/usr/local/lib/sasl`.
- Avec la version 2.1.1 de SASL :

```
/usr/local/lib/sasl2/smtpd.conf:  
pwcheck_method: auxprop  
auxprop_plugin: sql
```

Lancer le logiciel dans une cage chroot en activant le support SASL est un exercice intéressant.

Tester l'authentification SASL dans le serveur SMTP de Postfix

Pour tester la partie serveur, connectez-vous au serveur SMTP, vous devriez être capable de suivre l'échange suivant. Les informations envoyées par le client sont en gras.

```
220 server.host.tld ESMTP Postfix  
EHLO client.host.tld  
250-server.host.tld  
250-PIPELINING  
250-SIZE 10240000
```

```
250-ETRN
250-AUTH DIGEST-MD5 PLAIN CRAM-MD5
250 8BITMIME
AUTH PLAIN dGVzdAB0ZXN0AHRlc3RwYXNz
235 Authentication successful
```

Remplacez `dGVzdAB0ZXN0AHRlc3RwYXNz` par l'encodage base64 de la chaîne utilisateur\0utilisateur\0password (le \0 représente un octet nul). L'exemple ci-dessus correspond au nom d'utilisateur ``test'` avec le mot de passe ``testpass'`.

Pour générer les informations d'authentification encodées en base64, vous pouvez utiliser l'une des commandes suivantes :

```
% printf 'username\0username\0password' | mmencode

% perl -MMIME::Base64 -e \
    'print encode_base64("username\0username\0password");'

% printf 'username\0username\0password' | openssl base64
```

La commande `mmencode` est une part des logiciels `metamail`. `MIME::Base64` is disponible sur le site <http://www.cpan.org/>.

Lorsque vous postez des négociations SASL dans les listes publiques, surtout n'oubliez pas qu'il est très facile de retrouver le couple login/mot-de-passe codé en base64.

Dépanner SASL

Vous trouverez un répertoire "sample" dans les sources de Cyrus SASL. Lancez `make`, changez l'utilisateur en *postfix* (commande `su`) (ou en l'utilisateur passé dans le paramètre *mail_owner*) :

```
% su postfix
```

puis lancez le serveur et le client issus de la compilation dans deux terminaux séparés. Utilisez `strace` / `ktrace` / `truss` pour constater et résoudre les problèmes. Répétez l'opération précédente jusqu'à réussir à vous authentifier avec le client. Après seulement, retournez à Postfix.

Activer l'authentification SASL dans le client SMTP de Postfix

Activez l'authentification SASL du côté client et spécifiez une table contenant des informations utilisateur/mot-de-passe par machine ou destination. Postfix regarde d'abord le nom de machine; si aucune entrée n'est trouvée, il consulte la table avec la destination prochaine. Généralement, c'est la partie droite de l'adresse de messagerie, mais ce peut être l'information contenue dans le paramètre *relayhost* ou dans une table *transport*.

```
/etc/postfix/main.cf:
    smtp_sasl_auth_enable = yes
    smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd

/etc/postfix/sasl_passwd:
    foo.com          username:password
```

Documentation de Postfix en français

```
bar.com                username
[mail.fai]             username:password
[mail.fai]:submission username:password
```

Postfix version 2.3 supporte des informations de mot-de-passe SASL par utilisateur. Pour rechercher le mot de passe SASL de Postfix par utilisateur avant de le chercher par destination, indiquez :

```
/etc/postfix/main.cf:
  smtp_sender_dependent_authentication = yes
  smtp_sasl_auth_enable = yes
  smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd

/etc/postfix/sasl_passwd:
  user@example.com      username:password
  bar.com               username
  [mail.myisp.net]      username:password
  [mail.myisp.net]:submission username:password
```

Note : certains serveurs SMTP supportent uniquement l'authentification PLAIN ou LOGIN. Par défaut, le client SMTP de Postfix n'utilise pas de méthodes d'authentification qui envoient le mot de passe en clair et retarde la livraison avec le message d'erreur : "Authentication failed: cannot SASL authenticate to server". Pour activer l'authentification en clair, indiquez par exemple :

```
/etc/postfix/main.cf:
  smtp_sasl_security_options = noanonymous
```

Le fichier des mots de passe SASL du client est ouvert avant que le serveur SMTP n'entre dans la cage chroot ainsi vous pouvez laisser le fichier dans le répertoire /etc/postfix.

Note : certains serveurs SMTP supportant les mécanismes d'authentification qui, bien que disponibles sur le système client, ne fonctionnent pas en pratique ou ne possèdent pas les éléments appropriés pour s'authentifier. Il est possible avec le paramètre smtp_sasl_mechanism_filter de restreindre la liste des mécanismes serveur que le client smtp(8) prendra en considération :

```
/etc/postfix/main.cf:
  smtp_sasl_mechanism_filter = !gssapi, !external, static:all
```

Dans l'exemple ci-dessus, Postfix refusera d'utiliser les mécanismes qui requièrent une infrastructure particulière tel Kerberos.

Le client SMTP de Postfix SMTP est compatible avec les serveurs SMTP qui utilisent la syntaxe non-standard "AUTH=method..." dans la réponse à la commande EHLO ; il n'y a pas besoin d'une configuration particulière pour cela.

Références

- Le support SASL fut initialement implémenté dans Postfix par Till Franke de la société SuSE Rhein/Main AG.
- Wietse a réduit le code aux seules fonctionnalités nécessaires.
- Jason Hoos a contribué au support de la version 2 de SASL.
- Liviu Daia a ajouté smtpd_sasl_application_name, découpé reject_sender_login_mismatch en reject_authenticated_sender_login_mismatch et reject_unauthenticated_sender_login_mismatch, et

revisé la documentation.

- Wietse a effectué un nouvel examen du code pour ajouter le support plug-in pour des implémentations SASL multiples.
- Le plug-in serveur SMTP Dovecot a été initialement implémenté par Timo Sirainen de Procontrol, Finlande.

Support d'IPv6 dans Postfix

Introduction

Postfix 2.2 a introduit le support du protocole IPv6 (IP version 6). Le support d'IPv6 pour les versions précédentes de Postfix était disponible sous forme d'un patch. Le paragraphe "Compatibilité avec le support d'IPv6 dans Postfix <2.2" ci-après présente les différences entre ces implémentations.

La principale fonctionnalité d'IPv6 est qu'il utilise des adresses composée de 128 bits au lieu de 32 bits dans IPv4. Il peut ainsi connecter un plus grand nombre de machines et de réseaux sans mécanismes type NAT. Un autre avantage de ce grand espace d'adressage est qu'il rend quasiment impossible le scan aléatoire d'un réseau.

Postfix utilise le même protocole SMTP sur IPv6 que sur IPv4, et effectue une recherche DNS de type AAAA en complément des anciennes requêtes de type A. Les informations sur IPv6 sont disponibles sur <http://www.ipv6.org/>.

Ce document fournit des informations sur les sujets suivants :

- Plateformes supportées
- Configuration
- Limitations connues
- Compatibilité avec le support IPv6 de Postfix <2.2
- IPv6 pour les plateformes non supportées
- Références

Plateformes supportées

Postfix version 2.2 supporte IPv4 et IPv6 sur les plateformes suivantes :

- AIX 5.1+
- Darwin 7.3+
- FreeBSD 4+
- Linux 2.4+
- NetBSD 1.5+
- OpenBSD 2+
- Solaris 8+
- Tru64Unix V5.1+

Sur les autres systèmes, Postfix utilisera simplement IPv4 comme il l'a toujours fait.

Reportez-vous au paragraphe consacré pour les astuces pour le portage du support IPv6 de Postfix sur les autres environnements.

Configuration

Le support IPv6 de Postfix introduit deux nouveaux paramètres de configuration dans main.cf, et modifie sensiblement la syntaxe de notation des adresses dans les listes de correspondances telles mynetworks ou debug_peer_list.

La syntaxe des adresses IPv6 de Postfix est assez astucieuse car il y a peu d'emplacements où vous devez encadrer une adresse IPv6 dans des crochets "[]", et peu également où vous ne le devrez pas. Il n'est intéressant d'utiliser "[]" qu'aux endroits où vous devez le faire. Consultez la page de manuel postconf(5) à chaque fois que vous travaillez sur des paramètres de configuration en relation avec IPv6.

- Plutôt que d'écrire en dur les adresses de la boucle locale 127.0.0.1 et ::1 dans master.cf, indiquez "inet_interfaces = loopback-only" dans main.cf. Vous pouvez faire de même que le système utilise IPv6 ou non.
- Le premier nouveau paramètre est appelé inet_protocols. Il indique quels protocoles Postfix doit utiliser pour établir ou accepter des connexions réseau, et également quels types de consultations DNS Postfix doit utiliser.

```
/etc/postfix/main.cf:
# Vous devez arrêter et redémarrer Postfix après avoir changer ce paramètre
inet_protocols = ipv4          (DÉFAUT: n'active qu'IPv4)
inet_protocols = all          (active IPv4 et IPv6 s'il est supporté)
inet_protocols = ipv4, ipv6   (active IPv4 et IPv6)
inet_protocols = ipv6        (n'active qu'IPv6)
```

Par défaut, Postfix n'utilise qu'IPv4, car la plupart des systèmes ne sont pas raccordés à un réseau IPv6.

- ◆ Sur les systèmes qui combinent les piles IPv4 et IPv6, tenter de livrer du courrier via IPv6 échouera systématiquement avec une erreur "network unreachable", et ces tentatives ralentiront Postfix.
- ◆ Les noyaux Linux kernels ne chargent pas le module IPv6 par défaut. Toute tentative échouera immédiatement.

Note 1: vous devez arrêter et redémarrer Postfix après avoir changé le paramètre de configuration inet_protocols.

Note 2: si vous voyez des messages d'erreur tels que les suivants, c'est que vous n'avez pas activé IPv6 dans votre Linux. Reportez-vous sur <http://www.ipv6.org/> pour les trucs et astuces. Contrairement à d'autres systèmes, Linux ne dispose pas d'une pile combinant IPv4 et IPv6, et le support de protocole IPv6 n'est pas chargé par défaut.

```
postconf: warning: inet_protocols: IPv6 support is disabled: Address family not supported
postconf: warning: inet_protocols: configuring for IPv4 support only
```

Note 3: sur les anciens systèmes Linux et Solaris, le paramètre "inet_protocols = ipv6" n'interdira pas à Postfix d'accepter des connexions IPv4. Postfix présentera cependant les adresses IP du client au format IPv6. Dans tous les autres cas, Postfix présente toujours les adresses IPv4 clientes dans la forme traditionnelle séparée par des points d'IPv4.

- L'autre nouveau paramètre est smtp_bind_address6. Il indique l'adresse d'interface locale utilisée pour les connexions IPv6 sortantes, tels que le fait le paramètre smtp_bind_address pour IPv4:

```
/etc/postfix/main.cf:
```

Documentation de Postfix en français

`smtp_bind_address6 = 2001:240:587:0:250:56ff:fe89:1`

- Si vous laissez le paramètre `mynetworks` à sa valeur par défaut (i.e. `mynetworks` non renseigné dans `main.cf`) Postfix trouvera seul ses adresses de réseau. C'est typiquement ce à quoi ressemble :

```
% postconf mynetworks
mynetworks = 127.0.0.0/8 168.100.189.0/28 [::1]/128 [fe80::]/10 [2001:240:587::]/64
```

Si vous renseignez la valeur du paramètre `mynetworks` dans `main.cf`, vous devez mettre à jour la valeur `mynetworks` pour inclure les réseaux IPv6. Assurez-vous de bien spécifier les adresses IPv6 entre crochets "[]", comme suit :

```
/etc/postfix/main.cf:
mynetworks = ...réseaux IPv4... [::1]/128 [2001:240:587::]/64 ...
```

NOTE: lorsque vous configurez des listes de correspondances telles `mynetworks` ou `debug_peer_list`, vous devez indiquer les adresses IPv6 entre crochet "[]" dans les valeurs des paramètres de `main.cf` et dans les fichiers indiqués dans des correspondances `"/nom/de/fichier"`. Les adresses IPv6 contiennent le caractère ":" et risqueraient autrement d'être confondues avec des correspondances `"type:table"`.

Limitations connues

- L'ordre des tentatives de connexions sortantes IPv6/IPv4 n'est pas actuellement configurable. Généralement, IPv6 est testé avant IPv4.
- Postfix ne supporte pas actuellement les consultations DNSBL (listes noires en temps réel) pour les adresses clientes IPv6 ; généralement il n'y a pas de listes qui couvrent les espaces d'adresses IPv6.
- IPv6 n'a pas de classes d'adresses réseaux A, B, C, etc.. Avec les réseaux IPv6, le paramètre `"mynetworks_style = class"` a le même effet que `"mynetworks_style = subnet"`.
- Sur Tru64Unix and AIX, Postfix ne peut détecter le masque du réseau local et suppose toujours qu'il s'agit d'un réseau /128. Ce n'est un problème qu'avec `"mynetworks_style = subnet"` ou si `mynetworks` n'est pas explicite dans `main.cf`.

Compatibilité avec le support IPv6 de Postfix <2.2

Le support IPv6 de Postfix version 2.2 est basé sur le patch Postfix/IPv6 de Dean Strik et autres mais diffère en quelques points.

- `main.cf`: Le paramètre `inet_interfaces` ne supporte pas la notation `"ipv6:all"` ou `"ipv4:all"`. Utilisez le paramètre `inet_protocols` à la place.
- `main.cf`: Indiquez `"inet_protocols = all"` ou `"inet_protocols = ipv4, ipv6"` Pour activer à la fois le support d'IPv4 et d'IPv6.
- `main.cf`: Le paramètre `inet_protocols` contrôle également quelles requêtes DNS Postfix utilisera pour envoyer et recevoir d courrier.
- `main.cf`: Indiquez `"inet_interfaces = loopback-only"` pour n'écouter que sur les interfaces réseaux de la boucle locale.
- Les fonctionnalités `lmtp_bind_address` et `lmtp_bind_address6` ont été oubliées. Le client LMTP de Postfix sera absorbé par le client SMTP, il n'y a donc aucune raison d'ajouter des fonctionnalités au client LMTP.
- Le serveur SMTP requiert maintenant que les adresses IPv6 dans les commandes SMTP soient indiquées sous la forme `[ipv6:ipv6address]`, tel que décrit dans la [RFC 2821](#).
- Le code d'examen des adresses des réseaux IPv6 a été entièrement réécrit et devrait être plus proche des spécifications. Le résultat peut être incompatible avec le patch Postfix/IPv6.

IPv6 pour les plateformes non supportées

Faire fonctionner IPv6 sur Postfix sur les autres systèmes passe par les étapes suivantes :

- Indiquez comment Postfix peut trouver les interfaces réseaux locales. Postfix a besoin de ces informations pour éviter les boucles de messages et pour savoir si un message à destination de *user@[ipaddress]* concerne une destination locale ou distante.

Si votre système dispose de la routine `getifaddrs()` alors ajoutez ce qui suit dans la section concernant votre plateforme dans `src/util/sys_defs.h` :

```
#ifndef NO_IPV6
# define HAS_IPV6
# define HAVE_GETIFADDRS
#endif
```

Autrement, si votre système dispose de la commande `SIOCGLIF ioctl()` dans `/usr/include/*/*.h`, ajoutez ce qui suit dans la section concernant votre plateforme dans `src/util/sys_defs.h` :

```
#ifndef NO_IPV6
# define HAS_IPV6
# define HAS_SIOCGLIF
#endif
```

Dans les autres cas, Postfix devra utiliser les anciennes commandes `SIOCGIF` et ne disposera que de fonctionnalités IPv6 restreintes (il ne pourra pas trouver vos masques réseaux IPv6 nécessaires pour "mynetworks_style = subnet". Ajoutez ceci dans la section concernant votre plateforme dans `src/util/sys_defs.h` :

```
#ifndef NO_IPV6
# define HAS_IPV6
#endif
```

- Testez si Postfix peut trouver les informations concernant ses interfaces.

Après avoir compilé Postfix selon la méthode habituelle, entrez dans le répertoire `src/util` et tapez "**make inet_addr_local**". Lancer ce fichier à la main devrait afficher toutes les adresses des interfaces ainsi que leurs masques ; par exemple :

```
% make
% cd src/util
% make inet_addr_local
[... messages divers ...]
% ./inet_addr_local
[... messages divers ...]
./inet_addr_local: inet_addr_local: configured 2 IPv4 addresses
./inet_addr_local: inet_addr_local: configured 4 IPv6 addresses
168.100.189.2/255.255.255.224
127.0.0.1/255.0.0.0
fe80:1::2d0:b7ff:fe88:2ca7/ffff:ffff:ffff:ffff::
2001:240:587:0:2d0:b7ff:fe88:2ca7/ffff:ffff:ffff:ffff::
fe80:5::1/ffff:ffff:ffff:ffff::
::1/ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
```

L'exemple précédent concerne une ancienne machine FreeBSD. Les autres systèmes produisent des résultats légèrement différents.

Si aucun résultat utilisable ne sort, envoyez un message à la liste `postfix-users@postfix.org` et nous essaieront de vous aider avec ceci.

Références

Les informations suivantes sont pour partie issues de celles rassemblées par Dean Strik.

- Mark Huizer a écrit le patch IPv6 original pour Postfix.
- Jun-ichiro 'itojun' Hagino du projet KAME a écrit des modifications sensibles. Depuis lors, nous parlons du patch KAME.
- La distribution PLD Linux a porté le code sur d'autres piles (en particulier USAGI). Nous parlons alors du patch PLD. Une caractéristique importante de ce patch est qu'il peut marcher avec celui de Lutz Jaenicke.
- Dean Strik a étendu le support d'IPv6 aux autres plateformes que KAME et USAGI, mis à jour le patch pour respecter le développement de Postfix et créé un patch combiné IPv6 + TLS. Les informations sur ces travaux peuvent être trouvées sur le site Postfix de Dean Strik's : <http://www.ipnet6.org/postfix/>.
- Wietse Venema a pris le patch IPv6 de Dean Strik, l'a inclus dans Postfix 2.2, et en a profité pour éliminer tous le code spécifique IPv4 de Postfix qui pouvait l'être. Sur les systèmes sans bibliothèques ou noyau supportant IPv6, Postfix dispose d'une simple couche de compatibilité pour qu'il utilise IPv4 comme avant.

Support TLS de Postfix

ATTENTION

En activant le support TLS dans Postfix, vous n'obtenez pas seulement la possibilité de chiffrer les messages et d'authentifier les clients et les serveurs. Vous activez également des milliers de lignes de code de la bibliothèque OpenSSL. En supposant qu'OpenSSL est écrit avec autant de soin que le code écrit par Wietse, chaque groupe de 1000 lignes introduit statistiquement un bug dans Postfix.

Ce que le support TLS de Postfix fait pour vous

La couche de sécurité du transport TLS (*Transport Layer Security*, également appelée SSL) fournit les authentifications basées sur des certificats et le chiffrement des sessions. Une session chiffrée protège les informations transmises par message SMTP ou par les authentifications SASL.

Postfix version 2.2 introduit le support TLS tel que décrit dans la [RFC 3207](#). Le support TLS pour les versions antérieures de Postfix était disponible sous forme de patch. Le paragraphe "[Compatibilité avec le support TLS Postfix < 2.2](#)" ci-après présente les différences entre ces implémentations.

Sujets abordés par ce document :

- [Comment fonctionne le support TLS de Postfix](#)
- [Compiler Postfix avec le support de TLS](#)
- [Paramètres spécifiques au serveur SMTP](#)
- [Paramètres spécifiques au client SMTP](#)
- [Paramètres spécifiques au gestionnaire TLS](#)
- [Rapporter les problèmes](#)
- [Compatibilité avec le support TLS Postfix < 2.2](#)
- [Références](#)

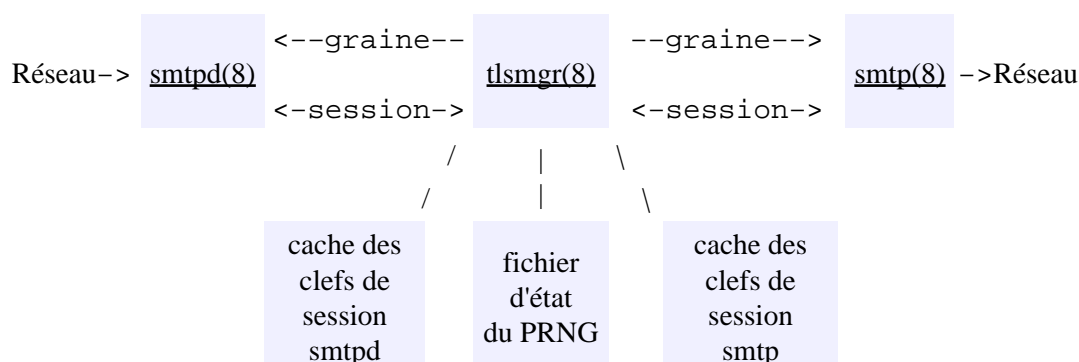
Et *last but not least* pour les impatientes :

- [Documentation rapide](#)

Comment fonctionne le support TLS de Postfix

Le schéma ci-dessous montre les principaux éléments de l'architecture TLS de Postfix et leurs relations. Les éléments colorés contenant un chiffre représentent des programmes démons de Postfix. Les autres éléments colorés représentent les éléments de stockages.

- Le serveur [smtpd\(8\)](#) implémente la partie serveur de TLS sur SMTP.
- Le client [smtp\(8\)](#) implémente la partie cliente de TLS sur SMTP.
- Le serveur [tlsmgr\(8\)](#) maintient le générateur de nombres pseudo-aléatoires (PRNG) qui égraine les moteurs TLS engines des processus du serveur [smtpd\(8\)](#) et du client [smtp\(8\)](#), et maintient le fichier de cache des clefs de session TLS.



Compiler Postfix avec le support de TLS

Pour compiler Postfix avec le support de TLS, nous devons d'abord générer les fichiers `make(1)` avec les définitions nécessaires. Ceci est fait en invoquant la commande `"make makefiles"` à la racine du répertoire des sources de Postfix avec les arguments présentés ci-dessous.

NOTE : n'utilisez pas Gnu TLS. Il interrompra spontanément un processus démon de Postfix qui se termine avec le code de retour 2 au lieu de permettre à Postfix de 1) rapporter l'erreur dans le journal et de 2) fournir le service en clair lorsqu'il est approprié.

- Si les fichiers include d'OpenSSL (tel `ssl.h`) sont dans le répertoire `/usr/include/openssl`, et que les bibliothèques OpenSSL (telles `libssl.so` et `libcrypto.so`) sont dans le répertoire `/usr/lib`:

```
% make tidy # si vous avez déjà compilé les sources dans ce répertoire
% make makefiles CCARGS="-DUSE_SSL" AUXLIBS="-lssl -lcrypto"
```

- Si les fichiers include d'OpenSSL (tels `ssl.h`) sont dans le répertoire `/usr/local/include/openssl`, et que les bibliothèques OpenSSL (telles `libssl.so` et `libcrypto.so`) sont dans le répertoire `/usr/local/lib`:

```
% make tidy # si vous avez déjà compilé les sources dans ce répertoire
% make makefiles CCARGS="-DUSE_SSL -I/usr/local/include" \
  AUXLIBS="-L/usr/local/lib -lssl -lcrypto"
```

Sur Solaris, utilisez l'option `-R` comme indiqué ci-dessous :

```
% make tidy # si vous avez déjà compilé les sources dans ce répertoire
% make makefiles CCARGS="-DUSE_TLS -I/usr/local/include" \
  AUXLIBS="-R/usr/local/lib -L/usr/local/lib -lssl -lcrypto"
```

Si vous devez appliquer d'autres personnalisations (telles les bases de données Berkeley DB, MySQL, PostgreSQL, LDAP ou SASL), lisez les documents README de Postfix correspondants, et combinez leurs instructions `"make makefiles"` avec les instructions ci-dessous :

```
% make tidy # si vous avez déjà compilé les sources dans ce répertoire
% make makefiles CCARGS="-DUSE_SSL \
  (autres options -D or -I)" \
  AUXLIBS="-lssl -lcrypto \
  (autres options -l pour les bibliothèques de /usr/lib \
  -L/chemin/ + -l options pour les autres bibliothèques)"
```


Pour terminer le processus de compilation, lisez les instructions de la page [INSTALL](#). Le support TLS désactivé par défaut dans Postfix, ainsi vous pouvez démarrer Postfix tel qu'il est installé.

Paramètres spécifiques au serveur SMTP

Sujets abordés dans ce paragraphe :

- [Configuration coté serveur du certificat et de la clef privée](#)
- [Enregistrement de l'activité TLS coté serveur](#)
- [Activer TLS dans le serveur SMTP de Postfix](#)
- [Vérification du certificat client](#)
- [Supporter l'authentification sur TLS seulement](#)
- [Cache des sessions TLS coté serveur](#)
- [Contrôle d'accès au serveur](#)
- [Contrôle du chiffrement coté serveur](#)
- [Contrôles divers du serveur](#)

Configuration coté serveur du certificat et de la clef privée

Pour utiliser TLS, le serveur SMTP de Postfix a besoin d'un certificat et d'une clef privée. Les deux doivent être au format PEM. La clef privée ne doit pas être chiffrée, ce qui signifie : la clef doit être accessible sans mot-de-passe. Le certificat et la clef privée peuvent être dans le même fichier.

Les certificats RSA et DSA sont tous deux supportés. Généralement, vous n'aurez que des certificats RSA issus d'une autorité de certification commerciale. En complément, les outils fournis avec OpenSSL généreront par défaut des certificats RSA. Vous pouvez avoir les deux en même temps, auquel cas le chiffrement utilisé détermine le certificat présenté. Pour les clients Netscape et les clients SSL ouverts sans choix particulier de chiffrement, le certificat RSA est préféré.

Pour permettre aux clients SMTP distants de vérifier le certificat du serveur SMTP de Postfix, le certificat de l'autorité (dans le cas d'une chaîne de certification, tous les certificats des autorités) doit être disponible. Vous devrez ajouter ces certificats au certificat du serveur, le certificat en premier suivi des autorités fournissant le certificat.

Exemple: le certificat pour "server.dom.ain" est issu de "intermediate CA" elle-même certifiée par "root CA". Créez le fichier server.pem avec :

```
cat server_cert.pem intermediate_CA.pem root_CA.pem > server.pem
```

Le certificat d'un serveur SMTP de Postfix utilisé ici doit être utilisable comme certificat d'un serveur SSL et donc passer le test "openssl verify -purpose sslserver ...".

Un client qui agréé l'autorité de certification (CA) racine dispose d'une copie locale de son certificat, donc il n'est pas nécessaire d'inclure le certificat racine ici. On réduit ainsi le surplus de trafic réseau lié à TLS.

Si vous voulez que le serveur SMTP de Postfix accepte les certificats des clients SMTP issus de ces mêmes autorités, vous pouvez également ajouter les certificats des autorités au fichier `$smtpd_tls_CAfile` ou les installer dans le répertoire `$smtpd_tls_CApath`. Lorsque vous utilisez une autorité racine, il n'est pas nécessaire d'agréer explicitement les autorités intermédiaires signée par l'autorité racine, sauf si le nombre `$smtpd_tls_ccert_verifydepth` est inférieur au nombre d'autorités de la chaîne de certification du client. Avec

une profondeur de vérification de 1, vous ne pouvez vérifier que les certificats directement issus d'une des autorités agréées. Avec une profondeur de 2, vous pouvez vérifier les certificats signés par l'autorité racine ou par une intermédiaire directe (sous réserve que le client soit correctement configuré pour fournir son certificat d'autorité intermédiaire).

Exemples de clefs et certificats RSA :

```
/etc/postfix/main.cf
smtpd_tls_cert_file = /etc/postfix/server.pem
smtpd_tls_key_file = $smtpd_tls_cert_file
```

L'équivalent DSA :

```
/etc/postfix/main.cf
smtpd_tls_dcert_file = /etc/postfix/server-dsa.pem
smtpd_tls_dkey_file = $smtpd_tls_dcert_file
```

Pour vérifier un certificat client SMTP, le serveur SMTP de Postfix a besoin de connaître les certificats des autorités les fournissant. Ces certificats au format "pem" peuvent être stockés dans un seul fichier `$smtpd_tls_CAfile` ou dans de multiples fichiers, une CA par fichier dans le répertoire `$smtpd_tls_CApith`. Si vous utilisez un répertoire, n'oubliez pas de créer les nécessaires liens "hash" avec :

```
# $OPENSSL_HOME/bin/c_rehash /chemin/du/répertoire
```

Le fichier `$smtpd_tls_CAfile` contient les certificats d'une ou de plusieurs autorités agréées. Le fichier est ouvert (avec les privilèges root) avant que Postfix n'entre dans l'optionnelle cage chroot et n'a donc pas besoin d'être accessible depuis la cage.

Des autorités additionnelles peuvent être ajoutées via le répertoire `$smtpd_tls_CApith`, auquel cas les certificats sont lus (avec le compte `$mail_owner`) dans ce répertoire en cas de besoin. Ainsi, le répertoire `$smtpd_tls_CApith` doit être accessible depuis l'intérieur de la cage chroot.

Lorsque vous configurez Postfix pour exiger les certificats clients (en indiquant `$smtpd_tls_ask_ccert = yes`), tous les certificats présents dans `$smtpd_tls_CAfile` sont envoyés au client, afin de lui permettre de choisir une identité signée par une autorité que vous agréez. Si aucun fichier `$smtpd_tls_CAfile` n'est indiqué, aucune liste de préférence n'est envoyée et le client est libre de présenter le certificat de son choix. La plupart des clients utilisent une identité fixée sans examiner la liste de préférence et vous pouvez réduire le surplus de trafic lié à la négociation TLS en installant tout ou partie de vos certificats dans `$smtpd_tls_CApith`. Dans ce cas, vous n'êtes pas obligé de renseigner `$smtpd_tls_CAfile`.

Notez que bien que les certificats clients sont utilisés pour autoriser l'accès aux clients authentifiés par TLS, il est préférable de ne pas exiger de certificats clients, car outre l'augmentation de trafic, certains clients (notamment gmail dans certains cas) ne sont pas en mesure de compléter l'échange TLS lorsqu'un certificat client est exigé.

Exemple:

```
/etc/postfix/main.cf :
smtpd_tls_CAfile = /etc/postfix/CAcert.pem
smtpd_tls_CApith = /etc/postfix/certs
```

Enregistrement de l'activité TLS coté serveur

Pour obtenir des informations complémentaires sur l'activité TLS du serveur SMTP de Postfix, vous pouvez augmenter le niveau de log de 0 à 4. Chaque niveau de log inclut également les informations des niveaux inférieurs.

- 0 Désactive l'enregistrement de l'activité TLS.
- 1 Enregistre les informations de négociation TLS et des certificats.
- 2 Enregistre les niveaux durant la négociation TLS.
- 3 Retranscrit en hexadécimal et ASCII le processus de négociation TLS
- 4 Retranscrit en hexadécimal et ASCII toute la transmission après STARTTLS

Utilisez le niveau 3 seulement en cas de problème. L'utilisation du niveau 4 est vivement déconseillé.

Exemple :

```
/etc/postfix/main.cf :  
    smtpd_tls_loglevel = 0
```

Pour inclure les informations sur le protocole et le chiffrement utilisés ainsi que les CommonName du client et de l'autorité émettrice dans l'en-tête de message "Received:", mettez la variable smtpd_tls_received_header à "yes". La valeur par défaut est "no" car l'information n'est pas nécessairement authentique. Seule l'information enregistrée par la destination finale est fiable car les en-têtes peuvent être modifiés par les serveurs intermédiaires.

Exemple :

```
/etc/postfix/main.cf :  
    smtpd_tls_received_header = yes
```

Activer TLS dans le serveur SMTP de Postfix

Par défaut, TLS est désactivé dans le serveur SMTP de Postfix, ainsi aucune différence avec Postfix standard n'est visible. Activez-le explicitement en utilisant "smtpd_use_tls = yes".

Exemple :

```
/etc/postfix/main.cf :  
    smtpd_use_tls = yes
```

A partir de là, le serveur SMTP de Postfix annonce le support de STARTTLS aux clients SMTP mais n'exige pas les clients l'utilisent.

Note : lorsqu'un utilisateur non privilégié invoque "sendmail -bs", STARTTLS n'est jamais proposé en raison d'une insuffisance de privilèges pour accéder à la clef privée du serveur.

Vous pouvez FORCER l'emploi de TLS, ainsi le serveur SMTP de Postfix n'accepte aucune commande (à l'exception de QUIT bien sûr) sans chiffrement TLS, en indiquant "smtpd_enforce_tls = yes". En accord avec la RFC 2487 ceci NE DOIT PAS être fait dans le cas d'un serveur SMTP référencé et public. Donc cette option est désactivée par défaut et ne doit que rarement être utilisée.

Exemple :

```
/etc/postfix/main.cf :  
    smtpd_enforce_tls = yes
```

TLS est parfois utilisé dans le mode non standard "wrapper" où le serveur utilise systématiquement TLS au lieu d'annoncer le support de STARTTLS et attendre que les clients requièrent les service TLS. Certains clients, nommés Outlook [Express] préfèrent utiliser le mode "wrapper". Ceci est vrai pour OE (Win32 < 5.0 et Win32 >= 5.0 lorsque le port utilisé diffère de 25) et OE (5.01 Mac et tous les portages).

Il est strictement déconseillé d'utiliser ce mode dans main.cf. Si vous voulez supporter ce service, activez un port particulier dans master.cf et indiquez "-o smtpd_tls_wrappermode = yes" en option de la ligne de commande de smtpd(8). Le port 465 (smtps) a été choisi pour cette fonctionnalité.

Exemple :

```
/etc/postfix/main.cf :  
    smtpd_tls_wrappermode = no
```

Vérification du certificat client

Pour recevoir le certificat d'un client SMTP extérieur, le serveur SMTP de Postfix doit le demander explicitement en envoyant les certificats \$smtpd_tls_CAfile au client. Malheureusement, les clients Netscape généreront une alerte si aucun certificat client ne correspond ou offriront à l'utilisateur un choix parmi une liste de certificats. Ceci peut être gênant, ainsi cette option est désactivée par défaut. Vous aurez également besoin du certificat si vous voulez accepter le relais par certificat avec, par exemple, l'option permit_tls_client_certs.

Exemple :

```
/etc/postfix/main.cf :  
    smtpd_tls_ask_ccert = no
```

Vous déciderez peut-être de requérir un certificat pour les client SMTP avant d'autoriser les connexions TLS. Cette fonctionnalité est incluse pour la perfection et implique "smtpd_tls_ask_ccert = yes".

Soyez attentif au fait que ceci interdit les connexions TLS sans certificat client conforme et n'a de sens que si les soumission sans TLS sont désactivées (smtpd_enforce_tls = yes). Autrement, les clients pourront outrepasser la restriction simplement en utilisant pas STARTTLS.

Lorsque TLS n'est pas forcé, la connexion sera traitée ainsi seulement si "smtpd_tls_ask_ccert = yes", sinon un avertissement est enregistré.

Exemple :

```
/etc/postfix/main.cf :  
    smtpd_tls_req_ccert = no
```

Une profondeur de vérification des certificats de 1 est suffisante si le certificat est directement issu d'une autorité listée dans le fichier CA. La valeur par défaut (5) devrait suffire pour les chaînes plus longues (autorité racine certifiant une autorité de laquelle le certificat est issu...)

Exemple :

```
/etc/postfix/main.cf :
  smtpd_tls_ccert_verifydepth = 5
```

Supporter l'authentification sur TLS seulement

Envoyer des données d'authentification sur un canal non crypté pose un problème de sécurité. Lorsque le chiffrement TLS est requis (`smtpd_enforce_tls = yes`), le serveur SMTP de Postfix annoncera AUTH et l'acceptera seulement après que la couche TLS ait été activée avec STARTTLS. Lorsque la couche TLS est optionnelle (`smtpd_enforce_tls = no`), il peut être pratique de n'offrir AUTH que si TLS est active. Pour maintenir la compatibilité avec les clients non-TLS, la valeur par défaut est d'accepter AUTH sans chiffrement. Pour changer ce comportement, indiquez "`smtpd_tls_auth_only = yes`".

Exemple :

```
/etc/postfix/main.cf :
  smtpd_tls_auth_only = no
```

Cache des sessions TLS coté serveur

Le serveur SMTP de Postfix et le client SMTP distant négocient une session qui utilise du temps CPU et de la bande passante. Par défaut, cette information de session est cachée seulement dans le processus `smtpd(8)` utilisant actuellement cette session et est perdue lorsque le processus s'arrête. Pour partager les informations de session entre de multiples processus `smtpd(8)`, un cache de session persistant peut être utilisé. Vous pouvez utiliser tous les types de bases de données qui peuvent stocker des objets de plusieurs octets et qui supportent l'opérateur de séquence. Les bases DBM ne peuvent être utilisées car elles ne stockent que de petits objets. Le cache est maintenu par le processus `tlsmgr(8)`, il n'y a donc pas de problèmes d'accès concurrents. Le cache des sessions est fortement recommandé car le coût des négociations TLS répétées est très élevé.

Exemple :

```
/etc/postfix/main.cf :
  smtpd_tls_session_cache_database = btree:/etc/postfix/smtpd_scache
```

Les informations de session cachée par les serveurs SMTP de Postfix expirent après un certain temps. Postfix/TLS n'utilise pas la valeur par défaut de OpenSSL (300 secondes), mais un temps plus long de 3600 secondes (=1 heure). La [RFC 2246](#) recommande un maximum de 24 heures.

Exemple :

```
/etc/postfix/main.cf :
  smtpd_tls_session_cache_timeout = 3600s
```

Contrôle d'accès au serveur

Le support TLS de Postfix introduit deux fonctionnalités additionnelles pour le contrôle d'accès au serveur SMTP de Postfix :

`permit_tls_clientcerts`

Autorise les requêtes des client SMTP si le certificat du client passe la vérification, et

si son empreinte est listée dans la liste des certificats clients (voyez la présentation de `relay_clientcerts` ci-dessous).

`permit tls all clientcerts`

Autorise les requêtes des client SMTP si le certificat du client passe la vérification.

`check ccert access type:table`

Si le certificat client passe la vérification, utilise son empreinte (*fingerprint*) comme clef pour la table d'accès indiquée.

La fonctionnalité `permit tls all clientcerts` doit être utilisé avec précaution, car elle peut engendrer trop de permissions d'accès. Utilisez cette fonctionnalité seulement si une autorité particulière fournit les certificats client et seule cette autorité est reconnue dans le liste des autorités certifiées. Si d'autres autorités sont reconnues, tous les propriétaires de certificats valides seront autorisés. La fonctionnalité `permit tls all clientcerts` peut être pratique pour un serveur relais créé spécialement.

Il est généralement recommandé de s'en tenir à la fonctionnalité `permit tls clientcerts` et de lister tous les certificats dans `$relay_clientcerts`, alors que `permit tls all clientcerts` ne permet aucun contrôle lorsqu'un certificat ne doit plus être utilisé (un employé parti...).

Exemple :

```
/etc/postfix/main.cf :
    smtpd_recipient_restrictions =
        ...
        permit tls clientcerts
        reject unauth_destination
        ...
```

Les routines de manipulation des listes de Postfix donnent un traitement particulier aux espaces et à certains autres caractères, rendant impossible l'utilisation des noms des certificats. A la place, nous utilisons les empreintes des certificats qui sont difficile à fausser mais faciles à utiliser dans les consultations. Les tables de correspondances de Postfix ont la forme de paires (clef, valeur) pairs. Puisque nous n'avons besoin que de la clef, la valeur peut être choisie librement, c'est à dire le nom de l'utilisateur ou de la machine.

Exemple :

```
/etc/postfix/main.cf :
    relay_clientcerts = hash:/etc/postfix/relay_clientcerts

/etc/postfix/relay_clientcerts :
    D7:04:2F:A7:0B:8C:A5:21:FA:31:77:E1:41:8A:EE:80 lutzpc.at.home
```

Contrôle du chiffrement coté serveur

Pour influencer le schéma de sélection du chiffrement du serveur SMTP de Postfix, vous pouvez indiquer une chaîne listant les chiffrements. Pour une description plus détaillée, reportez-vous à la documentation OpenSSL. Si vous ne savez pas de quoi il s'agit, ne touchez à rien et laissez les valeurs (openssl-)compilées par défaut !

N'UTILISEZ PAS de guillemets "" pour encadrer la chaîne, indiquez juste une chaîne !!!

Exemple :

```
/etc/postfix/main.cf :
```

```
smtpd_tls_cipherlist = DEFAULT
```

Si vous voulez utiliser les avantages des chiffrements avec EDH, les paramètres DH sont nécessaires. Au lieu d'utiliser les paramètres DH compilés pour 1024bit et 512bit, il est préférable de générer vos "propre" paramètres, entre autres les attaquants pourraient tenter une attaque force brute contre les paramètres utilisés par tout le monde. Pour cette raison, les paramètres choisis seront différents que ceux distribués avec les autres packages TLS.

Pour générer vos propres paramètres DH, utilisez :

```
% openssl gendh -out /etc/postfix/dh_1024.pem -2 -rand /var/run/egd-pool 1024
% openssl gendh -out /etc/postfix/dh_512.pem -2 -rand /var/run/egd-pool 512
```

Exemple :

```
/etc/postfix/main.cf :
smtpd_tls_dh1024_param_file = /etc/postfix/dh_1024.pem
smtpd_tls_dh512_param_file = /etc/postfix/dh_512.pem
```

Contrôles divers du serveur

Le paramètre smtpd_starttls_timeout limite le temps d'écriture et de lecture des opérations TLS durant les procédures de démarrage et d'arrêt TLS.

Exemple :

```
/etc/postfix/main.cf :
smtpd_starttls_timeout = 300s
```

Paramètres spécifiques au client SMTP

Sujets abordés dans ce paragraphe :

- [Configuration coté client du certificat et de la clef privée](#)
- [Enregistrement de l'activité TLS coté client](#)
- [Cache des sessions TLS coté client](#)
- [Activer TLS dans le client SMTP de Postfix](#)
- [Exiger le chiffrement TLS](#)
- [Désactiver la vérification du certificat serveur](#)
- [Politiques TLS par site](#)
- [Fermer un trou de sécurité lié au DNS avec des politiques TLS par site](#)
- [Découvrir les serveurs qui supportent TLS](#)
- [Profondeur de vérification du certificat serveur](#)
- [Contrôle du chiffrement coté client](#)
- [Contrôles divers du client](#)

Configuration coté client du certificat et de la clef privée

Durant la négociation de démarrage TLS, le client SMTP de Postfix peut présenter un certificat au serveur SMTP distant. Le client Netscape est plutôt intelligent ici et laisse l'utilisateur choisir seulement parmi les certificats qui correspondent aux certificats des autorités offerts par le serveur SMTP distant. Comme le client

SMTP de Postfix utilise la fonction "SSL_connect()" du package OpenSSL, on ne peut choisir qu'un certificat. Ainis pour l'instant, le défaut est de n'utiliser aucun certificat ni clef sauf si un est explicitement indiqué ici.

Les certificats RSA et DSA sont tous deux supportés. Vous pouvez avoir les deux en même temps, auquel cas le chiffrement utilisé détermine le certificat présenté.

Il est possible pour le client SMTP de Postfix d'utiliser la même paire clef/certificat que le serveur SMTP de Postfix. Si un certificat est présenté, il doit être au format PEM. La clef privée ne doit pas être chiffrée, en d'autre termes, elle doit être accessible sans mot-de-passe. Les deux parties (certificat et clef privée) peuvent être dans le même fichier.

Pour les les serveurs SMTP distant vérifient le certificat du client SMTP de Postfix, le certificat de l'autorité (dans le cas d'une chaîne de certificat, tous les certificats des autorités) doivent être disponibles. Vous devrez ajouter ces certificats au certificat client : le certificat client en premier, puis les autorités émettrices.

Exemple : le certificat de "client.dom.ain" est issu de "intermediate CA" elle-même certifiée par "root CA". Créez le fichier client.pem avec :

```
% cat client_cert.pem intermediate_CA.pem root_CA.pem > client.pem
```

Un certificat client SMTP de Postfix fourni ici doit être utilisable comme certificat client SSL et donc passer le test "openssl verify -purpose sslclient ...".

Un serveur qui agrée une autorité racine dispose d'une copie locale du certificat de cette autorité, ainsi il n'est pas nécessaire de l'inclure ici. L'enlever du fichier "client.pem" réduit le surplus de trafic lié à l'échange TLS.

Si vous voulez que le client SMTP de Postfix accepte les certificats issus de ces autorités, vous devez également ajouter leurs certificats dans le fichier smtp_tls_CAfile, ou les installer dans le répertoire smtp_tls_CApeth. Lorsque vous agréez une autorité racine, il n'est pas nécessaire d'agréer explicitement les autorités intermédiaires qu'elle a signé, sauf si le nombre \$smtp_tls_scert_verifydepth est inférieur au nombre d'autorités de la chaîne de certification. Avec une profondeur de 1, vous ne pouvez vérifier que les certificats directement issu d'une autorité agréée. Avec une profondeur de 2, vous pouvez vérifier les certificats signés par une autorité intermédiaire signée directement par une autorité que vous agréez (pour autant que le serveur soit correctement configuré pour fournir les certificats intermédiaires).

Exemples de clefs et certificats RSA :

```
/etc/postfix/main.cf :  
smtp_tls_cert_file = /etc/postfix/client.pem  
smtp_tls_key_file = $smtp_tls_cert_file
```

L'équivalent DSA :

```
/etc/postfix/main.cf :  
smtp_tls_dcet_file = /etc/postfix/client-dsa.pem  
smtp_tls_dkey_file = $smtpd_tls_cert_file
```

Pour vérifier un certiicat serveur SMTP extérieur, le client SMTP de Postfix doit reconnaître les certificats des autorités émettrices. Ces certificats au format PEM peuvent être stockés dans un seul fichier \$smtp_tls_CAfile ou dans plusieurs fichiers dans le répertoire \$smtp_tls_CApeth. Si vous utilisez un répertoire, n'oubliez pas de créer les nécessaires liens "hash" avec :


```
# $OPENSSL_HOME/bin/c_rehash /chemin/vers/le/répertoire
```

Le fichier `$smtpd_tls_CAfile` contient les certificats d'une ou de plusieurs autorités agréées. Le fichier est ouvert (avec les privilèges root) avant que Postfix n'entre dans l'optionnelle cage chroot et n'a donc pas besoin d'être accessible depuis la cage.

Des autorités additionnelles peuvent être ajoutées via le répertoire `$smtpd_tls_CApath`, auquel cas les certificats sont lus (avec le compte `$mail_owner`) dans ce répertoire en cas de besoin. Ainsi, le répertoire `$smtpd_tls_CApath` doit être accessible depuis l'intérieur de la cage chroot.

Le choix entre `$smtpd_tls_CAfile` et `$smtpd_tls_CApath` est une question d'espace/temps. S'il y a beaucoup d'autorités agréées, le coût de leur préchargement en mémoire peut dépasser l'économie d'un plus rapide accès lorsque le certificat est requis.

Exemple :

```
/etc/postfix/main.cf :  
smtpd_tls_CAfile = /etc/postfix/CAcert.pem  
smtpd_tls_CApath = /etc/postfix/certs
```

Enregistrement de l'activité TLS coté client

Pour obtenir des informations complémentaires sur l'activité TLS du client SMTP de Postfix, vous pouvez augmenter le niveau de log de 0 à 4. Chaque niveau de log inclut également les informations des niveaux inférieurs.

- 0 Désactive l'enregistrement de l'activité TLS.
- 1 Enregistre les informations de négociation TLS et des certificats.
- 2 Enregistre les niveaux durant la négociation TLS.
- 3 Retranscrit en hexadécimal et ASCII le processus de négociation TLS
- 4 Retranscrit en hexadécimal et ASCII toute la transmission après STARTTLS

Exemple :

```
/etc/postfix/main.cf :  
smtpd_tls_loglevel = 0
```

Cache des sessions TLS coté client

Le serveur SMTP de Postfix et le client SMTP distant négocient une session qui utilise du temps CPU et de la bande passante. Par défaut, cette information de session est cachée seulement dans le processus `smtpd(8)` utilisant actuellement cette session et est perdue lorsque le processus s'arrête. Pour partager les informations de session entre de multiples processus `smtpd(8)`, un cache de session persistant peut être utilisé. Vous pouvez utiliser tous les types de bases de données qui peuvent stocker des objets de plusieurs octets et qui supportent l'opérateur de séquence. Les bases DBM ne peuvent être utilisées car elles ne stockent que de petits objets. Le cache est maintenu par le processus `tlsmgr(8)`, il n'y a donc pas de problèmes d'accès concurrents. Le cache des sessions est fortement recommandé car le coût des négociations TLS répétées est très élevé. Les futurs serveurs SMTP de Postfix pourront limiter le nombre de sessions qu'un client est autorisé à négocier par unité de temps.

Exemple :

```
/etc/postfix/main.cf :  
smtp_tls_session_cache_database = btree:/etc/postfix/smtp_scache
```

Les informations de sessions cachées par les clients SMTP de Postfix expirent après un certain temps. Postfix/TLS n'utilise pas la valeur par défaut d'OpenSSL (300 secondes), mais un délai plus long de 3600 secondes (=1 heure). La [RFC 2246](#) recommande un maximum de 24 heures.

Exemple :

```
/etc/postfix/main.cf :  
smtp_tls_session_cache_timeout = 3600s
```

Activer TLS dans le client SMTP de Postfix

Par défaut, TLS est désactivé dans le client SMTP de Postfix, ainsi aucune différence avec Postfix classique n'est visible. Si vous activez TLS, le client SMTP de Postfix enverra STARTTLS lorsque le support TLS est annoncé par le serveur SMTP distant.

Lorsque le serveur accepte la commande STARTTLS, mais que la négociation TLS échoue et qu'aucun autre serveur n'est disponible, le client SMTP de Postfix retarde la tentative de livraison et le message reste en file d'attente. Après un échec de négociation, le canal de communication se trouve dans un état indéterminé et ne peut être réutilisé pour des livraisons en clair.

Exemple :

```
/etc/postfix/main.cf :  
smtp_use_tls = yes
```

Exiger le chiffrement TLS

Vous pouvez FORCER l'utilisation de TLS, ainsi le client SMTP de Postfix ne délivrera aucun message sur une connexion non chiffrée. Dans ce mode, le nom de machine du serveur SMTP doit correspondre aux informations contenues dans le certificat du serveur et ce dernier doit être issu d'une autorité reconnue par le client SMTP de Postfix. Si l'une de ces conditions n'est pas vérifiée, la livraison est retardée et le message reste en file d'attente.

Le nom de machine du serveur SMTP utilisé dans la vérification en question doit être le nom de machine principal (aucun CNAME n'est autorisé ici). Les comparaisons sont effectuées avec tous les noms fournis comme dNSNames dans le champ SubjectAlternativeName. Si aucun dNSNames n'est trouvé, le champ CommonName est utilisé. Ce comportement peut être modifié avec l'option `smtp_tls_enforce_peername` présentée ci-dessous.

Cette option n'est utilisable que si vous ne vous connectez qu'à des serveurs supportant la [RFC 2487](#) et qui présentent des certificats serveurs convenant aux deux conditions nécessaires. Un exemple pourrait être un client envoyant du courrier seulement à un commutateur de messagerie spécifique qui offre le nécessaire support STARTTLS.

Exemple :

```
/etc/postfix/main.cf :
```

smtp_enforce_tls = yes

Conformément à la [RFC 2487](#), l'exigence de l'examen du nom de machine n'est pas requis pour les clients MTA. Lorsque TLS est requis (smtp_enforce_tls = yes), l'option smtp_tls_enforce_peername peut être mise à "no" pour désactiver l'examen strict du nom de machine du serveur SMTP. Dans ce cas, les messages seront livrés sans regarder les champs CommonName, etc. listés dans le certificat.

En dépit du potentiel pour éliminer les attaques type "man-in-the-middle" ou autre, exiger une vérification des certificats suivant le nom n'est pas viable comme politique de livraison de courrier par défaut. Une part non négligeable des MTA supportant TLS disposent de certificats auto-signés ou de certificats issus d'une autorité privée. Sur une machine qui livre du courrier sur Internet, si vous indiquez smtp_enforce_tls = yes, vous devrez probablement ajouter smtp_tls_enforce_peername = no. Vous pouvez utiliser une politique TLS par site (voir ci-dessous) pour activer la vérification complète pour certaines destinations connues pour avoir des certificats valides.

Exemple :

```
/etc/postfix/main.cf :  
smtp_enforce_tls = yes  
smtp_tls_enforce_peername = yes
```

Politique TLS par site

Une faible part des serveurs proposent STARTTLS mais la négociation échoue, laissant le message dépasser le temps limite en file d'attente puis retourner à son expéditeur. Dans de tels cas, vous pouvez utiliser une politique par site pour désactiver TLS pour les sites à problème. Inversement, vous pouvez activer TLS seulement pour quelques sites et pas dans le cas général.

La table smtp_tls_per_site est consultée pour trouver une politique qui correspond aux informations suivantes.

nom de machine du serveur SMTP distant

C'est simplement le nom DNS du serveur auquel le client SMTP de Postfix se connecte ; ce nom peut être obtenu par d'autres consultation DNS telle les requêtes MX ou CNAME.

destination suivante

C'est normalement la partie domaine de l'adresse de destination, mais peut-être surchargée par les informations contenues dans la table transport(5), dans le paramètre relayhost, ou dans le paramètre relay_transport. Lorsque ce n'est pas le domaine du destinataire, la destination suivante peut prendre la forme spécifique à Postfix "[nom]", [nom]:port, "nom" or "nom:port".

Lorsque la consultation sur le nom de machine et sur la destination suivante réussissent, la politique correspondant au nom de machine ne prend pas automatiquement le pas sur l'autre. A la place, la préférence est donnée à la plus spécifique ou la plus sûre des politiques décrites ci-dessous.

La table smtp_tls_per_site utilise un simple format "*nom espace valeur*". Indiquez des noms de machine ou de destination sur la partie gauche – les cartes blanches ne sont pas autorisées. Sur la partie droite, indiquez l'un des mots-clefs suivants :

NONE

Ne pas utiliser TLS. Ceci surcharge un résultat de consultation moins spécifique **MAY** issu de la consultation avec l'autre clef (nom de machine ou destination suivante) et surcharge les paramètres globaux smtp_use_tls, smtp_enforce_tls, et smtp_tls_enforce_peername.

MAY

Tente d'utiliser TLS si le serveur en annonce le support, sinon utilise une connection en clair. Cette politique dispose d'une moindre préférence qu'un résultat plus précis (y compris **NONE**) issu de la consultation avec l'autre clef (nom de machine ou destination suivante) ou des cas où la configuration globale contient "smtp_enforce_tls = yes" ou "smtp_tls_enforce_peername = yes".

MUST_NOPEERMATCH

Exige le chiffrement TLS, mais pas que le nom de machine du serveur SMTP distant ne corresponde aux informations contenues dans le certificat, ou que celui-ci soit issu d'une autorité agréée. Cette politique dispose d'une préférence plus élevée que la moins sécurisée **NONE** ou la moins spécifique **MAY** qui pourraient résulter de la consultation avec l'autre clef, et surcharge les paramètres globaux smtp_use_tls, smtp_enforce_tls et smtp_tls_enforce_peername.

MUST

Exige le chiffrement TLS, une correspondance entre le nom de machine du serveur SMTP distant et les informations contenues dans son certificat et la signature d'une autorité agréée. Cette politique dispose d'une préférence plus élevée que les moins sécurisées **NONE** et **MUST_NOPEERMATCH** ou la moins spécifique **MAY** qui pourraient résulter de la consultation avec l'autre clef, et surcharge les paramètres globaux smtp_use_tls, smtp_enforce_tls et smtp_tls_enforce_peername.

Les préférences entre les politiques TLS globales (main.cf) et spécifiques par site peuvent être résumées comme suit :

- Lorsque ni le nom de machine du serveur SMTP distant ni la destination suivante ne sont trouvés dans la table smtp_tls_per_site, la politique basée sur smtp_use_tls, smtp_enforce_tls et smtp_tls_enforce_peername. Note : "smtp_enforce_tls = yes" et "smtp_tls_enforce_peername = yes" implique "smtp_use_tls = yes".
- Lorsque à la fois le nom de machine du serveur SMTP distant et la destination suivante produisent un résultat, la politique par site la plus spécifique (**NONE**, **MUST**, etc.) surcharge la moins spécifique (**MAY**) et la plus sécurisée (**MUST**, etc.) surcharge la moins sécurisée (**NONE**).
- Après que les politiques par site aient été combinées, le résultat surcharge généralement la politique globale. Une exception : une politique par site **MAY** est surchargée par la politique plus spécifique "smtp_enforce_tls = yes" avec une vérification du certificat serveur correspondant au paramètre smtp_tls_enforce_peername.

Fermer un trou de sécurité lié au DNS avec des politiques TLS par site

Tant qu'aucun mécanisme de consultation sécurisé du DNS n'existe, de faux noms de machines dans les réponses MX ou CNAME peuvent changer le nom de machine que Postfix utilise pour les recherches de politique de sécurité et la vérification de certificat. Même avec une correspondance parfaite entre le nom de machine du serveur et le certificat du serveur, il n'y a pas de garantie absolue que Postfix est connecté au bon serveur. Pour éviter ce trou de sécurité, procédez suivant les étapes suivantes :

- Éliminez les consultations MX. Utilisez une table de transport(5) locale pour les domaines sensibles avec une destination explicite smtp:[mailhost] ou smtp:[mailhost]:port (vous garantirez mieux la

sécurité de cette table que celle du DNS) ; dans la table smtp_tls_per_site utilisez la valeur **MUST** pour la clef [mailhost] ou smtp:[mailhost]:port. Ceci évite que des fausses informations sur les noms de machine dans les enregistrements MX du DNS ne changent le nom de machine que Postfix utilise pour les recherches de politique TLS et les vérifications des certificats.

- Désactivez les surcharges de noms de machine par CNAME. Dans main.cf ajoutez "smtp_cname_overrides_servername = no". Ceci évite qu'une fausse information dans les enregistrements CNAME du DNS ne changent le nom de machine que Postfix utilise pour les recherches de politique TLS et les vérifications des certificats. Cette fonctionnalité est disponible dans les versions 2.2.9 et supérieures de Postfix.

Exemple:

```
/etc/postfix/main.cf:
    smtp_tls_per_site = hash:/etc/postfix/tls_per_site
    relayhost = [msa.exemple.net]:587

/etc/postfix/tls_per_site:
# relayhost correspondance exacte pour la destination suivante
[msa.exemple.net]:587      MUST

# TLS ne doit pas être utilisé avec les serveurs MX de exemple.org.
exemple.org               NONE

# TLS ne doit pas être utilisé avec le serveur smtp.exemple.com.
smtp.exemple.com          NONE
```

Découvrir les serveurs qui proposent TLS

Si vous choisissez "par site" si oui ou non utiliser TLS, il peut être intéressant d'avoir une liste des sites qui offrent "STARTTLS". Nous pouvons les collecter avec cette option :

Si la fonctionnalité smtp_tls_note_starttls_offer est activée et qu'un serveur offre STARTTLS alors que TLS n'est pas activé pour ce serveur, le client SMTP de Postfix enregistre une ligne telle :

```
postfix/smtp[pid]: Host offered STARTTLS: [hostname.exemple.com]
```

Exemple :

```
/etc/postfix/main.cf :
    smtp_tls_note_starttls_offer = yes
```

Vérification du certificat du serveur

Lors de la vérification du certificat d'un serveur SMTP distant, une profondeur de vérification de 1 est suffisante si le certificat est directement issu d'une autorité indiquée dans smtp_tls_CAfile ou smtp_tls_CApith. La valeur par défaut de 5 doit suffire pour les chaînes plus longues (une autorité racine certifie une autorité particulière certifiant elle-même le certificat présenté...)

Exemple :

```
/etc/postfix/main.cf :
    smtp_tls_scert_verifydepth = 5
```

Contrôle du chiffrement coté client

Pour influencer le schéma de sélection du chiffrement du client SMTP de Postfix, vous pouvez indiquer une chaîne liste de chiffrement. Pour la description détaillée consultez la documentation OpenSSL. Si vous n'y connaissez rien, laissez simplement la valeur (openssl-)compilée par défaut !

N'UTILISEZ PAS de guillemets "" pour encadrer la chaîne, indiquez juste une chaîne !!!

Exemple :

```
/etc/postfix/main.cf :
smtp_tls_cipherlist = DEFAULT
```

Contrôles divers du client

Le paramètre `smtp_starttls_timeout` limite la durée des procédures d'établissement et de rupture de la session TLS. En cas de problèmes, le client SMTP de Postfix tente l'adresse suivante de la liste des échangeur de messagerie et retarde la livraison si aucun serveur alternatif n'est disponible.

Exemple :

```
/etc/postfix/main.cf :
smtp_starttls_timeout = 300s
```

Paramètres spécifiques au gestionnaire TLS

La sécurité des logiciels de chiffrement tels TLS dépend fortement de leur capacité à générer des nombres aléatoires pour les clefs et quelques autres informations. A cette fin, le processus `tlsmgr(8)` maintient un groupe de générateurs de nombres pseudo-aléatoires (Pseudo Random Number Generator PRNG). Il est consulté par les processus `smtp(8)` et `smtpd(8)` à leur initialisation. Par défaut, ces démons demande 32 octets à la fois, l'équivalent de 256 bits. C'est amplement suffisant pour générer des clefs de sessions 128 (ou 168) bits.

Exemple :

```
/etc/postfix/main.cf :
tls_daemon_random_bytes = 32
```

Pour alimenter son pool PRNG en mémoire, `tlsmgr(8)` lit l'entropie depuis une source extérieure au démarrage ainsi que pendant le fonctionnement. Indiquez une bonne source telle EGD ou `/dev/urandom` ; assurez-vous de n'utiliser que des sources non bloquantes. Si la source d'entropie n'est pas un fichier régulier, vous devez préfixer le nom de la source avec son type : "dev:" pour un fichier device ou "egd:" pour une source disposant d'une interface socket compatible EGD.

Exemples (utilisez en un seul dans `main.cf`) :

```
/etc/postfix/main.cf :
tls_random_source = dev:/dev/urandom
tls_random_source = egd:/var/run/egd-pool
```

Par défaut, `tlsmgr(8)` lit 32 octets à la fois de la source d'entropie externe. Cette valeur (<=>256 bits) est

largement suffisante pour générer des clefs symétriques de 128 bits. Avec des sources d'entropie EGD et dev:, tlsmgr(8) limite la quantité de données lue à la fois à 255 octets. Si vous utilisez un fichier régulier comme source d'entropie, une quantité supérieure peut être lue.

Exemple :

```
/etc/postfix/main.cf :  
    tls_random_bytes = 32
```

Pour mettre à jour son pool PRNG en mémoire, tlsmgr(8) interroge la source d'entropie externe après un délai aléatoire. Celui-ci est calculé en utilisant le PRNG, et est compris entre 0 and et le temps maximum indiqué au paramètre tls_random_reseed_period. Le délai maximal par défaut est d'1 heure.

Exemple :

```
/etc/postfix/main.cf :  
    tls_random_reseed_period = 3600s
```

Le processus tlsmgr(8) sauvegarde l'état du PRNG dans un fichier d'échange persistant à intervalles réguliers et lorsque le processus se termine, ainsi il peut récupérer cet état au prochain démarrage. Ce fichier est créé lorsqu'il n'existe pas. Son emplacement par défaut se trouve dans le répertoire de configuration de Postfix, ce qui n'est pas le meilleur emplacement pour les informations modifiées par Postfix. À la place, ce fichier devrait probablement se trouver dans la partition /var (mais **pas** dans la cage chroot).

Exemples:

```
/etc/postfix/main.cf:  
    tls_random_exchange_name = /etc/postfix/prng_exch  
    tls_random_prng_update_period = 3600s
```

Documentation rapide

Les étapes suivantes vont vous permettre de déparer rapidement. Comme vous signez votre propre certificat, vous obtiendrez le chiffrement TLS mais pas l'authentification TLS. C'est suffisant pour des tests et pour échanger des messages avec les sites avec lesquels vous n'avez pas de relation de certification. Pour obtenir une réelle authentification, le certificat de votre Postfix doit être signé par une autorité de certification reconnue et Postfix doit être configuré avec une liste des certificats des autorités agréées pour pouvoir vérifier les certificats présentés par les autres serveurs.

Dans les exemples ci-dessous, l'utilisateur est montré en police **gras** et un "#" indique un shell du super-utilisateur.

- Devenez votre propre autorité de certification pour pouvoir signer votre propre clef publique. Cet exemple utilise le script CA.pl fourni avec OpenSSL. Par défaut, OpenSSL l'installe dans /usr/local/ssl/misc/CA.pl, mais cela peut varier. Ce script crée une clef privée ./demoCA/private/cakey.pem et une publique ./demoCA/cacert.pem.

```
% /usr/local/ssl/misc/CA.pl -newca  
  
CA certificate filename (or enter to create)  
  
Making CA certificate ...
```

Documentation de Postfix en français

```
Using configuration from /etc/ssl/openssl.cnf
Generating a 1024 bit RSA private key
.....+++++
....+++++
writing new private key to './demoCA/private/akey.pem'
Enter PEM pass phrase:à-votre-choix
```

- Créez une clef privée non protégée par mot-de-passe pour la machine FOO et un certificat non signé.

```
% openssl req -new -nodes -keyout FOO-key.pem -out FOO-req.pem -days 365
Using configuration from /etc/ssl/openssl.cnf
Generating a 1024 bit RSA private key
.....+++++
....+++++
writing new private key to 'FOO-key.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:Ile de France
Locality Name (eg, city) []:Paris
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Porcupine
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:FOO
Email Address []:wietse@porcupine.org

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:à-votre-choix
An optional company name []:
```

- Signez le certificat de la machine FOO avec la clef privée de l'autorité créée plus haut.

```
% openssl ca -out FOO-cert.pem -infiles FOO-req.pem
Using configuration from /etc/ssl/openssl.cnf
Enter PEM pass phrase:whatever
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName             :PRINTABLE:'FR'
stateOrProvinceName     :PRINTABLE:'Ile de France'
localityName            :PRINTABLE:'Paris'
organizationName        :PRINTABLE:'Porcupine'
commonName              :PRINTABLE:'FOO'
emailAddress            :IA5STRING:'wietse@porcupine.org'
Certificate is to be certified until Nov 21 19:40:56 2005 GMT (365 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

- Installez la clef privée du serveur, son certificat et celui de l'autorité de certification. Ceci requiert les privilèges du super-utilisateur.

```
# cp demoCA/cacert.pem FOO-key.pem FOO-cert.pem /etc/postfix
# chmod 644 /etc/postfix/FOO-cert.pem /etc/postfix/cacert.pem
# chmod 400 /etc/postfix/FOO-key.pem
```


- Configurez Postfix, en ajoutant ce qui suit au fichier `/etc/postfix/main.cf`.

```
smtp_tls_CAfile = /etc/postfix/cacert.pem
smtp_tls_cert_file = /etc/postfix/FOO-cert.pem
smtp_tls_key_file = /etc/postfix/FOO-key.pem
smtp_tls_session_cache_database = btree:/var/run/smtp_tls_session_cache
smtp_use_tls = yes
smtpd_tls_CAfile = /etc/postfix/cacert.pem
smtpd_tls_cert_file = /etc/postfix/FOO-cert.pem
smtpd_tls_key_file = /etc/postfix/FOO-key.pem
smtpd_tls_received_header = yes
smtpd_tls_session_cache_database = btree:/var/run/smtpd_tls_session_cache
smtpd_use_tls = yes
tls_random_source = dev:/dev/urandom
```

Rapporter les problèmes

Lorsque vous rapportez un problème, soyez le plus complet possible. Les patches, si possible, sont également très appréciés.

Différenciez si possible :

- les problèmes dans le code TLS : `<postfix_tls@aet.tu-cottbus.de>`
- les autres problèmes dans Postfix : `<postfix-users@postfix.org>`

Compatibilité avec le support TLS de Postfix < 2.2

Le support TLS de Postfix version 2.2 est basé sur le patch TLS de Lutz Jänicke, mais diffère quelque peu.

- main.cf: utilisez "btree" au lieu de "sdbm" pour les bases de données utilisées pour le cache des sessions TLS.

Les bases de données utilisées pour le cache des sessions TLS sont utilisées maintenant seulement par le processus tlsmgr(8), ainsi il n'y a plus de problèmes d'accès concurrents. Comme Postfix dispose d'un client sdbm, la librairie sdbm (près de 1000 lignes de code) n'est pas incluse dans Postfix.

Les caches de session TLS peuvent utiliser toutes les bases de données qui peuvent stocker des objets de plusieurs kilooctets et qui implémentent l'opération de séquence. Dans la plupart des cas, btree est adéquat.

NOTE : Vous ne pouvez pas utiliser de bases dbm. Les objets de session TLS sont trop grands.

- master.cf: utilisez "unix" au lieu de "fifo" dans le type de service de tlsmgr.

Les processus smtp(8) et smtpd(8) utilisent maintenant un protocole client-serveur pour accéder au groupe de générateurs de nombres pseudo-aléatoires (PRNG) de tlsmgr(8), et pour accéder au cache des sessions TLS. Un tel protocole ne peut être utilisé dans des fifos.

- smtp_tls_per_site: la politique par site `MUST_NOPEERMATCH` ne peut surcharger le paramètre global `"smtp_tls_enforce_peername = yes"`.
- smtp_tls_per_site: un résultat de consultation combiné (NONE + MAY) pour (nom de machine et destination suivante) produit un résultat non intuitif suivant les paramètres main.cf. TLS est activé avec `"smtp_tls_enforce_peername = no"`, mais est désactivé lorsque `"smtp_enforce_tls = yes"` et `"smtp_tls_enforce_peername = yes"`.

Les limitations de smtp_tls_per_site ont été supprimées à la fin du cycle de support de Postfix 2.2.

Références

- Le support de TLS pour Postfix a été initialement développé par Lutz Jänicke de l'université "Cottbus Technical University".
- Wietse Venema a adopté le code, effectué quelques restructurations et compilé cette partie de la documentation depuis celle de Lutz.
- Victor Duchovni a été l'initiateur de la réimplémentation du code de smtp_tls_per_site pour les niveaux d'application, ce qui a grandement simplifié l'implémentation.

Installation de Postfix depuis le

code source

1 – but de ce document

Ce document est un document autonome destiné à vous aider à installer et lancer Postfix depuis les sources avec un nombre minimal d'étapes. Si vous utilisez une version pré-compilé de Postfix, vous devriez lire la documentation générale de Postfix qui vise à décrire plus précisément le système. Ce document autonome ne doit pas être considéré comme faisant partie de la documentation générale de Postfix.

Ce document décrit comment compiler, installer et configurer un système Postfix afin qu'il puisse faire l'une des opérations suivantes :

- Envoyer seulement du courrier sans changer l'installation existante de Sendmail.
- Envoyer et recevoir du courrier via une interface virtuelle toujours sans changer l'installation existante de Sendmail.
- Remplacer Sendmail.

Sujets abordés par ce document :

1. but de ce document
2. Conventions typographiques
3. Documentation
4. Compiler Postfix sur un système supporté
5. Porter Postfix sur un système non supporté
6. Installer le logiciel après une compilation réussie
7. Configurer Postfix pour envoyer seulement du courrier
8. Configurer Postfix pour envoyer et recevoir du courrier via une interface virtuelle
9. Remplacer Sendmail par Postfix
10. Édition obligatoire des fichiers de configuration
11. Mettre en cage chroot ou non
12. Soins et alimentation du système Postfix

2 – Conventions typographiques

Dans les instructions suivantes, une commande écrite comme suit :

```
# command
```

doit être exécutée par le superutilisateur.

Une commande passée après le '%' :

```
% command
```

doit être exécutée par un utilisateur non privilégié.

3 – Documentation

La documentation est disponible sous forme de fichiers README (démarré avec le fichier README_FILES/AAAREADME), de pages HTML (faites pointer votre navigateur sur "html/index.html") et de pages de manuel dans le style UNIX.

Vous pouvez consulter les fichiers README avec un pagineur tel `more(1)` ou `less(1)`, car ils utilisent des caractères spéciaux pour produire afficher des éléments en **gras**. Pour imprimer un fichier README sans les caractères spéciaux, utilisez la commande `col(1)`. Par exemple :

```
% col -bx <file | lpr
```

Pour consulter les pages de manuel avant d'installer Postfix, modifiez votre variable d'environnement `MANPATH` pour qu'elle pointe sur les sous-répertoire "man" ; assurez-vous d'utiliser un chemin absolu.

```
% export MANPATH; MANPATH="`pwd`/man:$MANPATH"
% setenv MANPATH "`pwd`/man:$MANPATH"
```

Un intérêt tout particulier doit-être porté à la page de manuel [postconf\(5\)](#) qui liste tous les plus de 300 paramètres de configuration. La version HTML de ces textes rend la navigation plus facile.

Tous les fichiers sources de Postfix ont leur propre page de manuel. Les outils pour extraire ces pages de manuel embarquées sont disponibles dans le répertoire "mantools".

4 – Compiler Postfix sur un système supporté

A l'heure actuelle, les systèmes suivants sont supportés par une version de Postfix :

- AIX 3.2.5, 4.1.x, 4.2.0, 4.3.x, 5.2
- BSD/OS 2.x, 3.x, 4.x
- Darwin 1.x
- FreeBSD 2.x, 3.x, 4.x, 5.x
- HP-UX 9.x, 10.x, 11.x
- IRIX 5.x, 6.x
- Linux Debian 1.3.1, 2.x, 3.x
- Linux RedHat 3.x (Janvier 2004) – 9.x
- Linux Slackware 3.x, 4.x, 7.x
- Linux SuSE 5.x, 6.x, 7.x
- Mac OS X
- NEXTSTEP 3.x
- NetBSD 1.x
- OPENSTEP 4.x
- OSF1.V3 – OSF1.V5 (Digital UNIX)
- Reliant UNIX 5.x
- Rhapsody 5.x
- SunOS 4.1.4 (Janvier 2004)
- SunOS 5.4 – 5.9 (Solaris 2.4..9)
- Ultrix 4.x (depuis longtemps...)

ainsi que des systèmes semblables.

4.1 – Démarrer la compilation

Sur Solaris, la commande "make" et les autres utilitaires pour le développement des logiciels sont dans le répertoire /usr/ccs/bin, ainsi vous DEVEZ avoir /usr/ccs/bin dans votre chemin de recherche de commandes (\$PATH).

Si vous devez compiler Postfix pour de multiples architectures, utilisez la commande "Indir" pour construire un répertoire virtuel avec des liens symboliques vers les fichiers sources. "Indir" est un élément de X11R6.

Si à un moment du processus de compilation vous obtenez un messages tel : "make: don't know how to ..." vous devriez pouvoir vous en sortir en lançant la commande suivante à la racine des sources :

```
% make -f Makefile.init makefiles
```

Si vous copiez les sources de Postfix après les avoir compilé sur une autre machine, il est préférable de lancer la commande suivante à la racine des sources :

```
% make tidy
```

Ceci nettoie les sources de toute dépendance.

4.2 – Quel compilateur utiliser

Pour compiler avec GCC ou avec un compilateur natif s'il est plus adapté à votre système, déplacez-vous simplement à la racine des sources et lancez :

```
% make
```

Pour compiler avec un autre compilateur que celui par défaut, vous devez indiquer son nom. Ci-dessous quelques exemples :

```
% make makefiles CC=/opt/SUNWsprow/bin/cc      (Solaris)
% make
```

```
% make makefiles CC="/opt/ansic/bin/cc -Ae"    (HP-UX)
% make
```

```
% make makefiles CC="purify cc"
% make
```

et ainsi de suite. Dans certains cas, l'optimisation est désactivé automatiquement.

4.3 – Compiler avec des extensions optionnelles

Par défaut, Postfix se compile en système de messagerie avec relativement peu d'extensions, support de base de données tierce, etc. doivent être configurées lorsque Postfix est compilé. Le document suivant décrit comment compiler Postfix avec le support des extensions :

Extension de Postfix	Document	Disponibilité
----------------------	----------	---------------

Bases de données Berkeley	DB_README	Postfix 1.0
Bases de données LDAP	LDAP_README	Postfix 1.0
Bases de données MySQL	MYSQL_README	Postfix 1.0
Expressions rationnelles compatibles Perl	PCRE_README	Postfix 1.0
Bases de données PostgreSQL	PGSQL_README	Postfix 2.0
Authentification SASL	SASL_README	Postfix 1.0
Chiffrement de session STARTTLS	TLS_README	Postfix 2.2

Note : le support IPv6 est compilé dans Postfix sur les systèmes d'exploitation qui supportent IPv6. Reportez-vous au fichier [IPV6_README](#) pour les détails.

4.4 – Surcharger les valeurs des paramètres par défaut

Tous les paramètres de configuration de Postfix peuvent être changés en éditant le fichier de configuration à l'exception d'un : le paramètre qui indique l'emplacement des fichiers de configuration de Postfix. Pour compiler Postfix avec un répertoire de configuration autre que /etc/postfix, utilisez :

```
% make makefiles CCARGS='-DDEF_CONFIG_DIR=\" /nouvel/emplacement \"'
% make
```

IMPORTANT : Assurez-vous d'utiliser des apostrophes simples. Ces détails sont importants.

Les paramètres dont la valeur par défaut peut être indiquée de cette manière sont :

Nom de macro	valeur par défaut pour	valeur habituelle
DEF_COMMAND_DIR	command_directory	/usr/sbin
DEF_CONFIG_DIR	config_directory	/etc/postfix
DEF_DAEMON_DIR	daemon_directory	/usr/libexec/postfix
DEF_MAILQ_PATH	mailq_path	/usr/bin/mailq
DEF_HTML_DIR	html_directory	no
DEF_MANPAGE_DIR	manpage_directory	/usr/local/man
DEF_NEWALIAS_PATH	newaliases_path	/usr/bin/newaliases
DEF_QUEUE_DIR	queue_directory	/var/spool/postfix
DEF_README_DIR	readme_directory	no
DEF_SENDMAIL_PATH	sendmail_path	/usr/sbin/sendmail

4.5 – Support de milliers de processus

Pour compiler Postfix pour une utilisation forte où vous pensez avoir besoin de plus de 1000 processus de livraison de courrier, vous devez surcharger la définition de la macro FD_SETSIZE pour que make select() fonctionne correctement :

```
% make makefiles CCARGS=-DFD_SETSIZE=2048
```

Attention : ce qui précède n'a aucun effet sur certaines versions de Linux. Apparemment, sur ces systèmes la valeur FD_SETSIZE peut être changée uniquement en utilisant des interfaces non documentées.

Généralement, cela se traduit par l'inclusion directe de <bits/types.h> (ce qui n'est pas autorisé) et réécrire la

macro `__FD_SETSIZE`. Soyez prudent, les interfaces non documentées peuvent changer à tout moment sans avertissements préalables.

4.6 – Finalement, compiler Postfix

Si la commande

```
% make
```

réussit, alors vous pouvez procéder à l'installation de Postfix (paragraphe 6).

Si la commande produit des messages d'erreur de compilation, il faut chercher sur le web ou poser une question sur la liste de diffusion `postfix-users@postfix.org` (en anglais), cherchez d'abord dans les archives de la liste pour ne pas reposer une question déjà traitée. Certains sites d'archives de liste de diffusion sont indiqués sur le site <http://www.postfix.org/>.

5 – Porter Postfix sur un système non supporté

Chaque type de système que Postfix connaît est identifié par un nom unique. Exemples : SUNOS5, FREEBSD4, et ainsi de suite. La première étape pour porter Postfix sur un nouveau système est de choisir un nom (SYSTEMTYPE) pour chaque nouveau système. Vous devez utiliser un nom qui inclut en suffixe le numéro de version majeure du système d'exploitation (tel SUNOS4 ou LINUX2), ainsi différentes versions du même système peuvent être supportées sans confusion.

Ajoutez un cas au script shell "makedefs" du répertoire racine des sources qui reconnait le nouveau système et émet les bonnes informations spécifiques au système. Créez un code robuste contre les erreurs du PATH de l'utilisateur ; si le système offre plusieurs noms d'UNIX (e.g. BSD et SYSV) assurez-vous de construire un code pour un le nom natif plutôt qu'un nom émulé.

Ajoutez un paragraphe "#ifdef SYSTEMTYPE" dans le fichier d'inclusion central `util/sys_defs.h`. Vous pouvez avoir à inventer de nouveaux noms de macro. Choisissez s'il vous plaît des noms de macro tels `HAS_DBM` ou `FIONREAD_IN_SYS_FILIO_H`.

Je vous recommande fortement de ne pas utiliser de "#ifdef SYSTEMTYPE" individuellement dans chacun des fichiers source. Bien que ça paraisse être la solution la plus rapide, il sera plus compliqué d'effectuer des changements de version.

6 – Installer le logiciel après une compilation réussie

Ce paragraphe décrit comment installer Postfix depuis le code source. Reportez-vous à la page [PACKAGE_README](#) si vous compilez un package pour une distribution sur d'autres systèmes. Lisez [auxiliary/MacOSX/README-INSTALL.OSX](#) pour plus d'informations sur l'installation de Postfix depuis les sources sur Mac OS X.

6.1 – Sauver les binaires Sendmail existants

IMPORTANT : si vous REMPLACEZ une installation de Sendmail existante avec Postfix, vous devriez sauvegarder les anciens programmes de sendmail afin de vider la file d'attente. Avec le compte du super-utilisateur, lancez la commande suivante (vos programmes sendmail, newaliases et mailq peuvent se situer à un autre

emplacement) :

```
# mv /usr/sbin/sendmail /usr/sbin/sendmail.OFF
# mv /usr/bin/newaliases /usr/bin/newaliases.OFF
# mv /usr/bin/mailq /usr/bin/mailq.OFF
# chmod 755 /usr/sbin/sendmail.OFF /usr/bin/newaliases.OFF \
  /usr/bin/mailq.OFF
```

6.2 – Create account and groups

Avant d'installer Postfix pour la première fois, vous devez créer un compte et un groupe :

- Créez un compte utilisateur "postfix" avec un numéro (id) et un numéro de groupe non utilisé par un autre compte utilisateur. De préférence, ce compte est ne doit pas pouvoir être utilisé pour se connecter. Il ne doit pas non plus avoir de shell exécutable ni de répertoire utilisateur. Mes entrées de fichiers passwd et group ressemblent à :

```
/etc/passwd:
postfix:*:12345:12345:postfix:/no/where:/no/shell

/etc/group:
postfix:*:12345:
```

Note : il ne doit pas y avoir d'espace avant "postfix:".

- Créez un groupe "postdrop" avec un numéro (id) non utilisé par un compte utilisateur ni par le compte "postfix". L'entrée de mon fichier "group" ressemble à

```
/etc/group:
postdrop:*:54321:
```

Note : il ne doit pas y avoir d'espace avant "postdrop:".

6.3 – Installer Postfix

Pour installer ou mettre à jour Postfix depuis des sources compilées, lancez l'une des commandes suivantes sous le compte super-utilisateur :

```
# make install          (version interactive, première installation)

# make upgrade          (version non-interactive, mises à jour)
```

- La version non-interactive ("make upgrade") requiert le fichier /etc/postfix/main.cf de l'installation précédente; Si ce fichier n'existe pas, utilisez l'installation interactive ("make install").
- La version interactive offre quelques choix pour les chemins qui peuvent surcharger interactivement et sauvegarder vos préférences dans le fichier /etc/postfix/main.cf pour faciliter les futures mises à jour.

6.4 – Configurer Postfix

Lisez ce paragraphe pour savoir comment utiliser Postfix sur une machine particulière :

- Envoyer du courrier seulement, sans changer une installation de Sendmail existante (paragraphe 7).

- Envoyer et recevoir du courrier par une interface virtuelle, toujours sans changer l'installation existante de Sendmail (paragraphe 8).
- Utiliser Postfix au lieu de Sendmail (paragraphe 9).

7 – Configurer Postfix seulement pour envoyer du courrier

Si vous souhaitez utiliser Postfix pour seulement envoyer du courrier, il n'y a pas à changer votre installation existante de Sendmail. À la place, changez votre agent de messagerie utilisateur pour qu'il appelle le programme sendmail de Postfix.

Suivez les instructions "Edition obligatoire des fichier de configuration" du paragraphe 10, et "mettre en cage chroot ou non" du paragraphe 11.

Vous DEVEZ commenter l'entrée "smtp inet" du fichier /etc/postfix/master.cf, pour éviter les conflits avec le réel sendmail : insérez un caractère "#" au début de la ligne qui définit le service smtpd service :

```
/etc/postfix/master.cf:
#smtp      inet  n       -       n       -       -       smtpd
```

Lancez le système Postfix :

```
# postfix start
```

ou si vous êtes nostalgique, utilisez la commande sendmail de Postfix :

```
# sendmail -bd -qwhatever
```

et regardez votre fichier de logs de messagerie pour détecter d'éventuels messages d'erreur. Son emplacement généralement /var/log/maillog, /var/log/mail ou /var/log/syslog. Il est défini dans le fichier /etc/syslog.conf.

```
% egrep '(reject|warning|error|fatal|panic):' /some/log/file
```

Note : le plus important message d'erreur est enregistré en premier. Les messages suivants n'ont pas d'importance.

Pour examiner la file d'attente des messages, utilisez une des commandes suivantes :

```
% mailq
% sendmail -bp
% postqueue -p
```

Voyez également le paragraphe "Soins et alimentation" (n° 12 ci-dessous).

8 – Configurer Postfix pour envoyer et recevoir du courrier par une interface virtuelle

Alternativement, vous pouvez utiliser le système Postfix pour envoyer ET recevoir le courrier tout en conservant intact votre installation Sendmail, en utilisant Postfix sur une adresse d'interface virtuelle. Configurez simplement votre agent utilisateur de messagerie pour invoquer directement le programme

sendmail de Postfix.

Dans le fichier `/etc/postfix/main.cf`, j'indiquerais

```
/etc/postfix/main.cf:
myhostname = virtual.host.tld
inet_interfaces = $myhostname
mydestination = $myhostname
```

Suivez les instructions du paragraphe 10 ("édition obligatoire des fichiers de configuration"), et lisez le paragraphe 11 ("mettre en cage chroot ou non").

Lancez le système Postfix :

```
# postfix start
```

ou, si vous êtes nostalgique, utilisez la commande sendmail de Postfix :

```
# sendmail -bd -qwhatever
```

et regardez votre fichier de logs de messagerie pour détecter d'éventuels messages d'erreur. Son emplacement généralement `/var/log/maillog`, `/var/log/mail` ou `/var/log/syslog`. Il est défini dans le fichier `/etc/syslog.conf`.

```
% egrep '(reject|warning|error|fatal|panic):' /some/log/file
```

Note : le plus important message d'erreur est enregistré en premier. Les messages suivants n'ont pas d'importance.

Pour examiner la file d'attente des messages, utilisez une des commandes suivantes :

```
% mailq
% sendmail -bp
% postqueue -p
```

Voyez également le paragraphe "Soins et alimentation" (n° 12 ci-dessous).

9 – Utiliser Postfix au lieu de Sendmail

Avant d'installer Postfix vous devriez sauvegarder toutes les commandes du programme sendmail tel que décrit au paragraphe 6. Assurez-vous de laisser fonctionner quelques jours l'ancien Sendmail pour traiter les éventuels messages non envoyés. Pour ce faire, arrêtez le démon et relancez-le avec l'option "-q" :

```
# /usr/sbin/sendmail.OFF -q
```

Note : il s'agit de l'ancienne syntaxe de sendmail. Les versions récentes utilisent des processus séparés pour la soumission des messages et la gestion des files d'attente.

Après avoir lu le paragraphe "édition obligatoire des fichiers de configuration" ci-dessous, vous pouvez lancer le système Postfix avec :

```
# postfix start
```

ou si vous êtes nostalgique, utilisez la commande sendmail de Postfix :

```
# sendmail -bd -qwhatever
```

et regardez votre fichier de logs de messagerie pour détecter d'éventuels messages d'erreur. Son emplacement généralement /var/log/maillog, /var/log/mail ou /var/log/syslog. Il est défini dans le fichier /etc/syslog.conf.

```
% egrep '(reject|warning|error|fatal|panic):' /some/log/file
```

Note : le plus important message d'erreur est enregistré en premier. Les messages suivants n'ont pas d'importance.

Pour examiner la file d'attente des messages, utilisez une des commandes suivantes :

```
% mailq
```

```
% sendmail -bp
```

```
% postqueue -p
```

Voyez également le paragraphe "[Soins et alimentation](#)" (n° 12 ci-dessous).

10 – Édition obligatoire des fichiers de configuration

Note : les éléments présentés dans ce paragraphe sont couverts avec plus de détails à la page [BASIC CONFIGURATION README](#). L'information présentée ici est destinée aux administrateurs expérimentés.

10.1 – Fichiers de configuration de Postfix

Par défaut, les fichiers de configuration de Postfix se situent dans /etc/postfix. Les deux plus importants sont main.cf et master.cf ; ces fichiers doivent appartenir à root. Donner à quelqu'un d'autre la permission d'écrire dans main.cf ou master.cf (ou dans leur répertoire parent) signifie donner des privilèges root à cette personne.

Dans /etc/postfix/main.cf, vous aurez à indiquer un nombre minimum de paramètres de configuration. Les paramètres de configuration de Postfix ressemblent à des variables shell, avec deux différences importantes : la première est que Postfix ne connaît pas les apostrophes comme le shell UNIX.

Vous renseignez un paramètre de configuration comme suit :

```
/etc/postfix/main.cf:
    parametre = valeur
```

et vous l'utilisez en le faisant précéder d'un caractère "\$" :

```
/etc/postfix/main.cf:
    autre_parametre = $parametre
```

Vous pouvez utiliser \$parametre avant qu'il ne soit déclaré (c'est la seconde différence avec les variables du shell UNIX). Le langage de configuration de Postfix utilise une évaluation faible et n'examine pas la valeur d'un paramètre avant qu'elle ne soit utilisée.

A chaque changement du fichier `main.cf` ou `master.cf`, lancez la commande suivante pour rafraîchir un système de messagerie en fonctionnement :

```
# postfix reload
```

10.2 – Domaine par défaut pour les adresses non qualifiées

En premier lieu, vous devez indiquer le domaine qui sera ajouté à une adresse non-qualifiée (c'est à dire une adresse sans `@domain.tld`). le paramètre `myorigin` contient par défaut le nom de la machine locale ce qui est généralement correct pour de tout petits sites.

Quelques exemples (utilisez-en un seulement) :

```
/etc/postfix/main.cf:
myorigin = $myhostname      (envoie le courrier comme "utilisateur@$myhostname")
myorigin = $mydomain        (envoie le courrier comme "utilisateur@$mydomain")
```

10.3 – De quels domaines recevoir le courrier localement

Ensuite, vous devez indiquer quelles adresses Postfix doit livrer localement.

Quelques exemples (utilisez-en un seulement) :

```
/etc/postfix/main.cf:
mydestination = $myhostname, localhost.$mydomain, localhost
mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
mydestination = $myhostname
```

Le premier exemple est approprié pour une station de travail, le second pour le serveur de messagerie d'un domaine entier. Le troisième exemple peut être utilisé sur une interface virtuelle.

10.4 – Adresses des interfaces des proxies/traducteurs d'adresses

Le paramètre `proxy_interfaces` contient toutes les adresses par lesquelles Postfix reçoit du courrier avant traduction par l'éventuel proxy ou traducteur d'adresse (NAT). Vous pouvez utiliser des noms de machines au lieu d'adresses de réseau.

IMPORTANT : vous devez spécifier votre adresse externe de proxy/NAT lorsque votre système est un MX de secours pour d'autres domaines, sinon la livraison des messages bouclera lorsque le serveur MX primaire est inactif.

Exemple : machine MX de secours derrière un NAT.

```
/etc/postfix/main.cf:
proxy_interfaces = 1.2.3.4 (adresse de réseau externe du proxy/NAT)
```

10.5 – De quels clients locaux relayer le courrier

Si votre machine est sur un réseau ouvert, vous devez alors indiquer quelles adresses IP sont autorisées à relayer leur courrier à destination d'Internet par votre machine. La valeur par défaut inclut tous les sous-réseaux sur lesquels est connectée votre machine. Ceci peut donner le droit de relais à trop de clients.

Mes paramètres sont :

```
/etc/postfix/main.cf:  
  mynetworks = 168.100.189.0/28, 127.0.0.0/8
```

10.6 – Quelles destinations relayées accepter pour les clients étrangers

Si votre machine est sur un réseau ouvert vous devez également spécifier comment Postfix transférera le courrier des étrangers. La valeur par défaut acceptera de relayer le courrier de tous les domaines (et les sous-domaines) listés dans \$mydestination. Ceci peut accorder trop de droits de relais. Les valeurs recommandées sont (utilisez-en une seulement) :

```
/etc/postfix/main.cf:  
  relay_domains = (aucun relais d'étranger)  
  relay_domains = $mydomain (mes domaines et sous-domaines)  
  relay_domains = $mydomain, autre.domaine.tld, ...
```

10.7 – Optionel: configurer une machine pour la livraison extérieure

Si vous êtes derrière un firewall, vous devez indiquer un relayhost. Si vous le pouvez, indiquez le nom de domaine ainsi Postfix peut consulter le DNS et peut basculer vers un MX secondaire lorsque le premier est inactif. Sinon, indiquez juste un non d'hôte en sûr.

Quelques exemples (utilisez-en un seulement) :

```
/etc/postfix/main.cf:  
  relayhost = $mydomain  
  relayhost = [mail.$mydomain]
```

La forme encadrée par [] élimine les consultations MX du DNS.

Par défaut, le client SMTP effectue des requêtes DNS même si vous indiquez une machine relais. Si votre machine n'a pas d'accès au serveur DNS, désactivez les consultations DNS du client SMTP comme suit :

```
/etc/postfix/main.cf:  
  disable_dns_lookups = yes
```

Le fichier STANDARD CONFIGURATION README contient plus d'exemples pour les réseaux protégés par firewall et/ou derrière un modem.

10.8 – Créez la base de données d'alias

Postfix utilise une table d'accès compatible Sendmail pour rediriger le courrier des destinataires locaux. Généralement, ces informations sont conservées dans deux fichiers : un fichier texte /etc/aliases et un fichier indexé /etc/aliases.db. La commande "postconf alias_maps" vous donnera l'emplacement exact du fichier texte.

Premièrement, assurez-vous de mettre à jour le fichier texte avec les alias pour root, postmaster et "postfix" qui transfèrent le courrier vers des personnes physiques. Postfix dispose d'un fichier d'exemple /etc/postfix/aliases que vous pouvez adapter.

```
/etc/aliases :
```

```
root: you
postmaster: root
postfix: root
bin: root
etcetera
```

NOTE : il ne doit pas y avoir d'espace avant ":".

Finalement, construisez la base de données des alias avec une des commandes suivantes :

```
# newaliases
# sendmail -bi
```

11 – Mettre en cage chroot ou non

Les processus démons de Postfix peuvent être configurés (via master.cf) pour fonctionner en cage chroot. Le processus tourne avec des privilèges limités et avec un accès restreint au répertoire des files d'attente de Postfix (/var/spool/postfix). Ceci fournit une barrière significative contre les intrusions. La barrière n'est pas imperméable mais difficile à franchir.

À l'exception des démons de Postfix qui livrent le courrier localement et/ou qui exécutent des commandes non-Postfix, tous les démons de Postfix peuvent être mis en cage.

Les sites requérant un haut niveau de sécurité pourront souhaiter mettre en cage tous les démons qui dialoguent sur le réseau : les processus [smtp\(8\)](#) et [smtpd\(8\)](#) et éventuellement le client [lmtp\(8\)](#). Le serveur personnel de l'auteur qui héberge le site porcupine.org lance tous les démons qui peuvent l'être en cage chroot (*Note du traducteur : ainsi que tout serveur Postfix installé avec la distribution Debian par défaut*).

Le fichier /etc/postfix/master.cf par défaut ne met aucun démon de Postfix en cage chroot. Pour activer les opérations de mise en cage chroot éditez le fichier /etc/postfix/master.cf. Les instructions sont dans le fichier.

Notez qu'un démon en cage chroot résout tous les noms de fichiers relativement à la racine de la cage : le répertoire des files d'attente de Postfix (/var/spool/postfix). Pour une utilisation réussie des cages chroot, de nombreux systèmes UNIX nécessitent que vous créiez certains fichiers ou fichiers de périphériques. Le répertoire exemples/chroot-setup des sources contient une collection de scripts qui vous aideront à faire fonctionner Postfix dans un environnement chroot sur différents systèmes d'exploitation.

En complément, vous aurez certainement besoin de configurer syslogd pour qu'il écoute sur une socket dans le répertoire des files d'attente. Quelques exemples pour des systèmes spécifiques :

FreeBSD:

```
# mkdir -p /var/spool/postfix/var/run
# syslogd -l /var/spool/postfix/var/run/log
```

Linux, OpenBSD:

```
# mkdir -p /var/spool/postfix/dev
# syslogd -a /var/spool/postfix/dev/log
```

12 – Soins et alimentation du système Postfix

Les processus démons de Postfix tournent en arrière-plan et enregistrent les problèmes et l'activité normale via le démon syslog. Les noms des fichiers de logs sont indiqués dans /etc/syslog.conf. Au minimum, vous

Documentation de Postfix en français

devrez utiliser quelque chose comme suit :

```
/etc/syslog.conf:
mail.err          /dev/console
mail.debug        /var/log/maillog
```

IMPORTANT : le démon syslogd ne créera pas les fichiers. Vous devrez le faire avant de (re)lancer syslogd.

IMPORTANT : sur Linux vous devez insérer un caractère "-" avant le nom du fichier : -/var/log/maillog, sinon syslogd utilisera plus de ressources système que Postfix.

Heureusement, le nombre de problèmes devrait être faible, mais il est généralement une bonne idée de lancer chaque nuit avant que les fichiers syslog soient tournés :

```
# postfix check
# egrep '(reject|warning|error|fatal|panic):' /un/fichier/de/log
```

- La première ligne (postfix check) indique à Postfix de rapporter les problèmes de propriété/permissions des fichiers.
- La seconde ligne recherche les problèmes à rapporter depuis le logiciel de messagerie y compris les relais courriers indésirables, etc. Vous pourrez appliquer certains processus supplémentaires pour éliminer les informations inintéressantes.

La page [DEBUG_README](#) décrit la signification des mentions "warning" etc. des journaux de Postfix.

Analyse des goulots

d'étranglement

But de ce document

Ce document présente le programme `qshape(1)` qui aide l'administrateur à voir la distribution des messages dans la file d'attente triés par date, expéditeur ou domaine destinataire. `qshape(1)` est fourni avec les sources de Postfix 2.1 source sous le répertoire "auxiliary".

Pour comprendre la sortie de `qshape(1)`, il est préférable de connaître les différentes files d'attente de Postfix. À cette fin, le rôle de chaque répertoire de file d'attente de Postfix est décrite brièvement dans les paragraphes "Info supplémentaire : répertoires de file d'attente de Postfix" à la fin de ce document.

Ce document couvre les sujets suivants :

- [Introduction de l'outil qshape](#)
- [Problèmes avec qshape](#)
- [Exemple 1: File d'attente saine](#)
- [Exemple 2: File d'attente retardée pleine de rejet de messages forgés](#)
- [Exemple 3: Congestion de la file d'attente active](#)
- [Exemple 4: Grand volume de messages retardés](#)
- [Informations d'arrière-plan : répertoires de files d'attente de Postfix](#)
 - ♦ [La file d'attente "maildrop"](#)
 - ♦ [La file d'attente "hold"](#)
 - ♦ [La file d'attente "entrante"](#)
 - ♦ [La file d'attente "active"](#)
 - ♦ [La file d'attente "retardée"](#)
- [Références](#)

Introduction de l'outil qshape

Lorsque le courrier est traité lentement ou la file d'attente est anormalement chargée, lancez `qshape(1)` en tant que super-utilisateur (root) pour faciliter la résolution du problème. Le programme `qshape(1)` montre une vue du contenu de la file d'attente de Postfix sous forme de tableau.

- Sur l'axe horizontal, il montre l'âge des messages avec une granularité plus fine pour les messages récents.
- L'axe vertical est renseigné avec les domaines de destination (ou d'expédition avec l'option "-s"). Les domaines concernés par le plus grand nombre de messages sont listés en premier.

Par exemple sur la sortie ci-dessous, nous voyons les 10 premières lignes des domaines d'expédition (courrier souvent forgé) de la "[file d'attente hold](#) (spam)" :

```
$ qshape -s hold | head
T  5 10 20 40 80 160 320 640 1280 1280+
```


Documentation de Postfix en français

TOTAL	486	0	0	1	0	0	2	4	20	40	419
yahoo.com	14	0	0	1	0	0	0	0	1	0	12
extremepricecuts.net	13	0	0	0	0	0	0	0	2	0	11
ms35.hinet.net	12	0	0	0	0	0	0	0	0	1	11
winnersdaily.net	12	0	0	0	0	0	0	0	2	0	10
hotmail.com	11	0	0	0	0	0	0	0	0	1	10
worldnet.fr	6	0	0	0	0	0	0	0	0	0	6
ms41.hinet.net	6	0	0	0	0	0	0	0	0	0	6
osn.de	5	0	0	0	0	0	1	0	0	0	4

- La colonne "T" montre le total des messages (émis dans ce cas) pour chaque domaine. Les colonnes suivantes montrent le nombre de messages par tranche d'âge. La ligne nommée "TOTAL" montre le total pour tous les domaines.
- Dans cet exemple, il y a 14 messages issus de yahoo.com, 1 datant dont l'âge se situe entre 10 et 20 minutes, 1 entre 320 et 640 minutes, et 12 âgés de plus de 1280 minutes (1 jour vaut 1440 minutes).

Par défaut, qshape montre les statistiques cumulées des files d'attente entrante et active qui sont les plus significatives pour analyser les performances.

On peut faire une requête sur une liste de files d'attente alternative :

```
$ qshape deferred | less
$ qshape incoming active deferred | less
```

qui montrera la distribution des messages de la file d'attente retardée ou le total des files d'attente active et retardée.

Les options de la ligne de commande contrôlent le nombre de lignes affichées, l'âge limite inférieur, le compte par domaine parent, et ainsi de suite. L'option "-h" présente un résumé des options disponibles.

Problèmes avec qshape

Les grands nombres sur la sortie de qshape représentent de grands nombres de messages destinés à (ou provenant de) un domaine particulier. Il devrait être possible de distinguer les domaines dont le nombre de message en attente est majoritairement du courrier entrant ou sortant.

Le problème des domaines expéditeurs ou destinataires apparaît au coin supérieur gauche du tableau de sortie. Souvenez-vous que la file d'attente active ne peut contenir plus de 20000 (\$qmgr message active limit) messages. Pour savoir si cette limite a été atteinte, utilisez :

```
$ qshape -s active | head           (montre les statistiques expéditeurs)
```

Si le compteur d'expédition est en deça de 20000, la file d'attente active n'est pas saturée, le domaine le plus chargé apparaît en haut du tableau.

La file d'attente active est également limitée à 20000 adresses de destination (\$qmgr message recipient limit). Pour connaître le niveau de saturation, utilisez :

```
$ qshape active | head              (montre les statistiques destinataires)
```

En ayant trouvé le domaine le plus chargé, il est souvent intéressant de chercher les logs récents concernant les messages appartenant au domaine en question.

Documentation de Postfix en français

```
# Recherche des livraisons à destination de exemple.com
#
$ tail -10000 /var/log/maillog |
    egrep -i ' : to=<.*@exemple\.com>,' |
    less

# Recherche des messages provenant de exemple.com
#
$ tail -10000 /var/log/maillog |
    egrep -i ' : from=<.*@exemple\.com>,' |
    less
```

Si vous voulez suivre le chemin d'un message particulier (à partir de son id) :

```
# Recherche les logs concernant le message dont l'id est 2B2A73FF68.
#
$ tail -10000 /var/log/maillog | egrep ' : 2B2173FF68: '
```

Observez également les messages d'attention du gestionnaire des files d'attente dans les logs. Ils peuvent suggérer une stratégie pour réduire les congestions.

```
$ egrep 'qmgr.*(panic|fatal|error|warning):' /var/log/maillog
```

Lorsque tous les tests échouent, essayez la liste de diffusion de Postfix pour obtenir de l'aide, mais n'oubliez pas d'inclure les 10 ou 20 premières lignes de la sortie de [qshape\(1\)](#).

Exemple 1: file d'attente saine

Lors de l'examen des files d'attente [entrante](#) et [active](#), dans des conditions normales (pas de congestion), ces files sont quasiment vides. Le courrier est expédié quasiment instantanément ou est retardé sans congestion dans la [file d'attente active](#).

```
$ qshape          (show incoming and file d'attente active status)

                T  5 10 20 40 80 160 320 640 1280 1280+
TOTAL          5  0  0  0  1  0  0  0  1  1  2
meri.uwasa.fi  5  0  0  0  1  0  0  0  1  1  2
```

Si quelqu'un examine ces files séparément, la [file d'attente entrante](#) est vide ou contient ponctuellement un ou deux messages, alors que la [file d'attente active](#) contient plus de messages, âgés pour certains :

```
$ qshape incoming

                T  5 10 20 40 80 160 320 640 1280 1280+
TOTAL          0  0  0  0  0  0  0  0  0  0  0

$ qshape active

                T  5 10 20 40 80 160 320 640 1280 1280+
TOTAL          5  0  0  0  1  0  0  0  1  1  2
meri.uwasa.fi  5  0  0  0  1  0  0  0  1  1  2
```

Exemple 2: File d'attente retardée pleine de rejet de messages forgés

Documentation de Postfix en français

Ceci provient d'un serveur où la validation des destinataires n'est pas disponible pour certains des domaines hébergés. Les attaques par dictionnaire sur le domaine non contrôlé engendre de nombreux messages de rejet. Ils surchargent la file d'attente, mais avec de bon réglages, ils ne saturent pas les files d'attente entrante ou active. Le très grand volume de messages retardés n'est pas une cause directe d'alarme.

```
$ qshape deferred | head
```

	T	5	10	20	40	80	160	320	640	1280	1280+
TOTAL	2234	4	2	5	9	31	57	108	201	464	1353
heyhihellothere.com	207	0	0	1	1	6	6	8	25	68	92
pleazerzoneprod.com	105	0	0	0	0	0	0	0	5	44	56
groups.msn.com	63	2	1	2	4	4	14	14	14	8	0
orion.toppoint.de	49	0	0	0	1	0	2	4	3	16	23
kali.com.cn	46	0	0	0	0	1	0	2	6	12	25
meri.uwasa.fi	44	0	0	0	0	1	0	2	8	11	22
gjr.paknet.com.pk	43	1	0	0	1	1	3	3	6	12	16
aristotle.algonet.se	41	0	0	0	0	0	1	2	11	12	15

Les domaines montrés sont de fréquents véhicules de spam et le plus grand volume se trouve dans les colonnes correspondant aux messages les plus âgés, montrant ainsi que les taux d'arrivée sont modérés. Les grands nombres sur des messages d'âge faible sont plus significatifs de problèmes. Les vieux messages n'allant nul part sont d'autant moins gênant que les files d'attente active et entrante sont faiblement chargées. Nous pouvons également voir que les messages non-livrables groups.msn.com sont arrivent avec un taux si faible qu'il ne s'agit pas d'une attaque par dictionnaire.

```
$ qshape -s deferred | head
```

	T	5	10	20	40	80	160	320	640	1280	1280+
TOTAL	2193	4	4	5	8	33	56	104	205	465	1309
MAILER-DAEMON	1709	4	4	5	8	33	55	101	198	452	849
exemple.com	263	0	0	0	0	0	0	0	0	2	261
exemple.org	209	0	0	0	0	0	1	3	6	11	188
exemple.net	6	0	0	0	0	0	0	0	0	0	6
exemple.edu	3	0	0	0	0	0	0	0	0	0	3
exemple.gov	2	0	0	0	0	0	0	0	1	0	1
exemple.mil	1	0	0	0	0	0	0	0	0	0	1

En regardant la distribution des expéditeurs, nous voyons que la plupart des messages sont des rejets.

Exemple 3: Congestion de la file d'attente active

Cet exemple est pris sur une discussion de février 2004 sur la liste de diffusion des utilisateurs de Postfix. La congestion a été rapportée avec de grandes files d'attente active et entrante et des nombres de processus d'agent de livraison proches des limites. Le fil de cette discussion est archivé ici :

<http://groups.google.com/groups?th=636626c645f5bbde>

En utilisant une version ancienne de qshape(1) on voyait seulement les messages de quelques destinations :

```
$ qshape (montre les statuts des files d'attente entrante et active)
```

	T	A	5	10	20	40	80	160	320	320+
TOTAL	11775	9996	0	0	1	1	42	94	221	1420
user.sourceforge.net	7678	7678	0	0	0	0	0	0	0	0
lists.sourceforge.net	2313	2313	0	0	0	0	0	0	0	0
gzd.gotdns.com	102	0	0	0	0	0	0	0	2	100

La colonne "A" montre le nombre de messages de la file d'attente active, et les colonnes numérotées montrent les totaux pour la file d'attente retardée. À 10000 messages (taille limite de la file d'attente active de Postfix 1.x) la file d'attente active est pleine. La file entrante se chargeait rapidement.

Avec les destinations à problème clairement identifiées, l'administrateur a rapidement trouvé et solutionné le problème. Il est nettement plus difficile d'obtenir ces mêmes informations des logs. Bien que une lecture attentive de la sortie de mailq(1) donnerait des résultats similaires, il est plus difficile de jauger l'ampleur du problème en regardant les messages en file d'attente.

Exemple 4: Grand volume de messages retardés

Lorsqu'un site à qui vous envoyés beaucoup de messages est arrêté ou lent, les messages encombrant rapidement la file d'attente retardée, ou pire, la file d'attente active. La sortie de `qshape` montrera de grands nombres pour les domaines destinataires de tous les âges à partir du début du problème :

```
$ qshape deferred | head
      T    5   10   20   40   80  160  320  640 1280 1280+
TOTAL 5000 200 200 400 800 1600 1000 200 200 200 200
highvolume.com 4000 160 160 320 640 1280 1440 0 0 0 0
...
```

Ici, la destination "highvolume.com" continue d'accumuler du courrier retardé. Les files d'attente entrante et active sont faiblement chargées, mais la file d'attente retardée a commencé à se charger 1 à 2 heures plus tôt et continue à grandir.

Si la destination en cause n'est pas arrêtée, mais seulement ralentie, on peut voir une congestion similaire dans la file d'attente active. Ce dernier type de congestion est plus alarmant ; on doit prendre des mesures pour s'assurer que le courrier soit retardé ou ajouter une règle d'accès demandant aux expéditeurs d'essayer d'envoyer leur courrier ultérieurement.

Si une destination à grand volume montre souvent ces signes suite à des connexions refusées par toutes ses machines MX ou avec la mention "421 Server busy errors", il est possible que le gestionnaire des files d'attente marque cette destination comme "morte" en dépit de la nature des erreurs. La destination sera réessayée plus tard après expiration du délai \$minimal_backoff_time. Si les erreurs sont assez fréquentes, il peut n'y avoir que peu de messages livrés avant que la destination ne soit marquée "morte".

Le MTA qui a été observé le plus fréquemment présentant ces signes est Microsoft Exchange, qui refuse les connexions sous la charge. Certains scanners-proxies antivirus en coupure du serveur Exchange propagent les connexions refusées au client avec le code "421" error.

Notez qu'il est désormais possible de configurer Postfix pour présenter des caractéristiques similaires en configurant mal le serveur anvil(8) (non inclus dans Postfix 2.1). N'utilisez pas anvil(8) pour limiter les taux, son but est la prévention des DoS (dénis de service) et la limite du taux doit être très généreuse !

À long terme, il est souhaité que la détection des machines mortes de Postfix et le mécanisme de contrôle de concurrence soit plus tolérant aux "nuisibles". Pour ceux qui doivent livrer un volume important de courrier vers une destination qui montre souvent ce genre d'erreurs, il y a un subtil contournement.

- Dans `master.cf` créez un clone du transporteur "smtp" dédié pour la destination en question.
- Dans `master.cf` configurez une limite de processus raisonnable pour ce transporteur (typiquement un

nombre compris entre 10 et 20).

- **IMPORTANT!!!** Dans `main.cf` configurez une très large limite de concurrence initiale vers cette destination (200).

```
/etc/postfix/main.cf:  
    initial_destination_concurrency = 200  
    transportname_destination_concurrency_limit = 200
```

Où *transportname* est le nom de l'entrée `master.cf` en question.

La conséquence de cette configuration surprenante est que 200 erreurs consécutives sont tolérées sans marquer cette destination comme morte tout en maintenant une concurrence raisonnable (10 à 20 processus). Cette astuce doit être réservée à des situations particulières : volume élevé de livraisons dans un canal capable de recevoir un grand nombre de messages mais régulièrement perturbé par des erreurs.

Lorsqu'une destination est incapable de tenir la charge même après que la limite des processus Postfix ait été réduite à 1, en désespoir on peut insérer un bref délai entre deux tentatives de livraison.

- Dans l'entrée de la table `transport` correspondant à la destination à problème, indiquez une machine morte comme premier saut.
- Dans l'entrée du fichier `master.cf` correspondant au transporteur, indiquez la destination à problème comme `fallback_relay` et indiquez une petite valeur `smtp_connect_timeout`.

```
/etc/postfix/transport:  
    problem.exemple.com    slow:[dead.host]  
  
/etc/postfix/master.cf:  
# service type  private unpriv  chroot  wakeup  maxproc command  
slow    unix    -         -        n        -        1      smtp  
-o fallback_relay=problem.exemple.com  
-o smtp_connect_timeout=1
```

Cette solution oblige le client `smtp(8)` de Postfix à attendre `$smtp_connect_timeout` secondes entre les livraisons. Cette solution dépend des détails de management des connexions, et doit être revue lorsque le cache des connexions SMTP est introduit.

Heureusement, une solution plus élégante à ces problèmes sera trouvée dans le futur.

Information d'arrière-plan : répertoires de files d'attente de Postfix

Les paragraphes suivants décrivent les files d'attente de Postfix : leur but, leur aspect normal et comment diagnostiquer les situations anormales.

La file d'attente "maildrop"

Les messages soumis via la commande `sendmail(1)` de Postfix mais pas encore déposés dans la file d'attente principale de Postfix par le service `pickup(8)`, attendent dans la file d'attente "`maildrop`". Des messages peuvent être ajoutés à la file d'attente "`maildrop`" même lorsque le système Postfix ne fonctionne pas. Ils seront pris en charge au démarrage de Postfix.

La file d'attente "maildrop" n'est gérée que par le service mono-thread pickup(8) qui l'examine périodiquement ou lorsque une notification d'arrivée est transmise par le programme postdrop(1). Le programme postdrop(1) est un programme d'aide setgid qui permet à un utilisateur non privilégié d'utiliser le programme sendmail(1) de Postfix pour injecter du courrier dans la file d'attente "maildrop" et notifier cette arrivée au service pickup(8).

Tous les messages entrant dans la file d'attente principale de Postfix passent par le service cleanup(8). Ce service est responsable de la réécriture de l'enveloppe et des en-têtes, de l'examen par expressions rationnelles des en-têtes et du corps, de l'ajout automatique des destinataires cachés (BCC) et garantit l'insertion du message dans la "file d'attente entrante" de Postfix.

En l'absence de consommation excessive de CPU lors de l'examen des en-têtes et du corps par cleanup(8) ou par un autre logiciel, la performance est de l'ordre de la capacité d'entrées/sortie du disque. Le taux auquel le service pickup(8) peut injecter du courrier dans la file d'attente est grandement déterminé par les temps d'accès au disque jusqu'à ce que le service cleanup(8) valide le stockage du message avant de retourner un code de succès. C'est également vrai pour les écritures de messages dans le répertoire "maildrop" par le programme postdrop(1).

Comme le service pickup est mono-thread, il ne peut livrer qu'un message à la fois à un rythme qui ne peut excéder les capacités d'entrées/sorties du disque (+ CPU si non négligeable) du service cleanup.

Une congestion dans cette file d'attente est significative d'une soumission locale excessive de messages ou peut-être d'une consommation excessive de CPU par le service cleanup(8) due à une table body_checks excessive.

Notez que si la file d'attente active est pleine, le service cleanup tentera de ralentir l'injection de message en attendant \$in_flow_delay secondes avant chaque message. Dans ce cas, la congestion de la file d'attente "maildrop" congestion risque d'être une conséquence de la congestion en aval en non un problème en soi.

Notez également qu'on ne doit pas tenter de livrer un grand volume de messages via le service pickup(8). Les sites à fort trafic doivent éviter les filtres de contenu qui réinjectent le courrier examiné via les commandes Postfix sendmail(1) et postdrop(1).

Un fort taux d'arrivée de messages soumis localement peut être une indication de boucles de message ou d'un programme de notifications emballé. Essayez de garder le volume de messages injectés localement à un niveau modéré.

La commande "postsuper -r" peut placer des messages sélectionnés dans la file d'attente "maildrop" pour réeffectuer le processus. C'est commode pour remettre à jour les paramètres de filtrage de contenu content filter. Remettre en file d'attente un grand nombre de messages en utilisant "postsuper -r" peut clairement causer un pic de taille dans la file d'attente "maildrop".

La file d'attente "hold"

L'administrateur peut définir des politiques d'accès(5) "smtpd", ou des examens d'en-têtes/du corps par le service cleanup(8) transférant automatiquement les messages de la file d'attente normale vers la file d'attente "hold". Les messages placés dans la file d'attente "hold" y restent jusqu'à intervention de l'administrateur. Aucune tentative de livraison périodique n'a lieu dans la file d'attente "hold". La commande postsuper(1) peut être utilisée manuellement pour renvoyer ces messages dans la file d'attente "retardée".

Des messages peuvent potentiellement rester dans la file d'attente "hold" pour un temps supérieur à l'espérance de vie maximale en file d'attente (après laquelle ils sont renvoyés à l'expéditeur). Si de tels "vieux" messages doivent être enlevés de la file d'attente "hold", ils doivent généralement être déplacés dans la file d'attente "maildrop", ainsi ils récupèrent une nouvelle marque de temps et ont plus de chances d'être livrés. Les messages "jeunes" peuvent être déplacés directement dans la file d'attente "retardée".

La file d'attente "hold" joue un petit rôle dans les performances de Postfix, car son but est plus de traquer le spam que d'améliorer les performances.

La file d'attente "entrante" (incoming)

Tous les nouveaux messages entrant dans Postfix sont écrits par le service cleanup(8) dans la file d'attente "entrante". Les nouveaux fichiers sont créés par l'utilisateur "postfix" avec des droits d'accès 0600. Lorsqu'un fichier est prêt à être traité, le service cleanup(8) change ces droits en 0700 et notifie l'arrivée au gestionnaire des files d'attente. Le gestionnaire des files d'attente ignore les fichiers incomplets dont les droits sont positionnés à 0600, considérant qu'ils sont encore traités par cleanup.

Le gestionnaire des files d'attente examine la file d'attente entrante envoyant tout nouveau message dans la file d'attente "active" si la limite de celle-ci n'est pas dépassée. Par défaut, la file d'attente active peut contenir 20000 messages. Une fois la limite de la file d'attente active atteinte, le gestionnaire des files d'attente arrête d'examiner la file entrante (ainsi que la file retardée, voir ci-dessous).

Dans des conditions normales, la file d'attente entrante est quasiment vide (elle n'a que des fichiers en mode 0600), et le gestionnaire des files d'attente peut importer les nouveaux messages dans la file d'attente active au fur et à mesure qu'ils arrivent.

La file d'attente entrante augmente dès que le taux de messages entrant dépasse le taux auquel le gestionnaire des files d'attente peut importer les messages dans la file d'attente active. Le principal facteur de ralentissement du gestionnaire des files d'attente est du aux requêtes de transport effectuées auprès du service trivial-rewrite. Si le gestionnaire des files d'attente est ordinairement chargé, évitez les services de consultation "lents" (MySQL, LDAP, ...) pour la recherche du transporteur ou accélérez les machines qui fournissent ce service.

Le paramètre in_flow_delay est utilisé pour limiter le taux d'arrivée lorsque le gestionnaire des files d'attente commence à faiblir. Le service cleanup(8) attendra \$in_flow_delay secondes avant de créer un nouveau fichier en file d'attente s'il ne peut obtenir de "jeton" du gestionnaire des files d'attente.

Depuis que le nombre de processus cleanup(8) est limité dans la plupart des cas par la concurrence des serveurs SMTP, le taux d'entrée peut excéder le taux de sortie de plus de "nb de connexions SMTP" / \$in_flow_delay messages par secondes.

Avec une limite de processus à 100 et un in_flow_delay de 1 seconde, le couple est assez fort pour limiter un injecteur simple à 1 message par seconde, mais pas assez pour infléchir un taux d'entrée excessif provenant de plusieurs sources au même moment.

Si un serveur est attaqué depuis plusieurs directions, augmentez in_flow_delay à 10 secondes seulement si la file d'attente entrante augmente alors même que la file d'attente active n'est pas remplie et que le service trivial-rewrite utilise un mécanisme de recherche du transporteur rapide.

La file d'attente "active"

Le gestionnaire des files d'attente est un ordonnaceur d'agent de livraison ; il travaille à assurer une livraison rapide et sûre des messages vers toutes les destinations avec les limites des ressources.

La file d'attente active est assez analogue à une file d'attente d'un processus fonctionnant dans un système d'exploitation. Les messages dans la file d'attente active sont prêts à être envoyés (runnable), mais ne sont pas nécessairement sur le point d'être envoyé (running).

Alors que beaucoup d'administrateurs Postfix pensent que la file d'attente "active" est un répertoire sur disque, la réelle file d'attente "active" est un ensemble de structures de données dans la mémoire du processus gestionnaire des files d'attente.

Les messages dans les files d'attente "maildrop", "hold", "incoming" et "retardée" (ci-dessous) n'occupent pas la mémoire ; ils sont stockés en sécurité sur le disque attendant leur tour. Les informations d'enveloppe des messages en file d'attente "active" sont gérées en mémoire, autorisant le gestionnaire des files d'attente à ordonnancer globalement, allouer les processus agent de livraison disponibles au message approprié de la file d'attente active.

Dans la file d'attente active, les messages (multi-destinataires) sont coupés en groupes de destinataires qui partagent la même combinaison transport/saut-suivant ; la taille du groupe est limitée par la limite de concurrence de destinataires du transporteur.

Les groupes de destinataires multiples (d'un ou plusieurs messages) sont mis en file d'attente pour la livraison via la combinaison transport/saut-suivant. La limite de concurrence de destination du transporteur désigne le nombre de tentatives de livraison simultanées pour chaque saut suivant. Les transporteurs dont la limite de destinataires concurrents est fixée à 1 sont spéciaux : ils sont groupés par adresse de destination actuelle plutôt que par saut suivant, activant ainsi les limites de concurrence par destinataire plutôt que par domaine. Les limites par destinataire sont plus appropriées à la livraison finale aux boîtes-aux-lettres qu'au transfert à un autre serveur.

Des congestions apparaissent dans la file d'attente active lorsqu'une ou plusieurs destinations reçoivent à un taux plus lent que le taux de message entrant correspondant. Si une destination est "morte" pendant un certain temps, le gestionnaire des files d'attente la marquera comme tel, et retardera immédiatement tous les messages pour cette destination sans essayer de les assigner à un agent de livraison. Dans ce cas les messages seront rapidement retirés de la file d'attente active et mis dans la file retardée. Si la destination est simplement ralentie, ou s'il y a un problème dû à un taux d'arrivée trop élevée, la file d'attente active grossira et sera dominée par les messages dont la destination est congestionnée.

La seule manière de réduire la congestion est soit de réduire le taux d'entrée soit d'augmenter le taux de sortie. Cette dernière proposition se traduit soit par une augmentation de la concurrence soit par la réduction de la latence de livraison.

Pour des sites à fort volume, un paramètre clef est le nombre d'agents de livraisons "smtp" autorisés pour les transports "smtp" et "relay". Ces sites ont tendance à envoyer à beaucoup de destinations dont beaucoup peuvent être mortes ou ralenties, donc une bonne fraction des agents de livraison disponibles seront bloqués attendant ces sites ralentis. Ainsi le courrier destiné au monde entier engendrera des latences, ainsi le seul moins d'augmenter la capacité de sortie est d'augmenter la concurrence des agents de livraisons.

La limite par défaut des processus "smtp" fixée à 100 est assez élevée pour beaucoup de sites et peut être

abaissée pour les sites disposant d'une faible bande passante (à n'utiliser que si le réseau est saturé). Lorsqu'on trouve que la file d'attente croît sur un système "peu actif" (CPU, I/O disque et réseau non saturés) la raison résultante de la congestion est une insuffisante concurrence face à la charge. Si le nombre de connexions sortantes SMTP (en état ESTABLISHED ou SYN_SENT) atteint la limite des processus, que le courrier est traité lentement et que le système et le réseau ne sont pas chargés, augmentez la limite des processus "smtp" et/ou "relay" !

En particulier pour le transporteur "relay", diminuez le temps limite des connexions SMTP (1 à 5 secondes) et augmentez la limite par défaut de concurrence par destination. Prenez en compte la latence en question lorsque 1 sur N des machines MX est hors service pour un site à fort volume et assurez vous que la concurrence configurée divisée par cette latence dépasse le taux de messages requis. Si vous gérez la destination, utilisez des répartiteurs de charge devant un groupe de machines MX. Ces équipements ont un meilleur MTBF et peuvent masquer les arrêts individuels de serveurs MX.

Si nécessaire, dédiez et optimisez un transporteur personnalisé pour ces destinations à fort volume.

Une autre cause commune de congestion est un vidage non garanti de la file d'attente retardée entière. La file d'attente retardée contient les messages dont la première tentative de livraison a échoué et risquent de ralentir la livraison (timeouts). Ceci signifie que la réaction la plus commune face une grosse file d'attente retardée (flush it!) est contre-productive et peut aggraver le problème. Ne videz pas la file d'attente retardée sauf si vous pensez que la majeure partie de son contenu est récemment devenu livrable (c'est à dire par exemple que relayhost est de nouveau disponible après un plantage) !

Notez que bien que le gestionnaire des files d'attente est été redémarré, il peut y avoir des messages dans le répertoire de la file d'attente active, mais la file d'attente active "réelle" en mémoire est vide. Pour rétablir l'état en mémoire, le gestionnaire des files d'attente déplace tous les messages de la file d'attente active dans la file d'attente entrante, puis utilise l'examen normal de la file entrante pour reconstituer la file d'attente active. Le processus de déplacement de tous les messages, de consultation des tables de transport (service de résolution trivial-rewrite(8)), et de ré-importation des messages en mémoire est cher. Évitez donc les redémarrages fréquents du gestionnaire des files d'attente.

La file d'attente "retardée" (deferred)

Lorsque tous les destinataires livrables d'un message sont livrés et que la livraison a échoué pour une raison temporaire (la livraison est susceptible de réussir plus tard), le message est placé dans la file d'attente retardée.

Le gestionnaire des files d'attente examine la file d'attente retardée périodiquement. L'intervalle d'examen est contrôlé par le paramètre queue_run_delay. Pendant qu'un examen de la file d'attente retardée est en cours, si un examen de la file d'attente entrante est également en cours (idéalement ils sont brefs tant que la file entrante est petite), le gestionnaire des files d'attente alterne entre le transfert des nouveaux messages entrants et celui des nouveaux messages "retardés". Cette stratégie "round-robin" évite l'encombrement des files d'attentes entrante et retardée.

Chaque examen de la file d'attente retardée ne transfère qu'une partie des messages vers la file d'attente active pour une nouvelle tentative. En effet, chaque message en file d'attente retardée se voit assigné un temps "cool-off" pendant lequel il est retardé. Ceci est fait en datant la modification du fichier dans le futur. Le fichier en file d'attente n'est pas éligible pour une nouvelle tentative avant que cette date soit atteinte.

Ce délai "cool-off" est compris entre \$minimal_backoff_time et \$maximal_backoff_time. La date de tentative suivante est calculée en doublant l'âge du message dans la file d'attente, et ajustée pour s'adapter aux limites.

Ceci signifie que les messages jeunes sont initialement représentés plus souvent que les anciens.

Si des sites à grand volume ont couramment de grandes files d'attente retardées, il peut être facile d'ajuster les délais queue_run_delay, minimal_backoff_time et maximal_backoff_time pour fournir des délais assez court au premier échec, avec peut-être des délais plus longs pour les échecs multiples, pour réduire le taux de retransmission des vieux messages et réduire ainsi la quantité de messages précédemment retardés dans la file d'attente active.

Une cause commune des grandes files d'attente retardées est l'échec de validation des destinataires à l'entrée des messages SMTP. Comme les spammers lancent couramment des attaques par dictionnaire depuis des adresses d'expéditions non livrables, les avis de rejets à destination de ces adresses de destination invalides encombrant la file d'attente retardée (et en proportion la file d'attente active). La validation des destinataires est vivement recommandée en utilisant les paramètres local_recipient_maps et relay_recipient_maps.

Lorsqu'une machine avec beaucoup de messages retardés est arrêté longtemps, il est possible que la file d'attente retardée entière retente simultanément la livraison. Ceci peut engorger la file d'attente active. Ce phénomène peut être répété approximativement toutes les maximal_backoff_time secondes si les messages sont de nouveau retardés après une brève période de congestion. Idéalement, un Postfix futur ajoutera un délai supplémentaire aléatoire au délai avant tentative (ou utiliser une combinaison de stratégies) pour réduire les chances de répéter ce phénomène.

Références

Le programme qshape(1) a été développé par Victor Duchovni de Morgan Stanley, qui a également écrit la première version de ce document.

Optimisation des performances de

Postfix

But de l'optimisation des performances de Postfix

Les trucs et astuces de ce document vous aideront à améliorer les performances des systèmes Postfix qui fonctionnent déjà. Si votre système Postfix n'est pas capable de recevoir et livrer du courrier, vous devez d'abord résoudre ces problèmes en utilisant au besoin la page [DEBUG README](#).

Pour optimiser les performances d'un filtre externe de contenu, consultez d'abord les pages [FILTER README](#) et [SMTPD PROXY README](#). Assurez-vous ensuite que le code du filtre de contenu ne comporte pas de latence. Autant que possible, évitez le recours aux sources de données externes ajoutant un délai non négligeable. Votre filtre de contenu doit fonctionner avec une faible concurrence pour éviter les surcharges de CPU/mémoire. Les sites à fort trafic devront éviter les consultations RBL, requêtes aux bases de données complexes et ainsi de suite.

Éléments de performance de la réception du courrier :

- [Éléments généraux de performance de la réception](#)
- [Faire plus de travail avec vos processus serveurs SMTP](#)
- [Ralentir les clients SMTP qui commettent beaucoup d'erreurs](#)
- [Mesures contre les clients qui ouvrent trop de connexions](#)

Éléments de performance de la livraison du courrier :

- [Éléments généraux de performance de la livraison](#)
- [Optimiser la fréquence de tentatives de livraison du courrier retardé](#)
- [Optimiser le nombre de livraisons simultanées](#)
- [Optimiser le nombre de destinataires par livraison](#)

Autres éléments de performance :

- [Optimiser le nombre de processus Postfix](#)
- [Optimiser le nombre de fichiers ou de sockets ouverts](#)

Les outils suivants peuvent être utilisés pour mesurer les performances du système de messagerie sous une charge artificielle. Ils ne sont normalement pas installés avec Postfix.

- [smtp-source, générateur de messages SMTP/LMTP](#)
- [smtp-sink, duplicateur de messages SMTP/LMTP](#)
- [qmqp-source, générateur de messages QMOP](#)
- [qmqp-sink, duplicateur de messages QMOP](#)

Éléments généraux de performance de la reception

- Avant de commencer, vous devez avoir compris les notions de file d'attente maildrop, file d'attente entrante, et file d'attente active présentées à la page [QSHAPE README](#).
- Lancez un serveur DNS local pour réduire le temps de latence des consultations DNS. Si vous utilisez de multiples systèmes Postfix, faites pointer chaque serveur DNS local sur le même serveur partagé pour réduire le nombre de consultations au travers du réseau.
- Éliminez les consultations LDAP non nécessaires, en indiquant un filtre de domaine. Ceci élimine les consultation pour des adresses extérieures, et élimine les consultations d'adresses partielles. Lisez [ldap_table\(5\)](#) pour plus de détails.

Lorsque Postfix répond lentement aux clients SMTP :

- Recherchez les symptômes de problèmes tel que décrit à la page [DEBUG README](#) et éliminez d'abord ces problèmes.
- Désactivez les examens header checks et body checks et regardez si le problème persiste.
- Désactivez la mise en cage chroot tel que décrit à la page [DEBUG README](#) et regardez si le problème persiste.
- Si Postfix enregistre le client SMTP comme "unknown" dans les journaux, vous avez un problème de service DNS : le serveur n'est pas le bon ou le fichier `resolv.conf` contient de mauvaises informations, ou encore un pare-feu bloque les requêtes ou les réponses DNS.
- Si le nombre de processus [smtpd\(8\)](#) atteint la limite autorisée indiquée dans le fichier `master.cf`, les nouveaux clients SMTP doivent attendre qu'un processus se libère. Augmentez la limite si la mémoire le permet. Lisez les instructions données au paragraphe "[Optimiser le nombre de processus Postfix](#)".

Faire plus de travail avec vos processus serveurs SMTP

Avec les versions 2.0 et antérieures de Postfix, le serveur [smtpd\(8\)](#) attend avant de rapporter une erreur à un client SMTP. Cette idée est appelée "tar pitting". Toutefois, ces délais ralentissent également Postfix. Lorsque le serveur [smtpd\(8\)](#) répond lentement, les sessions prennent plus de temps et ainsi plus de processus [smtpd\(8\)](#) sont nécessaires pour absorber la charge. Lorsque la limite des processus serveur [smtpd\(8\)](#) de Postfix est atteinte, les nouveaux clients doivent attendre qu'un processus se libère ralentissant ainsi tous les clients.

Vous pouvez accélérer accélérer le traitement des erreurs du serveur [smtpd\(8\)](#) en désactivant ce délai :

```
/etc/postfix/main.cf:
# Pas nécessaire avec Postfix >= 2.1
smtpd_error_sleep_time = 0
```

En renseignant ainsi le paramètre, les versions 2.0 et antérieures de Postfix peuvent servir plus de clients avec le même nombre de processus serveur SMTP. Le paragraphe suivant décrit comment Postfix traite les clients qui commettent beaucoup d'erreurs.

Ralentir les clients SMTP qui commettent beaucoup d'erreurs

Le serveur [smtpd\(8\)](#) de Postfix maintient un compteur d'erreurs par session. Celui-ci est remis à zéro lorsqu'un message est correctement transféré et est incrémenté lorsqu'une requête client n'est pas reconnue ou non implémentée, lorsqu'une requête viole une restriction d'accès, ou lorsque d'autres erreurs apparaissent.

Au fur et à mesure que le compteur d'erreurs par session croît, le serveur smtpd(8) change de comportement et commence à insérer des délais avant les réponses. L'idée est de ralentir ces clients pour limiter l'emploi des ressources. Ce comportement dépend des versions de Postfix.

IMPORTANT : Ces délais ralentissent également Postfix. Lorsqu'un trop fort délai est configuré, le nombre de sessions SMTP simultanées croît jusqu'à atteindre la limite des processus smtpd(8) et les nouveaux clients SMTP doivent attendre qu'un serveur smtpd(8) se libère.

Postfix version 2.1 et supérieures :

- Lorsque le compteur d'erreurs atteint \$smtpd_soft_error_limit (défaut : 10), le serveur smtpd(8) ralentit toutes les réponses consécutives ou non à une erreur de \$smtpd_error_sleep_time secondes (défaut : 1 seconde).
- Lorsque le compteur d'erreur atteint \$smtpd_hard_error_limit (défaut : 20) le serveur smtpd(8) de Postfix coupe la connexion.

Postfix version 2.0 et antérieures :

- Lorsque le compteur d'erreur est inférieur à \$smtpd_soft_error_limit (défaut : 10), le serveur smtpd(8) de Postfix patiente \$smtpd_error_sleep_time secondes avant chaque réponse consécutive à une erreur (1 seconde avec Postfix 2.0, 5 secondes avec Postfix 1.1 et antérieurs).
- Lorsque le compteur d'erreurs atteint \$smtpd_soft_error_limit, le serveur smtpd(8) de Postfix retarde toutes les réponses de "nombre d'erreurs" secondes ou \$smtpd_error_sleep_time, s'il est plus élevé.
- Lorsque le compteur d'erreurs atteint \$smtpd_hard_error_limit (défaut : 20) le serveur smtpd(8) de Postfix coupe la connexion.

Mesures contre les clients qui ouvrent trop de connexions

Note : cette fonctionnalité n'est pas inclus dans Postfix version 2.1.

Le serveur smtpd(8) de Postfix peut limiter le nombre de connexions simultanées venant du même client SMTP, ainsi que le nombre de connexions que le client est autorisé à faire par unité de temps. Ces statistiques sont maintenues par le serveur anvil(8) (explication : si anvil(8) s'arrête, les limites de connexions ne sont plus comptées).

IMPORTANT : Ces limites sont conçues pour protéger le serveur smtpd(8) contre les abus flagrants. N'utilisez pas ces limites pour réguler le trafic légitime du courrier : le courrier sera grossièrement ralenti.

- Un client SMTP peut effectuer \$smtpd_client_connection_count_limit connexions simultanées (défaut : 50). Ceci est la moitié de la limite par défaut des processus.
- Un client SMTP peut livrer \$smtpd_client_message_rate_limit messages par unité de temps (défaut : pas de limites).
- Un client SMTP peut adresser \$smtpd_client_recipient_rate_limit destinataires par unité de temps (défaut : pas de limites).
- Un client SMTP peut effectuer \$smtpd_client_connection_rate_limit connexions par unité de temps (défaut : pas de limites).
- Ces limites ne sont pas appliquées aux clients SMTP des réseaux indiqués par \$smtpd_client_connection_limit_exceptions (défaut : les clients des réseaux \$mynetworks peuvent faire un nombre illimité de connexions).

- Le paramètre anvil_rate_time_unit indique l'unité de temps pendant laquelle les taux de connexions client sont évalués (défaut : 60s).

Éléments généraux de performance de la livraison

- Avant de commencer, vous devez avoir compris les notions de file d'attente maildrop, file d'attente entrante, file d'attente active et file d'attente retardée présentées à la page [QSHAPE README](#).
- En cas de livraison lente, lancez l'outil qshape comme décrit à la page [QSHAPE README](#).
- Soumettez plusieurs destinataires par message au lieu de soumettre les messages avec seulement quelques destinataires.
- Soumettez le courrier via SMTP au lieu d'utiliser /usr/sbin/sendmail. Vous devez ajuster le paramètre smtpd_recipient_limit.
- Ne surchargez pas le disque avec trop de soumission de messages. Optimisez le taux de soumission en ajustant le nombre de soumissions parallèles et/ou la valeur du paramètre in_flow_delay.
- Lancez un serveur DNS local pour réduire la latence des consultations DNS. Si vous utilisez plusieurs systèmes Postfix, faites pointer chaque serveur DNS local vers un serveur relais partagé pour réduire le nombre de consultations à travers le réseau.
- Réduisez les valeurs smtp_connect_timeout et smtp_helo_timeout pour que Postfix n'ai pas trop de connexions semi-ouvertes avec des serveurs smtpd(8) ne répondant pas.
- Utilisez un transporteur de messages dédié pour les destinations à problème, avec des timeouts réduits et une concurrence ajustée. Reportez-vous au paragraphe "[Optimiser le nombre de livraisons simultanées](#)" ci-dessous.
- Utilisez une machine fallback_relay pour le courrier qui ne peut être livré à la première tentative. Cette machine "garde-fou" peut utiliser des délais entre tentatives plus courts pour atteindre les destinations à problème. Reportez-vous au paragraphe "[Optimiser la fréquence de tentatives de livraison du courrier retardé](#)" ci-dessous.
- Accélérez les mises à jour du disque avec un grand cache d'écriture persistant (64MB). Ceci autorise les mises à jour à être triées pour optimiser la vitesse d'accès sans compromettre l'intégrité du système de fichier lorsque le système plante.
- Utilisez un disque à état solide (un disque RAM persistant). C'est une solution chère qui peut être utilisée en combinaison avec des temps limites SMTP courts et une machine "garde-fou" fallback_relay qui traite le courrier des destinations à problème.

Optimiser le nombre de livraisons simultanées

Bien que Postfix puisse être configuré pour faire fonctionner 1000 processus client SMTP en même temps, il est rarement souhaitable qu'il fasse 1000 connexions simultanées vers la même machine distante. Pour cette raison, Postfix dispose de mécanismes de sûreté pour éviter ce problème (nommé "thundering herd").

Le gestionnaire des files d'attente implémente l'équivalent de la stratégie de démarrage lent du protocole TCP : lors de la livraison vers un site, il envoie un petit nombre de messages dans un premier temps puis augmente la concurrence tant que tout va bien et augmente la concurrence en cas de congestion.

- Le paramètre initial_destination_concurrency (défaut : 5) contrôle le nombre de messages initialement envoyés à la même destination avant d'adapter la concurrence de livraison. Bien sûr, cette valeur est effective seulement tant qu'elle n'excède pas la limite des processus et de concurrence du transporteur.
- Le paramètre default_destination_concurrency_limit (défaut : 20) contrôle le nombre de messages qui peuvent être envoyés à la même destination simultanément. Vous pouvez surcharger cette valeur pour un transporteur de messages particulier en ajoutant son nom d'entrée master.cf devant "_destination_concurrency_limit".

Exemples de limites de concurrence spécifique à un transporteur :

- Le paramètre local_destination_concurrency_limit (défaut : 2) contrôle le nombre de messages délivrés simultanément au même destinataire local. La valeur limite recommandée est basse car la livraison à la même boîte-aux-lettres doit être faite séquentiellement, rendant inutile le parallélisme massif. Une autre bonne raison de limiter la concurrence de livraison au même destinataire : si le destinataire a une commande shell coûteuse dans son fichier .forward, ou si le destinataire est un gestionnaire de liste de diffusion, vous ne souhaitez sans doute pas lancer trop d'instances de ce processus en même temps.
- La valeur par défaut smtp_destination_concurrency_limit=20 semble suffisante pour charger sensiblement un système sans l'envoyer dans ses limites. Faites attention si vous élevez cette valeur.

Les valeurs par défaut de limite concurrence ci-dessus correspondent parfaitement à un large panel de situations. Un changement non réfléchi de ces paramètres face à une congestion peut aggraver le problème. En particulier, les fortes concurrences de destination ne doivent jamais être générales. Elles ne doivent être utilisées que pour les transporteurs qui livrent un petit nombre de sites à fort volume.

Une situation commune où une forte concurrence est souhaitable concerne les passerelles relayant un fort volume de messages entre Internet et un intranet. Approximativement la moitié des messages (en supposant que les trafics entrant et sortant sont équivalents en volume) seront destinés au commutateur interne. Comme ces commutateurs de messages recevront du courrier externe enquiemement de la passerelle, il est raisonnable de configurer la passerelle pour faire des requêtes plus lourdes sur les serveurs SMTP internes.

L'optimisation des limites de concurrence entrantes doit être testé.

Un commutateur à forte capacité devrait être capable de traiter 50 à 100 connexions simultanées (au lieu de 20 par défaut), en particulier si la passerelle transfère le courrier de multiples machines MX. Lorsque toutes les machines MX sont actives et acceptent les connexions en temps réel, le taux de transfert sera élevé. Si une de ces machines MX est hors service et ne répond pas du tout, la latence de connexion correspondante décroît de $1/N * \$smtp_connection_timeout$, où N est le nombre de machines MX. ces limites descendent au plus à (concurrence de la destination * N / $\$smtp_connection_timeout$).

Par exemple, avec une concurrence de destination fixée à 100 et 2 machines MX, chaque machine recevra 50 connexions simultanées. Si un MX s'arrête et que le temps limite de connexion SMTP par défaut est de 30s, la limite de transfert est de $100 * 2 / 30 \approx 6$ messages par seconde. Ceci montre que les destinations à fort volume avec une bonne connectivité et de multiples machines MX ont besoin d'un temps limite de connexion plus bas, des valeurs de l'ordre de 5s voire 1s peuvent être utilisées pour prévenir les congestions lorsqu'un ou plus, mais pas toutes les machines MX sont hors service.

Si nécessaire, mettez une plus forte valeur transporteur_destination_concurrency_limit (dans main.cf car il s'agit d'un paramètre du gestionnaire des files d'attente) et une plus basse valeur smtp_connection_timeout (avec l'option "-o" dans master.cf pour surcharger le paramètre main.cf car ce paramètre n'est pas relatif à un transporteur) pour le transporteur relais et tous les transporteurs dédiés aux destinations à fort volume.

Optimiser le nombre de destinataires par livraison

Le paramètre default_destination_recipient_limit (défaut : 50) contrôle le nombre de destinataires qu'un agent de livraison de Postfix inclura dans chaque copie d'un message. Vous pouvez surcharger ce paramètre pour des agents de livraison particuliers. Par exemple, "uucp_destination_recipient_limit = 100" limitera le nombre de destinataires UUCP à 100.

Si un message excède la limite de destinataires pour une destination, le gestionnaire des files d'attente de Postfix coupera la liste des destinataires en de plus petites listes. Postfix tentera d'envoyer de multiples copies du message en parallèle.

IMPORTANT : Faites attention en augmentant la limite des destinataires par livraison de message ; certains serveurs smtpd(8) coupent la connexion lorsqu'ils dépassent la mémoire allouée ou lorsque la limite des destinataires est atteinte, et le message n'est jamais livré.

Le paramètre smtpd_recipient_limit (défaut : 1000) contrôle le nombre de destinataires que serveur smtpd(8) de Postfix acceptera par livraison. La limite par défaut est supérieure à ce qu'un client SMTP raisonnable enverra. Cette limite existe pour protéger le système de messagerie local contre un client tournant en boucle.

Optimiser la fréquence de tentatives de livraison du courrier retardé

Lorsqu'un agent de livraison de Postfix (smtp(8), local(8), etc.) n'est pas capable de livrer un message il peut blamer le message lui-même, ou blamer la partie tierce.

- Lorsque l'agent de livraison blame un message, le gestionnaire des files d'attente donne au fichier en file d'attente une date de modification dans le futur, ainsi il ne sera pas traité avant. Par défaut, le temps d'attente est le temps depuis que le message est arrivé. Ce résultat est appelé "exponential backoff behavior".
- Lorsque l'agent de livraison blame la partie receptionnaire (par exemple un utilisateur destinataire local, ou une machine distante), le gestionnaire des files d'attente n'avance pas seulement la date du fichier dans la file d'attente, mais met également cette partie tierce dans une liste "morte" pour qu'elle soit ignorée pendant un certain temps.

Ce processus est gouverné par quelques paramètres.

queue_run_delay (défaut : 1000 secondes)

Fréquence d'examen de la file d'attente retardée par le gestionnaire des files d'attente.

minimal_backoff_time (défaut : 1000 secondes)

Délai minimum avant nouvel examen et délai minimum de mise en liste "morte".

maximal_backoff_time (défaut : 4000 secondes)

Délai maximum avant nouvel examen suite à un échec de livraison.

maximal_queue_lifetime (défaut : 5 jours)

Délai à partir duquel un message est retourné comme non livrable. Indiquez 0 pour retourner les messages immédiatement après le premier échec de livraison.

bounce_queue_lifetime (défaut : 5 jours, disponible à partir de la version 2.1 de Postfix)

Temps maximum de présence d'un message MAILER-DAEMON en file d'attente avant qu'il soit considéré non livrable. Indiquez 0 pour supprimer ces messages dès le premier échec.

qmgr_message_recipient_limit (défaut : 20000)

Taille de beaucoup de structures de données en mémoire du gestionnaire des files d'attente. Entre autres, ce paramètre limite la taille de la liste volatile en mémoire des destinations "mortes". Les destinations supplémentaires ne sont pas ajoutées.

IMPORTANT : si vous augmentez la fréquence de tentative de livraison des messages retardés ou si vous videz la file d'attente des messages retardés, vous pourrez trouver que les performances de Postfix se dégradent. Les symptômes sont les suivants :

- La file d'attente active est saturée avec les messages à problème. Les nouveaux messages entrent dans la file d'attente active seulement lorsqu'un vieux message est retardé. C'est un processus qui atteint généralement les limites de temps d'une ou plusieurs connexions SMTP.
- Tous les agents de livraison disponibles sont rapidement occupés à tenter de se connecter à des sites non joignables. Les nouveaux messages doivent attendre qu'un agent de livraison se libère. C'est un processus qui atteint généralement les limites de temps d'une ou plusieurs connexions SMTP.

Lorsque le courrier est fréquemment retardé, régler le problème est toujours meilleur que d'augmenter la fréquence des tentatives de livraisons. Toutefois, si vous ne pouvez contrôler que cette fréquence, utilisez une machine fallback relay "garde-fou" dédiée pour les destinations à problème ainsi elles ne ruinent pas les performances de la livraison normale.

Optimiser le nombre de processus Postfix

Le paramètre de configuration default_process_limit donne un contrôle direct sur le nombre de processus démons que Postfix lancera. Depuis Postfix 2.0 la limite est fixée à 100 processus client smtp, 100 processus serveurs smtp, et ainsi de suite. Cela peut dépasser les capacités des systèmes pauvres en mémoire comme des réseaux à faible bande passante.

Vous pouvez changer la limite globale des processus en renseignant default_process_limit dans le fichier main.cf. Par exemple, pour lancer 10 processus client smtp, 10 processus serveur smtp et ainsi de suite :

```
/etc/postfix/main.cf:
    default_process_limit = 10
```

Vous devez lancer "postfix reload" pour activer ces changements. Les limites sont contrôlées par le démon master(8) de Postfix qui ne relit pas automatiquement main.cf lorsqu'il change.

Vous pouvez surcharger la limite des processus pour des démons particulier de Postfix en éditant le fichier main.cf de Postfix. Par exemple, si vous ne voulez pas recevoir 100 messages SMTP en même temps, mais ne voulez pas changer les limites de processus pour la livraison locale, vous pouvez indiquer :

```
/etc/postfix/master.cf:
# =====
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
# =====
. . .
smtp          inet  n       -       -       -       10       smtpd
. . .
```

Optimiser le nombre de fichiers ou de sockets ouverts

Lorsque Postfix ouvre trop de fichiers ou de sockets, les processus plantent et le système enregistre des erreurs "file table full".

- Réduisez le nombre de processus comme décrit au paragraphe "Optimiser le nombre de processus Postfix" ci-dessus. Moins de processus ouvrent moins de fichiers et de sockets.
- Configurez le noyau pour ouvrir plus de fichiers et de sockets. Ces détails sont très dépendants du système et changent avec les versions du système d'exploitation. Assurez-vous de vérifier ces informations avec votre guide d'optimisation du système :

Documentation de Postfix en français

- ◆ Certains paramètres de noyaux FreeBSD peuvent être indiqués dans le fichier `/boot/loader.conf`, et d'autres peuvent être changés avec les commandes `sysctl` suivant le numéro de version.

```
kern.ipc.maxsockets="5000"  
kern.ipc.nmbclusters="65536"  
kern.maxproc="2048"  
kern.maxfiles="16384"  
kern.maxfilesperproc="16384"
```

- ◆ Les paramètres du noyau Linux peuvent être indiqués dans le fichier `/etc/sysctl.conf` et peuvent être également changés avec les commandes `sysctl` :

```
fs.file-max=16384  
kernel.threads-max=2048
```

- ◆ Les paramètres du noyau Solaris peuvent être indiqués dans le fichier `/etc/system`, tel que décrit dans l'entrée "How can I increase the number of file descriptors per process?" de la page [Solaris FAQ](#).

```
* set hard limit on file descriptors  
set rlim_fd_max = 4096  
* set soft limit on file descriptors  
set rlim_fd_cur = 1024
```

Howto déboguage Postfix

But de ce document

Ce document explique comment déboguer des éléments du système de messagerie Postfix lorsque le fonctionnement ne correspond pas à ce qui est attendu. Les méthodes varient de l'enregistrement volubile dans les journaux jusqu'à faire fonctionner un processus démon sous le contrôle d'un debugger ou d'un traceur d'appel.

On supposera ici que les fichiers de configuration de Postfix `main.cf` et `master.cf` sont stockés dans le répertoire `/etc/postfix`. Vous pouvez utiliser la commande "**`postconf config_directory`**" pour trouver l'emplacement actuel de ce répertoire sur votre machine.

Listées dans l'ordre de niveau d'investigation, les techniques de déboguage sont les suivantes :

- Recherche des signes manifestes de problème
- Déboguer Postfix de l'intérieur
- Désactiver la mise en cage dans `master.cf`
- Journaux verbeux pour des connexions SMTP spécifiques
- Enregistrer une session SMTP avec un sniffer réseau
- Rendre les programmes démons de Postfix plus bavards
- Tracer manuellement un processus démon de Postfix
- Tracer automatiquement un processus démon de Postfix
- Lancer des programmes démons avec le debugger interactif `xxgdb`
- Lancer des programmes démons avec un debugger non-interactif
- Comportement déraisonnable
- Rapporter les problèmes à la liste `postfix-users@postfix.org`

Recherche des signes manifestes de problème

Postfix enregistre tous les échec et succès de livraison dans un fichier journal. Ce fichier est généralement `/var/log/maillog` ou `/var/log/mail` ; le chemin exact est défini dans le fichier `/etc/syslog.conf`.

Lorsque Postfix ne reçoit ni n'envoie de message, la première chose à faire est de rechercher les erreurs qui indiquent un fonctionnement anormal de Postfix :

```
% egrep '(warning|error|fatal|panic):' /un/fichier/de/log | more
```

Note : les messages les plus importants apparaissent en DÉBUT de sortie. Les messages d'erreurs suivants sont moins intéressants.

La nature de chaque problème est indiquée comme suit :

- "**panic**" indique un problème dans le logiciel lui-même que seul un programmeur peut régler. Postfix ne peut démarrer tant qu'il n'est pas fixé.

- **"fatal"** résultat d'un fichier manquant, de permissions incorrectes, de paramètres incorrects dans les fichiers de configuration que vous pouvez résoudre. Postfix ne peut démarrer tant qu'il n'est pas fixé.
- **"error"** rapporte une erreur fatale ou non. Postfix ne peut démarrer tant qu'il n'est pas fixé.
- **"warning"** indique une erreur non fatale. Ce sont des problèmes que vous pouvez ne pas être apte à régler (tels un serveur DNS fonctionnant mal quelque part sur le réseau) mais peut également indiquer une erreur de configuration locale qui peut devenir un problème ultérieurement.

Déboguer Postfix de l'intérieur

Avec les versions 2.1 et supérieures de Postfix, vous pouvez demander à Postfix de produire des rapports d'erreurs de livraison à des fins de débogage. Ces rapports ne montrent pas seulement les adresses d'expédition/de destination après réécriture et substitutions d'alias ou transfert, ils montrent également des informations sur la livraison en boîte-aux-lettres à une commande non-Postfix, les réponses des serveurs SMTP distants, et ainsi de suite.

Postfix peut produire deux types de rapports de livraison à des fins de débogage :

- **What-if:** rapporte ce qui devrait arriver mais ne délivre pas le message. Ce mode opératoire est appelé avec :

```
$ /usr/sbin/sendmail -bv address...  
Le rapport sera envoyé à <votre nom de login>.
```

- **What happened:** livre le message et rapporte les succès et/ou échecs y compris ceux des serveurs SMTP distants. Ce mode opératoire est appelé avec :

```
$ /usr/sbin/sendmail -v address...  
Le rapport sera envoyé à <votre nom de login>.
```

Ces rapports contiennent les informations générées par les agents de livraison de Postfix. Comme ces processus sont des démons et n'interagissent pas directement avec les utilisateurs, le résultat est envoyé par message à l'expéditeur du message de test. Le format de ces rapports est pratiquement identique à ceux des notifications ordinaires de non-livraison.

Un exemple détaillé de rapport de livraison est présenté au paragraphe [debugging](#) à la fin de la page [ADDRESS REWRITING README](#).

Désactiver la mise en cage dans master.cf

Une faute courante est d'activer la mise en cage chroot dans le fichier master.cf de Postfix sans suivre toutes les étapes de constitution de la cage chroot. Ceci entraîne un plantage des processus démons de Postfix dans tous les cas à des fichiers manquants.

L'exemple ci-dessous montre un serveur SMTP configuré avec la mise en cage chroot désactivée :

```
/etc/postfix/master.cf:  
# =====  
# service type private unpriv chroot wakeup maxproc command  
# (yes) (yes) (yes) (never) (100)  
# =====  
smtp inet n - n - - smtpd
```

Recherchez dans master.cf tous les processus qui n'ont pas la mise en cage chroot désactivée. Si vous en trouvez, faites une copie du fichier master.cf de Postfix et éditez l'entrée en question. Après avoir exécuté la commande "**postfix reload**", vérifiez si le problème a disparu.

Si c'est le cas, félicitations. Laisser Postfix tourner dans cette configuration est adéquate pour beaucoup de sites. Si vous préférez remettre le chroot, lisez la page [BASIC CONFIGURATION README](#) pour préparer la mise en cage chroot.

Journaux verbeux pour des connexions SMTP spécifiques

Dans /etc/postfix/main.cf, listez les noms de sites ou adresses dans le paramètre `debug_peer_list`. Par exemple, pour rendre le logiciel plus bavard pour les connexions sur la boucle locale :

```
/etc/postfix/main.cf:
    debug_peer_list = 127.0.0.1
```

Vous pouvez indiquer plusieurs machines, domaines, adresses ou réseaux/masque. Pour rendre les changements effectifs immédiatement, exécutez la commande "**postfix reload**".

Enregistrer une session SMTP avec un sniffer réseau

Cet exemple utilise **tcpdump**. Pour enregistrer une conversation, vous devez indiquer un buffer assez grand avec l'option "-s" sinon, vous perdrez tout ou partie du contenu des paquets.

```
# tcpdump -w /nom/de/fichier -s 2000 host exemple.com and port 25
```

Lancez cette commande et tapez Ctrl-C pour arrêter. Pour voir les données, utilisez un visualisateur binaire, **ethereal**, or utilisez mon utilitaire **tcpdumpx** disponible à l'adresse [ftp://ftp.porcupine.org/pub/debugging/](http://ftp.porcupine.org/pub/debugging/).

Rendre les programmes démons de Postfix plus bavards

Ajoutez une ou plusieurs options "-v" aux définitions des démons choisis dans /etc/postfix/master.cf et tapez "**postfix reload**". Ceci entraînera la journalisation de toute l'activité dans le démon syslog. Exemple :

```
/etc/postfix/master.cf:
smtp      inet  n       -       n       -       -       smtpd -v
```

Rend le serveur SMTP de Postfix plus bavard. Pour diagnostiquer les problèmes de réécriture d'adresses, on peut ajouter l'option "-v" au démon [cleanup\(8\)](#) et/ou [trivial-rewrite\(8\)](#), et pour les problèmes de livraison, on peut ajouter l'option "-v" au gestionnaire des files d'attente [qmgr\(8\)](#) ou [oqmgr\(8\)](#), ou aux agents de livraison [lmtp\(8\)](#), [local\(8\)](#), [pipe\(8\)](#), [smtp\(8\)](#), ou [virtual\(8\)](#).

Tracer manuellement un processus démon de Postfix

Beaucoup de systèmes vous autorisent à inspecter manuellement un processus tournant avec un traceur d'appels système. Par exemple :

```
# trace -p process-id (SunOS 4)
# strace -p process-id (Linux et beaucoup d'autres)
# truss -p process-id (Solaris, FreeBSD)
```

```
# ktrace -p process-id (generic 4.4BSD)
```

Encore plus d'informations peuvent être obtenues des appels de bibliothèques. Exemples :

```
# ltrace -p process-id (Linux, également porté sur FreeBSD et BSD/OS)
# sotruss -p process-id (Solaris)
```

Lisez la documentation de votre système pour plus de détails.

Tracer un processus tournant peut donner des informations sur ce qu'un processus tente de faire. Cette technique fournit le maximum d'informations que vous pouvez obtenir sans utiliser un programme debugger interactif comme décrit dans le dernier paragraphe.

Tracer automatiquement un processus démon de Postfix

Postfix peut attacher un traceur d'appels dès qu'un processus démon démarre. Les traceurs d'appel peuvent être utilisés dans plusieurs cas.

1. Traceurs d'appels système tels **trace**, **truss**, **strace**, ou **ktrace**. Ils montrent les communications entre les processus et le noyau.
2. Traceurs d'appels aux bibliothèques tels **sotruss** et **ltrace**. Ils montrent les appels aux routines des bibliothèques et donnent une meilleure idée de ce que fait un processus.

Ajoutez une option **-D** à la commande suspecte dans `/etc/postfix/master.cf`. Par exemple :

```
/etc/postfix/master.cf:
smtp      inet  n       -       n       -       -       smtpd -D
```

Editez l'option debugger_command dans `/etc/postfix/main.cf` pour que Postfix invoque le traceur d'appel de votre choix, par exemple :

```
/etc/postfix/main.cf:
debugger_command =
    PATH=/bin:/usr/bin:/usr/local/bin:
    (truss -p $process_id 2>&1 | logger -p mail.info) & sleep 5
```

Tapez "**postfix reload**" et observez les journaux.

Lancer des programmes démons avec le debugger interactif **xxgdb**

Si vous avez un système X-Windows installé sur votre machine Postfix, alors un debugger interactif tel **xxgdb** peut être adapté.

Renseignez debugger_command dans `/etc/postfix/main.cf` pour qu'il invoque **xxgdb** :

```
/etc/postfix/main.cf:
debugger_command =
    PATH=/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin
    xxgdb $daemon_directory/$process_name $process_id & sleep 5
```

Assurez-vous que **xxgdb** est dans les chemins de recherche des commandes (\$PATH), et exportez **XAUTHORITY** pour que le contrôle d'accès X fonctionne, Par exemple :

```
% setenv XAUTHORITY ~/.Xauthority (syntaxe csh)
$ export XAUTHORITY=$HOME/.Xauthority (syntaxe sh)
```

Ajoutez une option **-D** à la définition du démon suspect dans /etc/postfix/master.cf, Par exemple :

```
/etc/postfix/master.cf:
smtp      inet  n       -       n       -       -       smtpd -D
```

Arrêtez et redémarrez le système Postfix. Ceci est nécessaire pour que Postfix se lance avec les **XAUTHORITY** et **DISPLAY** appropriés.

Pendant que le démon suspect est démarré, une fenêtre du debugger apparaît et vous pouvez observer en détail ce qui se passe.

Lancer des programmes démons avec un debugger non-interactif

Si vous n'avez pas de X-Windows installed on the Postfix machine, ou si vous n'êtes pas familier avec les debuggers interactifs, vous pouvez essayer **gdb** en mode non interactif, et obtenir une trace de la pile d'appels lorsque le processus plante.

Renseignez debugger_command dans le fichier /etc/postfix/main.cf pour qu'il invoque le debugger **gdb** :

```
/etc/postfix/main.cf:
debugger_command =
    PATH=/bin:/usr/bin:/usr/local/bin; export PATH; (echo cont; echo
    where; sleep 8640000) | gdb $daemon_directory/$process_name
    $process_id 2>&1
    >$config_directory/$process_name.$process_id.log & sleep 5
```

Ajoutez un option **-D** au démon suspect dans le fichier /etc/postfix/master.cf, par exemple :

```
/etc/postfix/master.cf:
smtp      inet  n       -       n       -       -       smtpd -D
```

Tapez "**postfix reload**" pour activer les changement de configuration.

Pendant que le processus démon suspect est lancé, un fichier de sortie est créé nommé avec le nom du démon et son numéro de processus (par exemple, smtpd.12345.log). Lorsque le processus plante, une trace de la pile d'appel (avec sortie de la commande "**where**") est écrit dans ce fichier journal.

Comportement déraisonnable

Parfois, le comportement apparent de Postfix ne semble pas correspondre au code source. Comment un programme peut-il dévier des instructions données par son auteur ? Il y a deux possibilités :

- Le compilateur s'est trompé. Ceci arrive rarement.
- La hardware s'est trompé. La machine a-t-elle assez de mémoire ECC ?

Dans les deux cas, le programme en fonctionnement n'est pas le programme supposé être exécuté donc tout peut arriver.

Il y a une troisième possibilité :

- Des bugs dans le logiciel système (noyau ou librairies).

Les fautes relatives au Hardware ne se reproduisent pas exactement de la même manière après un redémarrage du système. Il y a peu que Postfix puisse faire sur un hardware défaillant. Assurez-vous d'utiliser du matériel qui peut détecter les erreurs de mémoire. Les systèmes critiques masquent le matériel réel.

Lorsqu'un compilateur fait une erreur, le problème peut être reproduit chaque fois que le programme résultat est lancé. Les erreurs du compilateur ont plus de chance d'arriver dans l'optimiseur de code. Si le problème est reproductible après les redémarrages, il peut être intéressant de recompiler Postfix en désactivant l'optimisation et d'observer la différence.

Pour recompiler Postfix en désactivant l'optimisation :

```
% make tidy
% make makefiles OPT=
```

Ceci produit des Makefiles qui ne requièrent pas l'optimisation.

Une fois les Makefiles créés, compilez le logiciel :

```
% make
% su
Password:
# make install
```

Si le problème continue, il est tant de demander de l'aide à votre vendeur.

Rapporter les problèmes à la liste postfix-users@postfix.org

Les personnes inscrits sur la liste postfix-users@postfix.org sont d'une aide précieuse, en particulier si VOUS leur fournissez assez d'information. Souvenez-vous que ce sont des bénévoles et que leur temps est limité. *Attention, il s'agit d'une liste de diffusion anglophone ! Si vous ne maîtrisez pas suffisamment la langue de Shakespeare, vous pouvez poser vos questions sur le groupe de news fr.comp.mail.serveur.*

Lorsque vous rapportez un problème, assurez-vous d'inclure les informations suivantes.

- Résumé du problème. S'il vous plait, n'envoyez pas seulement des logs sans explications ou ce que VOUS pensez disfonctionner.
- Complétez les messages d'erreur. Utilisez s'il vous plait le copier-coller ou utilisez une pièce jointe au lieu de réciter les informations de mémoire.
- Journaux de Postfix logging. Lisez le texte du haut de la page [DEBUG README](#) pour trouver les journaux. Ne frustrez pas les bénévoles en vous trompant de journaux.
- Utilisez une adresse de test pour ne pas révéler votre véritable adresse mail ou des mots-de-passe.
- Si vous ne pouvez pas utiliser une adresse de test, anonymisez les informations. Remplacez chaque lettre par un "A", chaque nombre par un "D" pour que les bénévoles puissent reconnaître les erreurs

de syntaxe.

- Sortie de "postconf -n". N'envoyez pas votre fichier main.cf ou plus de 400 lignes de sortie de **postconf**.
- Utilisez plutôt la sortie de l'outil "postfinger". Il peut être trouvé sur <http://ftp.wl0.org/SOURCES/postfinger>.
- Si le problème est relatif à SASL, incluez la sortie de l'outil **saslfinger**. Il est disponible sur <http://postfix.state-of-mind.de/patrick.koetter/saslfinger/>.
- Si le problème concerne un encombrement des files d'attente, incluez la sortie de l'outil qshape, comme décrit à la page [QSHAPE_README](#).
- Si le problème est relatif au protocole (timeouts des connexions ou un serveur SMTP se plaignant des erreurs de syntaxe etc.), enregistrez une session avec tcpdump, comme décrit à la page [DEBUG_README](#).

Inspection du contenu par Postfix

Postfix supporte 3 méthodes d'inspection du contenu, depuis l'inspection légère ligne par ligne avant la mise en file d'attente jusqu'à l'examen sophistiqué après cette mise en file d'attente. Chaque approche poursuit un but différent.

Inspection légère intégrée en temps réel

Cette méthode examine le message AVANT la mise en file d'attente et utilise l'inspection des en-têtes et du contenu intégrée à Postfix. Bien que la finalité principale est de stopper la propagation des vers et virus, elle est également pratique pour bloquer les messages indésirables et les notifications issues des anti-virus. Les expressions rationnelles intégrées ne sont pas conçues pour implémenter une détection générale des SPAMS et des virus. Pour ceci, vous devez utiliser une des méthodes d'inspection du contenu décrites plus loin dans ce document. Pour plus de détails, reportez-vous aux pages [BUILTIN_FILTER_README](#) (inspection intégrée) et [BACKSCATTER_README](#) (traitement des notifications issues des autres systèmes).

Inspection lourde externalisée et en temps différé

Cette méthode examine le courrier APRÈS le stockage en file d'attente et utilise des protocoles standards comme SMTP ou "lancement par PIPE et attente du status de sortie". Cette inspection après-stockage vous permet d'utiliser des filtres complexes sans causer des interruptions liées aux timeouts lors de la réception du courrier et sans faire exploser les ressources de votre système. Pour plus de détails sur cette approche, reportez-vous à la page [FILTER_README](#).

Inspection intermédiaire externalisée et en temps réel

Les deux méthodes suivantes inspectent le message AVANT qu'il ne soit stocké en file d'attente :

- ◇ la première méthode utilise le protocole SMTP et est décrite à la page [SMTPD_PROXY_README](#). Cette approche est disponible sur les versions 2.1 et supérieures de Postfix ;
- ◇ la seconde méthode utilise le protocole Milter de Sendmail 8 et est décrite à la page [MILTER_README](#). Cette approche est disponible sur les versions 2.3 et supérieures de Postfix.

Bien que ces approches paraissent attractives, elles présentent quelques sérieuses limites auxquelles vous devez prendre garde. Premièrement, le logiciel d'inspection du contenu doit achever son examen dans un temps limité ; si l'examen dure trop, le délai de livraison peut être dépassé. Deuxièmement, ce logiciel doit fonctionner avec une quantité de mémoire limitée ; sinon, le système risque de planter. L'inspection en temps réel limite la capacité de votre système ainsi que la sophistication du filtre.

Le filtrage avancé n'est pas intégré à Postfix pour de bonnes raisons : l'écriture un système de routage du courrier ne se conçoit pas comme l'écriture d'un anti-spam ou d'un anti-virus. Postfix encourage l'utilisation de filtres externes et de protocoles standards car cela vous permet de choisir séparément le meilleur MTA (mail transfert agent) et le meilleur logiciel inspecteur de contenu. Des informations sur les logiciels d'inspection de contenu externes peuvent être trouvées sur le site de Postfix (<http://www.postfix.org/>) et sur la liste postfix-users@postfix.org.

Arrêt des notifications indésirables

Introduction

Les fonctionnalités décrites ici sont disponibles sur les versions 2.0 et supérieures de Postfix

Sujets abordés dans ce document :

- Que sont les notifications indésirables ?
- Comment bloquer les notifications à destination d'adresses aléatoires ?
- Comment bloquer les notifications à destination d'adresses réelles ?
 - ◆ Bloquer les notifications avec un HELO renseigné
 - ◆ Bloquer les notifications contenant un expéditeur
 - ◆ Bloquer les notifications avec d'autres informations
 - ◆ Bloquer les notifications des antivirus

Que sont les notifications indésirables ?

Lorsque les spammers ou les vers envoient du courrier avec des adresses d'expédition forgées, les sites innocents sont envahis par des avis de non remise. C'est ce qu'on appelle les notifications indésirables (backscatter mail).

Comment bloquer les notifications à destination d'adresses aléatoires ?

Si votre machine reçoit des notifications indésirables à destination d'adresses aléatoires, configurez Postfix pour rejeter le courrier des utilisateurs inexistantes comme décrit dans les pages LOCAL RECIPIENT README (destinataires locaux) et STANDARD CONFIGURATION README (configuration standard).

Si vous utilisez une version 2.0 de Postfix ou antérieure, désactivez la fonctionnalité "attendre avant rejet" ("pause before reject") du serveur SMTP. Si votre système est soumis à une forte charge, il ne doit pas perdre de temps.

```
/etc/postfix/main.cf:  
# Pas nécessaire avec Postfix versions 2.1 et supérieures.  
smtpd_error_sleep_time = 0
```

Comment bloquer les notifications à destination d'adresses réelles ?

Lorsque les notifications passent la barrière "destinataire inconnu", il ne faut pas désespérer. Beaucoup de systèmes insèrent les en-têtes du message d'origine dans la notification. Ces en-têtes de message contiennent des informations que vous pouvez utiliser pour reconnaître et bloquer les indésirables.

Bloquer les notifications avec un HELO renseigné

Bien que mon adresse mail soit "wietse@porcupine.org", tous mes systèmes de messagerie s'annoncent comme "machine.porcupine.org" dans la commande SMTP HELO. Ainsi si un message retourné contient un en-tête Received: tel que suit :

```
Received: from porcupine.org ...
```

Alors je sais que qu'il s'agit d'un message forgé. Les messages réellement envoyés par mes systèmes ressemblent à :

```
Received: from machine.porcupine.org ...
```

Pour les mêmes raisons, les en-têtes suivants semblent être le résultat d'un message forgé :

```
Received: from host.example.com ([1.2.3.4] helo=porcupine.org) ...
Received: from [1.2.3.4] (port=12345 helo=porcupine.org) ...
Received: from host.example.com (HELO porcupine.org) ...
Received: from host.example.com (EHLO porcupine.org) ...
```

Un autre signe fréquent de forgeage est l'en-tête Message-ID:. Mon système produit un Message-ID: du style <stuff@hostname.porcupine.org>. Les suivants sont forgés, en particulier le premier

```
Message-ID: <1cb479435d8eb9.2beb1.qmail@porcupine.org>
Message-ID: <yulszqocfzsficvzzju@porcupine.org>
```

Pour bloquer de telles notifications, j'utilise les paramètres header_checks et body_checks comme suit :

```
/etc/postfix/main.cf:
header_checks = regexp:/etc/postfix/header_checks
body_checks = regexp:/etc/postfix/body_checks

/etc/postfix/header_checks:
/^Received: +from +(porcupine\.org) +/
    reject forged client name in Received: header: $1
/^Received: +from +[^ ]+ +\(((^ ]+ +[he]+lo=|[he]+lo +)(porcupine\.org)\)/
    reject forged client name in Received: header: $2
/^Message-ID:.*@(porcupine\.org)/
    reject forged domain name in Message-ID: header: $1

/etc/postfix/body_checks:
/^[> ]*Received: +from +(porcupine\.org) /
    reject forged client name in Received: header: $1
/^[> ]*Received: +from +[^ ]+ +\(((^ ]+ +[he]+lo=|[he]+lo +)(porcupine\.org)\)/
    reject forged client name in Received: header: $2
/^[> ]*Message-ID:.*@(porcupine\.org)/
    reject forged domain name in Message-ID: header: $1
```

Notes :

- L'exemple est simplifié dans un but pédagogique. En réalité, mes correspondances listent de multiples domaines : "(domaine1|domaine2|...)".
- Les "\" correspondent à de véritables ".". Sans le "\", le "." correspond à n'importe quel caractère.
- Les "\"(" et "\)" correspondent à de véritables parenthèses "(" et ")". Sans le "\", les parenthèses "(" et ")" groupent les opérateurs.

Avertissements

Les clients de messagerie Netscape (and par conséquent Mozilla) envoient un nom de domaine identique à celui de l'expéditeur dans le HELO. Si vous utilisez de tels clients, alors les correspondances précédentes bloqueront le courrier légitime.

Je n'ai qu'une telle machine sur mon réseau et pour éviter que son courrier soit bloqué, je l'ai configuré pour envoyer le courrier avec une adresse utilisateur@machine.porcupine.org. Sur le serveur Postfix, une correspondance canonique traduit cette adresse temporaire en utilisateur@porcupine.org.

```
/etc/postfix/main.cf:
    canonical_maps = hash:/etc/postfix/canonical

/etc/postfix/canonical:
    @hostname.porcupine.org @porcupine.org
```

C'est pratique si vous avez peu de systèmes qui envoient de telles commandes HELO et que vous n'avez jamais besoin d'envoyer du courrier à ces machines.

Une alternative consiste à supprimer le nom de machine avec le masquage d'adresse comme décrit à la page [ADDRESS REWRITING README](#).

Bloquer les notifications contenant un expéditeur

Comme beaucoup de personnes, j'ai quelques adresses dans des domaines que j'utilisais autrefois. Le courrier de ces adresses est transféré vers mon adresse actuelle. La majorité des notifications indésirables que je reçois provient de ces adresses. De telles notifications sont faciles à stopper.

```
/etc/postfix/main.cf:
    header_checks = regexp:/etc/postfix/header_checks
    body_checks = regexp:/etc/postfix/body_checks

/etc/postfix/header_checks:
    /^(From|Return-Path):.*[[:<:]](user@domain\.tld)[[:>:]]/
    reject forged sender address in $1: message header: $2

/etc/postfix/body_checks:
    /^[> ]*(From|Return-Path):.*[[:<:]](user@domain\.tld)[[:>:]]/
    reject forged sender address in $1: message header: $2
```

Notes :

- Cet exemple est simplifié pour des raisons pédagogiques. En réalité, mes correspondances listent plusieurs adresses email : "(user1@domain1\.tld|user2@domain2\.tld)".
- Le "[[:<:]]" correspond au début d'un mot et le "[[:>:]]" à la fin. Sur certains systèmes vous devez indiquer "\"<" et "\">" à la place. Pour plus de détails, reportez-vous à la documentation correspondant à votre système.
- Les "\". " correspondent à de véritables ". ". Sans le "\" , le ". " correspond à n'importe quel caractère.

Bloquer les notifications avec d'autres informations

Un autre signe de forgeage peut être détecté dans les adresses IP enregistrées dans l'en-tête Received: suivant votre nom de machine ou domaine HELO. Ces informations doivent être utilisées avec précaution. Certains

serveurs sont derrière un traducteur d'adresses et ne voient jamais l'adresse IP réelle du client.

Bloquer les notifications des antivirus

Après l'élimination des messages forgés reconnus, il reste une catégorie de notification à éliminer : les notifications des logiciels anti-virus. Malheureusement, certains ne savent pas que les virus forgent les adresses d'expédition. Pour ne rien arranger, ces logiciels ne savent pas non plus rapporter le problème de livraison, nous devons donc utiliser d'autres techniques.

Reconnaître les antivirus est un processus enclin d'erreurs car il y a beaucoup de formats de rapports différents. La suite est seulement un petit exemple de reconnaissance d'en-têtes de messages. Pour obtenir une liste plus grande d'expression caractérisant les notifications des antivirus, consultez la page <http://www.dkuug.dk/keld/virus/> ou <http://www.t29.dk/antiantivirus.txt>.

```
/etc/postfix/header_checks:  
/^Subject: *Your email contains VIRUSES/ DISCARD virus notification  
/^Content-Disposition:.*VIRUS1_DETECTED_AND_REMOVED/  
DISCARD virus notification  
/^Content-Disposition:.*VirusWarning.txt/ DISCARD virus notification
```

Reclamation auprès des opérateurs : s'il vous plait, n'aggavez pas le problème en retournant les messages forgés. Vous n'harcelez que des innocents. Si vous devez retourner le message au supposé expéditeur, renvoyez tous les en-têtes pour qu'il puisse se défendre des messages forgés.

Inspection du contenu intégrée à

Postfix

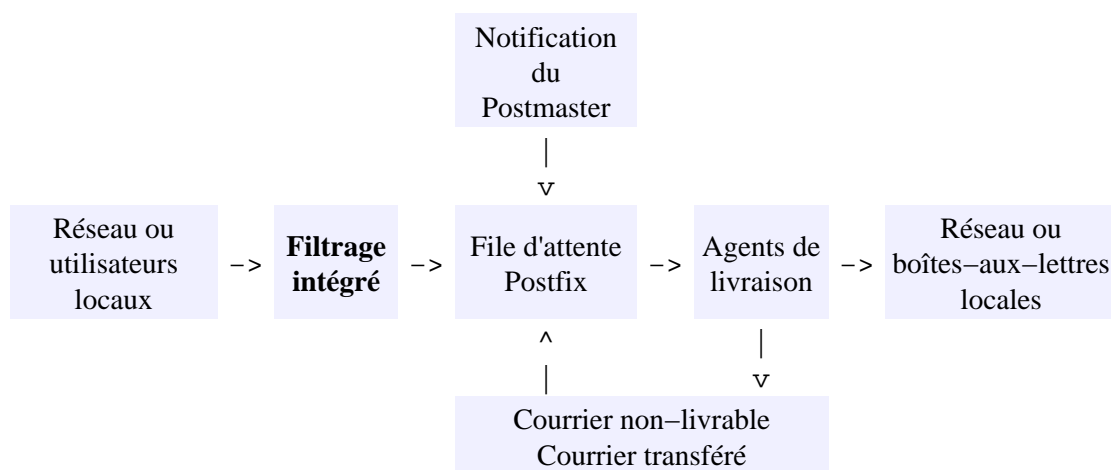
Introduction à l'inspection du contenu intégrée à Postfix

Postfix intègre un mécanisme de filtrage qui examine les en-têtes et le corps des messages une ligne à la fois avant la mise en file d'attente. Ce filtrage est généralement utilisé avec des expressions rationnelles POSIX ou PCRE (*Perl Compatible Regular Expressions*) comme décrit à la page de manuel [header_checks\(5\)](#).

L'originalité de ce filtrage est de stopper efficacement des virus ou vers spécifiques. Ce filtre a également aidé à bloquer le pourriel, les rebonds issus de virus et vers et notifications des antivirus. Pour plus d'information, reportez-vous à la page [BACKSCATTER_README](#) (notifications indésirables).

Puisque ce filtrage est optimisé pour stopper des virus ou vers spécifiques, il possède des limites qui n'en font pas un système général de détection des virus et pourriels. Pour ceci, vous devez utiliser un filtre externe comme présenté aux pages [FILTER_README](#) et [SMTPD_PROXY_README](#).

Le diagramme suivant présente le fonctionnement de cette inspection intégrée :



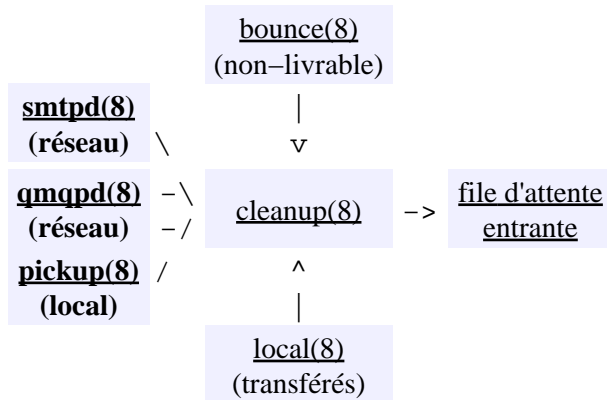
Ce schéma précise les travaux de filtrage se font pendant que Postfix reçoit le message. Ceci signifie que Postfix peut le rejeter sans avoir à retourner un avis de non-remise à l'expéditeur (qui est souvent artificielle). Cependant cet capacité a un prix : si l'inspection prend trop de temps, le client distant risque d'interrompre la livraison (timeout) et recommencer aussitôt.

Sujets abordés dans ce document :

- [Quel courrier est assujéti à l'examen des en-têtes/du corps](#)
- [Limites de l'examen des en-têtes/du corps](#)
- [Prévenir le blocage du rapport quotidien de livraison](#)
- [Configurer l'examen des en-têtes/du corps seulement pour le courrier venant de l'extérieur](#)
- [Configurer l'examen des en-têtes/du corps seulement pour le courrier de certains domaines](#)

Quel courrier est assujéti à l'examen des en-têtes/du corps

L'examen des en-têtes/du corps de Postfix est implementé dans le serveur cleanup(8) avant qu'il injecte le courrier dans la file d'attente entrante (incoming). Le schéma suivant se concentre sur le serveur cleanup(8) et montre ses différentes sources. Afin de simplifier le schéma, les sources de notifications au postmaster ont été omises car elles peuvent être produite par beaucoup de processus démons.



Pour de bonnes raisons, seuls les messages qui entrent depuis l'extérieur de Postfix sont inspectés. Il serait inutile de refiltrer de messages déjà filtrés et risquerait de bloquer les notifications du postmaster. Le tableau ci-dessous résume quels courriers peuvent être soumis au filtrage d'en-tête/du corps.

Type de message	Source	Examen d'en-tête/de corps ?
Avis de non-remise	<u>bounce(8)</u>	Non
Courrier du réseau	<u>smtpd(8)</u>	Configurable
Courrier du réseau	<u>qmqpd(8)</u>	Configurable
Emission locale	<u>pickup(8)</u>	Configurable
Transfert local	<u>local(8)</u>	Non
Notification au postmaster	nombreuses	Non

Comment Postfix décide-t-il quels messages doivent être filtrés ? Il serait imprudent de faire prendre cette décision au serveur cleanup(8) server, car ce programme reçoit des messages de nombreuses sources différentes. Ainsi, le filtrage d'en-tête/de corps est requis par la source. Des exemples montrant comment l'activer avec smtpd(8), qmqpd(8) ou pickup(8) sont montrés ci-dessous aux paragraphes "Configurer l'examen des en-têtes/du corps seulement pour le courrier venant de l'extérieur" et "Configurer l'examen des en-têtes/du corps seulement pour le courrier de certains domaines".

Limites de l'examen des en-têtes/du corps

- L'examen des en-têtes/du corps ne décode pas les en-têtes ou le corps du message. Par exemple, si le corps du message est encodé en BASE64 (RFC 2045) alors vos expressions rationnelles devront correspondre. De même les en-têtes contenant des caractères non ASCII (RFC 2047) devront être comparés à leur forme encodée.
- L'examen des en-têtes/du corps ne peut filtrer une combinaison des lignes des en-têtes et du corps. Il inspecte un en-tête à la fois ou une ligne du corps et ne peut prendre une décision au regard d'une autre ligne.
- L'examen des en-têtes/du corps ne peut dépendre du destinataire d'un message.

- ◆ Un message peut avoir de multiples destinataires et tous reçoivent le même traitement. Des contournements ont été proposés qui impliquent un traitement séparé par groupe de destinataire mais la performance du serveur SMTP est sensiblement altérée et le dispositif ne fonctionne pas pour les messages non-SMTP.
- ◆ Certaines sources envoient les en-têtes et le corps avant les informations sur la destination. Il serait très pénalisant de stocker en mémoire le message entier avant de décider s'il doit être filtré et il serait impensable d'appliquer le filtrage et mémoriser toutes les actions avant de savoir si elles doivent être exécutées.
- En dépit des avertissements, certaines personnes essaient d'utiliser le filtrage intégré pour la détection générale des virus et courriers indésirables en utilisant des centaines voire des milliers d'expressions rationnelles. Cela peut entraîner un résultat catastrophique. Les symptômes sont les suivants :
 - ◆ Les processus cleanup(8) utilisent toute la CPU et/ou la mémoire disponible et le système swappe. Ceci ralentit toute la livraison des messages entrants.
 - ◆ Comme Postfix a besoin de plus de temps pour recevoir un message, le nombre de sessions SMTP simultanées atteint la limite configurée dans le fichier master.cf.
 - ◆ Pendant que tous les processus du serveur SMTP attendent que les serveurs cleanup(8) se terminent, les nouveaux clients SMTP doivent attendre qu'un serveur SMTP se libère. Ceci met en timeout la livraison des messages avant même qu'elle ait commencée.

Le remède pour ce problème est simple : n'utilisez pas le filtrage intégré pour la détection générale des virus et courriers indésirables et ne filtrez pas le courrier avant qu'il soit mis en file d'attente. Lorsque la performance est une préoccupation, utilisez un filtre externe qui n'est lancé que lorsque le message est en file d'attente comme décrit à la page [FILTER README](#).

Prévenir le blocage du rapport quotidien de livraison

Ce qui suit est issu de la FAQ Pflogsumm de Jim Seymour

(<http://jimsun.linuxnet.com/downloads/pflogsumm-faq.txt>). Pflogsumm est un programme qui analyse les logs de Postfix y les rejets de messages. Si ces logs contiennent du texte justifiant le rejet par les expressions body_checks, alors le rapport sera rejeté par la même expression body_checks. Ce problème ne se pose pas avec les expressions header_checks car elles ne sont pas appliquées au contenu du rapport quotidien.

Vous configurez Postfix pour examiner le corps (body_checks). Les rapports Pflogsumm contiennent les chaînes éliminées dans son rapport. Il y a plusieurs solutions pour régler le problème.

Contribution de Wolfgang Zeikat :

```
#!/usr/bin/perl
use MIME::Lite;

### Create a new message:
$msg = MIME::Lite->new(
    From      => 'your@send.er',
    To        => 'your@recipie.nt',
    # Cc       => 'some@other.com, some@more.com',
    Subject   => 'pflogsumm',
    Date      => `date`,
    Type      => 'text/plain',
    Encoding  => 'base64',
    Path      => '/tmp/pflogg',
);
```

```
$msg->send;
```

Où `"/tmp/pflog"` est la sortie de `Pflogsumm`. Ceci transforme ce fichier en attachement `base64`.

Note de Wietse : si vous utilisez ceci sur une machine accessible par des utilisateurs non sûrs, il est préférable de stocker ce rapport dans un répertoire non positionné en écriture `pout=r` tous.

Dans la suite du thread dans la liste de diffusion `postfix-users`, Ralf Hildebrandt a noté :

"mpack fait la même chose."

Et il le fait. Chacun utilisera l'outil qu'il préfère.

D'autres solutions invoquent d'autres règles `body_checks` qui font exception des rapports, mais ce n'est pas recommandé. Ces règles ralentissent le serveur et compliquent la maintenance.

Configurer l'examen des en-têtes/du corps seulement pour le courrier venant de l'extérieur

Ce qui suit ne s'applique qu'aux versions 2.1 et supérieures de Postfix. Les versions antérieures ne supportent pas le dispositif `receive_override_options`.

La plus simple approche est de configurer UN Postfix avec de multiples adresses IP de serveur SMTP dans le fichier `master.cf` :

- Deux adresses de serveurs SMTP pour les utilisateurs internes uniquement avec désactivation du filtrage d'en-tête/du corps et un pickup local également sans filtrage.

```
/etc/postfix.master.cf:
# =====
# service      type  private unpriv  chroot  wakeup  maxproc command
#              (yes)  (yes)   (yes)   (never) (100)
# =====
1.2.3.4:smtp    inet  n       -       n       -       -       smtpd
-o receive_override_options=no_header_body_checks
127.0.0.1:smtp  inet  n       -       n       -       -       smtpd
-o receive_override_options=no_header_body_checks
pickup         fifo  n       -       n       60      1       pickup
-o receive_override_options=no_header_body_checks
```

- Un serveur SMTP pour le courrier provenant de l'extérieur avec filtrage activé via le fichier `main.cf`.

```
/etc/postfix.master.cf:
# =====
# service      type  private unpriv  chroot  wakeup  maxproc command
#              (yes)  (yes)   (yes)   (never) (100)
# =====
1.2.3.5:smtp    inet  n       -       n       -       -       smtpd
```

Configurer l'examen des en-têtes/du corps seulement pour le courrier de certains domaines

Documentation de Postfix en français

Ce qui suit ne s'applique qu'aux versions 2.1 et supérieures de Postfix. Les versions antérieures ne supportent pas le dispositif receive_override_options.

Si vous êtes un fournisseur de service MX et vous voulez désactiver le filtrage des en-têtes/du corps pour certains domaines, vous pouvez configurer UN Postfix avec plusieurs adresses IP de serveur SMTP dans le fichier. Chaque adresse fournit un service différent.

```
/etc/postfix/master.cf:
# =====
# service      type  private unpriv  chroot  wakeup  maxproc command
#              (yes)  (yes)   (yes)   (never) (100)
# =====
# SMTP service for domains with header/body checks turned on.
1.2.3.4:smtp    inet  n       -       n       -       -       smtpd

# SMTP service for domains with header/body checks turned off.
1.2.3.5:smtp    inet  n       -       n       -       -       smtpd
        -o receive_override_options=no_header_body_checks
```

Une fois que c'est fait, vous pouvez configurer les champs MX du DNS pour router chaque domaine sur sa propre instance SMTP.

Filtrage du contenu après mise en

file d'attente

Introduction

Ce document nécessite une version 2.1 ou supérieure de Postfix.

Normalement, Postfix reçoit le courrier, le stocke en file d'attente puis le livre. Avec le filtrage du contenu externalisé décrit ici, le courrier est filtré **APRÈS** sa mise en file d'attente. Cette approche découple les processus de réception et de filtrage et vous donne un contrôle maximum sur le nombre de processus que vous êtes disposés à lancer en parallèle.

Le filtre de contenu après mise en file d'attente est censé être utilisé comme suit :



Ce document décrit les implémentations qui utilisent une instance unique de Postfix pour tout : recevoir, filtrer et livrer le courrier. Les applications qui utilisent deux instances séparées de Postfix seront couvertes par une prochaine version de ce document.

Le filtre de contenu après mise en file d'attente ne doit pas être confondu avec l'approche décrite à la page [SMTPD_PROXY_README](#) où les messages SMTP entrants sont filtrés **AVANT** leur stockage en file d'attente.

Ce document décrit deux approches pour filtrer tout le courrier comme pour filtrer selectivement :

- [Principes d'opération](#)
- Filtrage simple de contenu
 - ◆ [Exemple de filtrage simple de contenu](#)
 - ◆ [Performances du filtrage simple de contenu](#)
 - ◆ [Limites du filtrage simple de contenu](#)
 - ◆ [Désactiver le filtrage simple de contenu](#)
- Filtrage avancé de contenu
 - ◆ [Exemple de filtrage avancé de contenu](#)
 - ◆ [Performances du filtrage avancé de contenu](#)
 - ◆ [Désactiver le filtrage avancé de contenu](#)
- Filtrage sélectif de contenu
 - ◆ [Filtrer le courrier des utilisateurs extérieurs seulement](#)
 - ◆ [Filtres différents par domaine](#)
 - ◆ [Actions FILTER dans les tables d'accès ou d'en-tête/contenu](#)

Principes d'opération

Un filtre de contenu externe reçoit les messages non filtrés de Postfix (comme décrit plus loin ci-dessous) et fait l'une des actions suivantes :

1. Re-injecte le message dans Postfix éventuellement en ayant changé le contenu et/ou la destination.
2. Rejette le message (en renvoyant un code de status à Postfix). Postfix le renvoie alors à l'expéditeur.

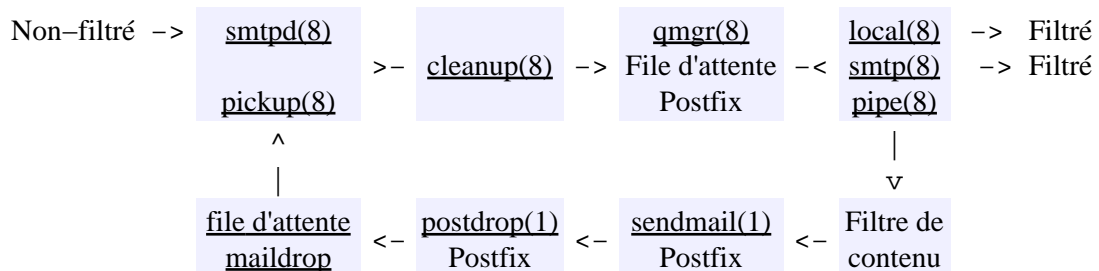
NOTE : à cette époque de vers et de messages de spam forgés, c'est une TRÈS MAUVAISE IDÉE de renvoyer le virus à l'expéditeur dont il n'est probablement pas l'initiateur. Il vaut mieux éliminer les virus connus et mettre en quarantaine les éléments suspects afin qu'un humain puisse décider quoi faire.

Exemple de filtrage simple de contenu

Le premier exemple est simple à mettre en uvre. Postfix reçoit le courrier du réseau via le serveur smtpd(8) et livre les messages non filtrés à un filtre de contenu avec l'agent de livraison pipe(8) de Postfix. Le filtre de contenu ré-injecte les messages filtrés dans Postfix avec la commande sendmail(1), ainsi Postfix peut le livrer à la destination finale.

Ceci signifie que le courrier soumis par la commande sendmail(1) de Postfix ne peut être filtré sur son contenu.

Dans le schéma ci-dessous, les noms suivis par un numéro représentent des commandes ou démons Postfix. Reportez-vous à la page OVERVIEW pour une présentation de l'architecture de Postfix.



Le filtre de contenu peut être un simple script shell comme suit :

```

1 #!/bin/sh
2
3 # Filtre shell simple. Il doit être appelé comme suit:
4 #       /path/to/script -f expéditeur destinataire...
5
6 # Répertoires
7 INSPECT_DIR=/var/spool/filter
8 SENDMAIL="/usr/sbin/sendmail -G -i" # n'employer JAMAIS "-t" ici
9
10 # Codes de retour issus de <sysexit.h>
11 EX_TEMPFAIL=75
12 EX_UNAVAILABLE=69
13
14 # Nettoyage lors en sortant ou lors d'une interruption
15 trap "rm -f in.$$" 0 1 2 3 15
16
17 # Démarrage du processus.
```

Documentation de Postfix en français

```
18 cd $INSPECT_DIR || {
19     echo $INSPECT_DIR n'existe pas; exit $EX_TEMPFAIL; }
20
21 cat >in.$$ || {
22     echo Impossible de sauvegarder le message dans un fichier; exit $EX_TEMPFAIL; }
23
24 # Ecrivez votre filtrage de contenu ici.
25 # filter <in.$$ || {
26 #     echo Contenu de message content rejeté; exit $EX_UNAVAILABLE; }
27
28 $SENDMAIL "$@" <in.$$
29
30 exit $?
```

Notes :

- Ligne 8 : l'option `-G` n'a aucun effet avant Postfix 2.3 et depuis, désactive la réécriture des adresses dans les en-têtes de messages.
- Ligne 8 : l'option `-i` indique de ne pas interrompre la lecture lorsqu'un ligne ne contient que ".".
- Ligne 8 : n'employer JAMAIS l'option `-t` ici. Ceci empêcherait la livraison comme si on envoyait un message de liste de diffusion de nouveau à la liste.
- Ligne 21 : la première action consiste à enregistrer le message dans un fichier et ensuite seulement le filtrer via un programme tiers.
- Ligne 22 : si le message n'a pas pu être enregistré, la livraison est retardée en renvoyant un code d'erreur 75 (`EX_TEMPFAIL`). Postfix place alors le message dans la file d'attente des messages retardés et tente une nouvelle livraison ultérieurement.
- Ligne 25 : vous devrez indiquer ici un réel programme de filtrage qui reçoit le contenu par l'entrée standard.
- Ligne 26 : si le filtre de contenu trouve un problème, le message est renvoyé en renvoyant un code d'erreur status 69 (`EX_UNAVAILABLE`). Postfix retournera le message à l'expéditeur comme non-livable.
- NOTE : à cette époque de vers et de messages de spam forgés, c'est une TRÈS MAUVAISE IDÉE de renvoyer le virus à l'expéditeur dont il n'est probablement pas l'initiateur. Il vaut mieux éliminer les virus connus et mettre en quarantaine les éléments suspects afin qu'un humain puisse décider quoi faire.
- Ligne 28 : si le contenu est accepté, il est passé à la commande `sendmail` de Postfix et le status de sortie du script est celui de cette commande. Postfix livrera le message comme d'habitude.
- Ligne 30 : le script retourne le status de la commande `sendmail` de Postfix.

Je vous suggère de lancer d'abord ce script à la main avec un vrai message (en-têtes+contenu) jusqu'à ce que vous soyez satisfait des résultats :

```
% /chemin/du/script -f sender recipient... <message-file
```

Une fois les résultats du filtrage satisfaisant :

- Créez un compte utilisateur dédié nommé par exemple "filter". Cet utilisateur transporte tous les contenus de message potentiellement dangereux – c'est pourquoi on utilise un compte séparé. N'utilisez pas "nobody", et surtout pas "root" ou "postfix".
- Créez un répertoire `/var/spool/filter` accessible uniquement à l'utilisateur "filter" dans lequel le script est supposé stocker ses fichiers temporaires.
- Configurez Postfix pour livrer le courrier au filtre de contenu avec l'agent de livraison pipe(8).

```
/etc/postfix/master.cf :
```

```
# =====
# service type  private unpriv  chroot  wakeup  maxproc command
#               (yes)   (yes)   (yes)   (never) (100)
# =====
filter    unix  -          n        n        -        10        pipe
flags=Rq user=filter argv=/path/to/script -f ${sender} -- ${recipient}
```

L'exemple de configuration ci-dessus lance au plus 10 filtres en parallèle. Au lieu de 10 processus concurrents, utilisez le nombre adapté à votre machine. Les logiciels d'inspection de contenu peuvent consommer beaucoup de ressources, c'est pourquoi vous devez indiquer une limite de processus concurrents.

- Pour activer le filtrage de contenu pour le courrier arrivant via SMTP seulement, ajoutez "`-o content_filter=filter:dummy`" à l'entrée du fichier master.cf qui définit le serveur SMTP de Postfix :

```
/etc/postfix/master.cf :
# =====
# service type  private unpriv  chroot  wakeup  maxproc command
#               (yes)   (yes)   (yes)   (never) (100)
# =====
smtp            inet  ...other stuff here, do not change...  smtpd
                -o content_filter=filter:dummy
```

La ligne "content_filter" oblige Postfix à ajouter une requête de filtrage à chaque message entrant, avec un contenu "filter:dummy". Cet enregistrement surcharge le routage normal du message pour le rediriger vers le filtre.

Le paramètre de configuration content_filter accepte la même syntaxe que la partie droite d'une table de transport.

- Lancez "`postfix reload`" pour activer ces changements.

Performances du filtrage simple de contenu

Avec le script shell script montré ci-dessus vous diminuerez les performances d'un facteur quatre pour les messages en transit arrivant et partant par SMTP. Vous diminuerez encore ces performances pour chaque fichier temporaire créé ou effacé par le processus de filtrage de contenu. Cet impact est moindre pour le courrier livré ou soumis localement car ces livraisons sont déjà plus lentes que le transit SMTP.

Limites du filtrage simple de contenu

Le problème avec les filtres de contenus tels le script présenté ci-avant est qu'ils ne sont pas très robustes. La principale raison est que le logiciel ne parle pas un protocole bien défini avec Postfix. Si le script shell de filtrage s'interrompt car le shell rencontre des problèmes d'allocation mémoire, le script ne produira pas un code d'erreur conforme au fichier `/usr/include/sysexits.h`. Au lieu d'être mis en file d'attente retardée, le message sera retourné. Le même défaut peut apparaître si le logiciel de filtrage rencontre lui même un problème de ressources.

La méthode simple de filtrage de contenu ne peut être invoquée par les résultats des consultations header_checks ou body_checks. Ces expressions seront appliquées de nouveau après la ré-injection avec la commande `sendmail` de Postfix créant une boucle de message. La méthode avancée de filtrage de contenu (voir ci-dessous) permet de désactiver les consultations header_checks ou body_checks pour les messages filtrés.

- éditez le fichier master.cf file, effacez le texte "`-o content_filter=filter:dummy`" de l'entrée définissant le serveur SMTP de Postfix.
- lancez "**postsuper -r ALL**" pour supprimer les informations de filtrage de contenu des fichiers déjà en file d'attente.
- relancez "**postfix reload**".

- La ligne "content_filter" oblige Postfix à ajouter un enregistrement de requête de contenu à chaque message entrant avec le contenu "scan:localhost:10025". Ces enregistrements sont ajoutés par les serveurs smtpd(8) et pickup(8) (et qmqpd(8) si vous avez activé ce service).
- Les requêtes de filtrage de contenu sont stockées dans les files d'attente, c'est ainsi que Postfix garde trace des messages qui doivent être filtrés. Lorsqu'une file d'attente contient une requête de filtrage de contenu, le gestionnaire des files d'attente (qmgr(8)) livre le message au filtre indiqué sans tenir compte de sa destination finale.
- La ligne "receive_override_options" désactive les manipulations d'adresses avant le filtrage permettant ainsi au filtre de voir l'adresse originale au lieu du résultat de la substitution des alias virtuels, des translations canoniques, de l'ajout automatique d'un destinataire caché (bcc), du masquage d'adresse, etc.

Filtrage avancé de contenu : envoyer les messages non filtrés au filtre de contenu

Dans cet exemple, "scan" est une instance de client SMTP de Postfix avec quelques paramètres de configuration différents. C'est pourquoi il est lancé ainsi dans le fichier master.cf de Postfix :

```
/etc/postfix/master.cf :
# =====
# service type private unpriv chroot wakeup maxproc command
# (yes) (yes) (yes) (never) (100)
# =====
scan unix - - n - 10 smtp
-o smtp_send_xforward_command=yes
```

- Ceci lance au plus 10 filtres de contenu en parallèle. Remplacez cette limite par une plus adaptée à votre système. Les logiciels d'inspection de contenu peuvent consommer beaucoup de ressources, c'est pourquoi il faut en limiter le nombre.
- Avec "-o smtp_send_xforward_command=yes", le transport "scan" essaiera de transférer le nom du client d'origine et l'adresse IP au processus smtpd après filtrage, ainsi les messages filtrés seront enregistrés avec l'adresse IP réelle du client. Reportez-vous aux pages smtp(8) et XFORWARD_README pour plus d'informations.

Filtrage avancé de contenu : lancer le filtre de contenu

Le filtre de contenu peut être activé par le service spawn de Postfix qui est l'équivalent de inetd. Par exemple, pour lancer 10 processus de filtrage de contenu sur le port local 10025 :

```
/etc/postfix/master.cf :
# =====
# service type private unpriv chroot wakeup maxproc command
# (yes) (yes) (yes) (never) (100)
# =====
localhost:10025 inet n n n - 10 spawn
user=filter argv=/chemin/vers/le/filtre localhost 10026
```

- "filter" est un compte utilisateur local dédié. Cet utilisateur ne se loguera jamais et peut recevoir un mot de passe "*", un shell et un répertoire utilisateur inexistant. Cet utilisateur manipulera tous les contenus de messages potentiellement dangereux – c'est pourquoi il doit s'agir d'un compte séparé.

Si votre filtre inclut des fonctionnalités réseau SMTP (telles les passerelles anti-virus), vous pouvez le lancer

en mode autonome et ne pas utiliser le service spawn de Postfix.

Filtrage avancé de contenu : ré-injecter les messages dans Postfix

Le travail du filtre de contenu est soit de renvoyer le message avec un suivi du diagnostic, soit de le ré-injecter dans Postfix sur le port local 10026.

Le plus simple des filtres de contenu se contente de recopier en sortie les commandes SMTP et les données reçues en entrée. S'il y a un problème, tout ce qu'il a à faire est de répondre au `.` de Postfix (commande de fin de DATA) avec `550 content rejected` et se deconnecter sans envoyer le `.` sur la connexion qui ré-injecte le message dans Postfix.

```
/etc/postfix/master.cf :
# =====
# service      type  private unpriv  chroot  wakeup  maxproc command
#                  (yes)   (yes)   (yes)   (never) (100)
# =====
localhost:10026 inet  n      -       n       -       10      smtpd
-o content_filter=
-o receive_override_options=no unknown recipient checks,no header body checks
-o smtpd_helo_restrictions=
-o smtpd_client_restrictions=
-o smtpd_sender_restrictions=
-o smtpd_recipient_restrictions=permit mynetworks,reject
-o mynetworks=127.0.0.0/8
-o smtpd_authorized_xforward_hosts=127.0.0.0/8
```

- Note : n'utilisez pas d'espaces autour des caractères "=" ou ",".
- Note : le serveur SMTP ne doit pas avoir une limite de processus inférieure à l'entrée "filter" du fichier master.cf.
- "-o content_filter=" surcharge le paramètre du fichier main.cf et indique de ne pas filtrer le contenu des messages arrivant du filtre. Sans ce paramètre, vous créez une boucle de message.
- "-o receive_override_options" surcharge le paramètre du fichier main.cf. Il est complémentaire des options spécifiées dans le fichier main.cf :

- ◆ Désactive la détection des destinataires inconnus et les examens d'en-têtes/de contenu. Ce travail a déjà été effectué avant le filtrage de contenu et le répéter serait inutile.
- ◆ Active la substitution des alias virtuels, les correspondances canoniques, le masquage d'adresse et autres traductions d'adresses.

Ces options de reception surchargées sont implémentées sur le serveur SMTP lui-même ou passées au serveur cleanup.

- "-o smtpd_xxx_restrictions" et "-o mynetworks=127.0.0.0/8" surchargent les paramètres du fichier main.cf. Ils désactivent les contrôles anti-spam inutiles ici.
- Avec "-o smtpd_authorized_xforward_hosts=127.0.0.0/8", le transport "scan" essayera de transférer le nom et l'adresse originaux du client, ainsi les messages filtrés seront logués avec les réels caractéristiques du client. Reportez-vous aux pages XFORWARD_README et smtpd(8) pour plus d'informations.

Performance du filtrage avancé de contenu

Avec l'approche "sandwich" de filtrage de contenu décrite ici, il est important de faire correspondre le nombre de filtres concurrentiels à la CPU, la mémoire et les ressources entrées/sorties disponibles. Trop peu de

processus filtres encombre la file d'attente active même avec peu de volume de trafic ; à l'inverse, trop de processus filtres risque d'interrompre la livraison au filtre à cause d'un manque de ressources.

Actuellement, l'optimisation des performances de filtrage est un processus difficile et générateur d'erreurs ; l'analyse est faussée car les messages filtrés et non filtrés partagent la même file d'attente. Comme indiqué dans l'introduction, le filtrage de contenu avec de multiples instances de Postfix sera décrit dans une prochaine version de ce document.

Désactiver le filtrage avancé de contenu

Pour désactiver le filtrage de contenu :

- Effacez les deux lignes suivantes du fichier main.cf. Les autres changements fait pour activer le filtrage seront sans effet après.

```
/etc/postfix/main.cf :
    content_filter = scan:localhost:10025
    receive_override_options = no_address_mappings
```

- lancez "**postsuper -r ALL**" pour supprimer les informations de filtrage de contenu des fichiers déjà en file d'attente.
- relancez "**postfix reload**".

Filtrer le courrier des utilisateurs extérieurs seulement

L'approche la plus simple est de configurer UNE instance de Postfix avec de multiples adresses IP de serveurs SMTP dans le fichier master.cf :

- Deux adresses IP de serveur SMTP pour le courrier interne uniquement avec le filtrage de contenu désactivé.

```
/etc/postfix/master.cf :
# =====
# service      type  private unpriv  chroot  wakeup  maxproc command
#               (yes)  (yes)   (yes)   (never) (100)
# =====
1.2.3.4:smtp    inet  n       -       n       -       -       smtpd
-o smtpd_client_restrictions=permit mynetworks,reject
127.0.0.1:smtp  inet  n       -       n       -       -       smtpd
-o smtpd_client_restrictions=permit mynetworks,reject
```

- Une adresse de serveur SMTP pour le courrier venant de l'extérieur uniquement avec filtrage actif.

```
/etc/postfix/master.cf :
# =====
# service      type  private unpriv  chroot  wakeup  maxproc command
#               (yes)  (yes)   (yes)   (never) (100)
# =====
1.2.3.5:smtp    inet  n       -       n       -       -       smtpd
-o content_filter=xxx:yyy
-o receive_override_options=no_address_mappings
```

Avec cette configuration, vous pouvez suivre la même procédure que présentée dans les exemples de filtrage "simple" ou "avancé" si ce n'est que vous ne devez pas renseigner les paramètres "content_filter" et "receive_override_options" dans le fichier main.cf.

Filtres différents par domaine

Si vous êtes un fournisseur MX et voulez appliquer différents filtres suivant les domaines, vous pouvez configurer UNE instance de Postfix avec de multiples adresses IP de serveur SMTP dans le fichier master.cf. Chaque adresse fournit un service de filtrage différent.

```
/etc/postfix/master.cf :
# =====
# service      type  private unpriv  chroot  wakeup  maxproc command
#                (yes)   (yes)   (yes)   (never) (100)
# =====
# service SMTP pour les domaines filtrés par xxx:yyy
1.2.3.4:smtp inet n      -      n      -      -      smtpd
        -o content_filter=xxx:yyy
        -o receive_override_options=no_address_mappings

# service SMTP pour les domaines filtrés par zzz:www
1.2.3.5:smtp inet n      -      n      -      -      smtpd
        -o content_filter=zzz:www
        -o receive_override_options=no_address_mappings
```

Avec cette configuration, vous pouvez suivre la même procédure que présentée dans les exemples de filtrage "simple" ou "avancé" si ce n'est que vous ne devez pas renseigner les paramètres "content_filter" et "receive_override_options" dans le fichier main.cf.

Configurez les enregistrements MX du service DNS pour router chaque domaine sur sa propre instance de serveur SMTP.

Actions FILTER dans les tables d'accès ou d'en-tête/contenu

Les configurations de filtrages présentées jusqu'ici sont statiques. Les messages suivant un chemin donné sont soit toujours filtrés soit jamais. Depuis la version 2.0 de Postfix, vous pouvez activer le filtrage de contenu au fil de l'eau.

Pour activer le filtrage de contenu avec une table de règles d'accès(5) :

```
/etc/postfix/access:
whatever          FILTER xxx:yyy
```

Pour activer le filtrage de contenu avec une table de correspondances header_checks(5) ou body_checks(5) :

```
/etc/postfix/header_checks:
/whatever/        FILTER xxx:yyy
```

Vous également renseigner ainsi toutes les tables d'accès du serveur smtpd et les tables d'examen d'en-têtes/de contenu du serveur cleanup. Cette fonctionnalité doit être utilisée avec soin : vous devez désactiver toutes les fonctionnalités anti-spam dans les serveurs smtpd et cleanup après-filtrage, sinon vous créerez une boucle de message.

Limites :

Documentation de Postfix en français

- Les actions FILTER des tables d'accès et d'examen d'en-têtes/de contenu ont la préférence sur les filtres définis dans le paramètre content_filter du fichier main.cf.
- Si un message déclenche plusieurs actions de filtrage, seule la dernière sera prise en compte.
- Le même filtrage est appliqué à tous les destinataires du même message.

Filtrage de contenu avant mise en

file d'attente avec Postfix

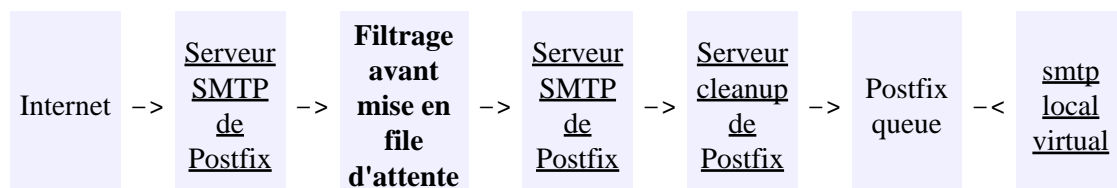
ATTENTION ATTENTION ATTENTION

Le filtrage de contenu avant mise en file d'attente décrit ici n'est utilisable que sur les sites à faible trafic. Reportez-vous au paragraphe "[Pour et contre](#)" de ce document pour plus de détails

La fonctionnalité de filtrage de contenu avant mise en file d'attente de Postfix

Depuis la version 2.1, le serveur SMTP de Postfix peut transférer tout le courrier entrant à un serveur mandataire de filtrage de contenu qui inspecte les messages AVANT leur mise en file d'attente.

Le filtrage de contenu avant mise en file d'attente est censé être utilisé comme suit :



Le filtrage de contenu avant mise en file ne doit pas être confondu avec l'approche du filtrage décrite à la page [FILTER_README](#) où le courrier est filtré APRÈS la mise en file d'attente.

Ce document aborde les sujets suivants :

- [Principes de l'opération](#)
- [Pour et contre le filtrage de contenu avant mise en file d'attente](#)
- [Configurer le dispositif proxy SMTP de Postfix](#)
- [Paramètres de configuration](#)
- [Comment Postfix dialogue avec le filtre avant mise en file d'attente](#)

Principes de l'opération

Le serveur SMTP de Postfix avant-filtrage reçoit le courrier de l'Internet et effectue les contrôles d'accès et de relais habituels, l'authentification SASL, les consultations des listes noires, le rejet des expéditeurs et destinataires inexistantes, etc. Le filtre avant mise en file d'attente reçoit le contenu d'un message non filtré et effectue l'une des opérations suivantes :

1. Ré-injecte le message dans Postfix par SMTP, éventuellement après en avoir modifié le contenu et/ou la destination.

2. Rejette le message en renvoyant un code de statut SMTP à Postfix. Ce dernier transmet ce statut au client distant. Ainsi, Postfix n'a pas à générer de message de renvoi.

Le serveur SMTP après-filtrage de Postfix reçoit les messages filtrés. A partir de là, Postfix procède comme d'habitude.

Le filtre avant mise en file d'attente décrit ici fonctionne comme le filtre après mise en file d'attente décrit à la page [FILTER_README](#). Dans la plupart des cas, vous pourrez utiliser le même logiciel avec les limites discutées au paragraphe "[Pour et contre](#)" ci-dessous.

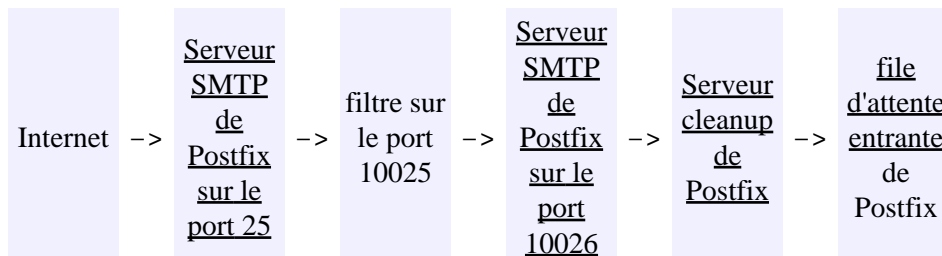
Pour et contre le filtrage de contenu avant mise en file d'attente

- Pour : Postfix peut rejeter le courrier avant son transfert en file d'attente, ainsi Postfix n'a pas à renvoyer les messages à l'expéditeur (qui est généralement forgé). Les messages non acceptés restent à la charge du client SMTP distant.
- Contre : le client SMTP distant attend une réponse SMTP dans un temps donné. A mesure que la charge de votre système augmente, de moins en moins de cycles de CPU sont disponibles pour répondre avant la limite et vous devrez soit arrêter de recevoir le courrier soit cesser de le filtrer. C'est pour cette raison que ce filtrage ne doit être utilisé que sur des sites à trafic modéré.
- Contre : le logiciel de filtrage de contenu peut utiliser beaucoup de mémoire. Pour éviter un dépassement de mémoire vous devez réduire le nombre de processus SMTP avant-filtrage, ce qui évitera de planter votre système. Ceci suppose que vos clients SMTP devront attendre parfois assez longtemps la remise des messages.

Configurer le dispositif proxy SMTP de Postfix

Dans l'exemple suivant, le serveur SMTP avant-filtrage de Postfix transmet les messages à un filtre de contenu en écoute sur le port 10025 de la machine locale. Le serveur SMTP après-filtrage reçoit les messages filtrés sur le port 10026. A partir de là, le traitement fonctionne comme d'habitude.

Le filtre de contenu lui-même n'est pas décrit ici. Vous pouvez utiliser tous les filtres compatibles SMTP. Pour ceux qui ne peuvent dialoguer avec le protocole SMTP, le proxy de Bennett Todd's implémente un bon framework de filtrage de contenu PERL/SMTP. Reportez-vous à la page <http://bent.latency.net/smtpprox/>.



Ceci est configuré en éditant le fichier master.cf :

```
/etc/postfix/master.cf:
# =====
# service type private unpriv chroot wakeup maxproc command
#               (yes)   (yes)   (yes)   (never) (100)
# =====
```

Documentation de Postfix en français

```
#
# Serveur SMTP avant-filtrage. Reçoit le courrier du réseau et
# le passe au filtre de contenu sur le port 10025.
#
smtp      inet  n       -       n       -       20       smtpd
        -o smtpd_proxy_filter=127.0.0.1:10025
        -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=127.0.0.0/8
        -o smtpd_client_restrictions=
        -o smtpd_helo_restrictions=
        -o smtpd_sender_restrictions=
        -o smtpd_recipient_restrictions=permit_mynetworks,reject
        -o smtpd_data_restrictions=
        -o mynetworks=127.0.0.0/8
        -o receive_override_options=no_unknown_recipient_checks
```

Note : n'utilisez pas d'espaces, autour des caractères "=" et ",".

L'entrée du serveur SMTP avant-filtrage est une version modifiée de celle du serveur SMTP par défaut, normalement configuré au début du fichier master.cf :

- Le nombre de sessions SMTP est réduit de la valeur 100 (par défaut) à 20 seulement pour éviter le plantage du système.
- L'option "-o smtpd_client_connection_count_limit=10" évite à un client SMTP d'utiliser les 20 processus du serveur SMTP. Cette limite n'est pas nécessaire si vous recevez tout le courrier d'un relais approuvé.

Note : ce paramètre est ignoré sur la version stable 2.1 de Postfix. Cette fonctionnalité ne sera disponible que sur les versions expérimentales jusqu'à la sortie de la version 2.2.

- L'option "-o smtpd_proxy_filter=127.0.0.1:10025" indique au serveur SMTP avant-filtrage qu'il doit transférer le courrier entrant au filtre de contenu écoutant sur le port 10025.
- Postfix 2.3 support à la fois les filtres TCP et du domaine UNIX. Le filtre exemple ci-dessus pourrait être indiqué ainsi : "inet:127.0.0.1:10025". Pour indiquer un filtre dans le domaine UNIX, indiquez "unix:chemin". Un chemin relatif est interprété à partir du répertoire des files d'attente de Postfix.

Le serveur SMTP après-filtrage est une nouvelle entrée du fichier master.cf :

- ":10026" le met en écoute sur la boucle locale uniquement sans s'exposer au réseau. Ne mettez jamais le serveur après-filtrage en écoute sur Internet :—)
- L'option "-o smtpd_authorized_xforward_hosts=127.0.0.0/8" permet au serveur SMTP après-filtrage de recevoir les informations du client SMTP distant du serveur SMTP avant-filtrage, ainsi les démons de Postfix travaillant après le filtrage peuvent enregistrer les propriétés du client et non localhost[127.0.0.1].
- Les autres paramètres du serveur SMTP après-filtrage évitent la duplication du travail qui a été déjà effectué par le serveur SMTP avant-filtrage.

Par défaut, le filtre dispose de 100 secondes pour faire son travail. En cas de dépassement, Postfix reporte une erreur au client SMTP distant. Vous pouvez augmenter cette limite (reportez-vous au paragraphe ci-dessous) mais vous risquez de dépasser la limite accordée par le client.

Paramètres de configuration

Paramètres qui contrôlent le mandatement :

- smtpd_proxy_filter (syntaxe : hôte:port) : la machine et le port TCP du filtre de contenu avant mise en file d'attente. Si aucun hôte ou hôte: n'est spécifié la valeur retenue est la machine locale.
- smtpd_proxy_timeout (défaut: 100s): temps limite pour se connecter au filtre de contenu avant mise en file d'attente et pour envoyer et recevoir les commandes et les données. Toutes les erreurs du proxy sont loguées. Pour des raisons de confidentialité, tout ce que voit le client SMTP distant est "451 Error: queue file write error". Il ne serait pas bon de divulguer plus de détails aux étrangers.
- smtpd_proxy_ehlo (défaut: \$myhostname) : le nom de machine à utiliser lors de l'envoi de la commande EHLO au filtre de contenu.

Comment Postfix dialogue avec le filtre avant mise en file d'attente

Le serveur SMTP avant-filtrage de Postfix se connecte au filtre de contenu, délivre un message et se déconnecte. Lorsqu'il envoie le message au filtre, Postfix dialogue avec le protocole ESMTP mais n'utilise pas de commande pipeling. Postfix génère ses propres commandes EHLO, XFORWARD (pour loguer l'adresse IP du client et non localhost[127.0.0.1]), DATA et QUIT, et transfère des copies non modifiées de tous les commandes MAIL FROM et RCPT TO qu'il n'a pas rejeté lui-même. Postfix n'envoie pas d'autres commandes SMTP.

Le filtre de contenu doit accepter les mêmes syntaxes des commandes MAIL FROM et RCPT TO que le serveur SMTP de Postfix et doit transmettre ces mêmes commandes sans modifications au serveur SMTP aval situé après le filtre. Si le filtre de contenu ou le serveur SMTP aval ne supportent pas toutes les fonctionnalités ESMTP que le serveur SMTP Postfix amont, vous devez désactiver les fonctionnalités concernées avec le paramètre smtpd_discard_ehlo_keywords.

Lorsque le filtre rejette un contenu, il devrait renvoyer une réponse négative au serveur SMTP Postfix amont et couper la connexion avec le serveur SMTP aval sans terminer la conversation SMTP.

Support Milter de Postfix avant

mise en file d'attente

Introduction

Postfix version 2.3 introduit le support des applications Milter (mail filter) de Sendmail 8. Ces applications fonctionnent en dehors du MTA pour inspecter les événements SMTP (connect, helo, mail from, etc.) et le contenu du message. Une application Milter peut ordonner au MTA d'accepter, rejeter, mettre en attente ou en quarantaine une connexion, une commande ou le contenu d'un message ; d'effacer ou d'ajouter un destinataire ou en en-tête de message ; et de remplacer un en-tête de message ou son contenu entier. Tout ceci se passe avant la mise en file d'attente.

Le support de Milter a été ajouté à Postfix car il existe une grande collection d'applications, pas seulement pour bloquer les messages non souhaités, mais également pour vérifier l'authenticité (exemples: SenderID+SPF et Domain keys) ou pour signer les messages (exemple: Domain keys). Inventer un autre dispositif pour tous ces logiciels serait un piètre usage des ressources humaines.

Postfix 2.3 implemente toutes les requêtes du protocole Milter de Sendmail 8 jusqu'à la version 4, à l'exception d'une : le remplacement du contenu d'un message. Reportez-vous à ce propos au paragraphe, limites à la fin de ce document.

Ce document aborde les sujets suivants :

- Compiler les applications Milter
- Lancer les applications Milter
- Configurer Postfix
- Contournement des problèmes
- Limitations

Compiler les applications Milter

Bien que les applications Milter peuvent être écrites en C, JAVA or Perl, ce texte ne parle que des applications écrites en C. Pour celles-ci, vous avez besoin d'une librairie qui implemente the protocol Milter. Postfix ne fournit pas actuellement une telle librairie.

Sur certaines distributions récentes Linux et *BSD, la librairie libmilter de Sendmail est installée par défaut. Les applications telles dk-milter et sid-milter sont compilées ainsi :

```
$ gzcac dk-milter-x.y.z.tar.gz | tar xf -
$ cd dk-milter-x.y.z
$ make
[...nombreuses lignes affichées...]
```

Sur les autres plateformes, vous avez deux options :

- Installer la librairie **libmilter** de Sendmail et les fichiers "include" correspondants, puis compiler les applications Milter. Sur les systèmes Linux, **libmilter** fait généralement partie du package **sendmail-devel**.
- Ne pas installer les librairies **libmilter** de Sendmail, mais utiliser les librairies libmilter compilées depuis le code source de Sendmail :

```
$ gzcac sendmail-x.y.z.tar.gz | tar xf -
$ cd sendmail-x.y.z
$ make
[...nombreuses lignes affichées...]
```

Après ceci, suivez les instructions d'installation incluses dans les sources de l'application Milter pour indiquer l'emplacement des fichiers "include" et de la librairie libmilter. Généralement, ces paramètres se configurent dans un fichier nommé `sid-filter/Makefile.m4` ou équivalent :

```
dnl APPENDEF(`confINCDIRS', ` -I/some/where/sendmail-x.y.z/include')
dnl APPENDEF(`confLIBDIRS', ` -L/some/where/sendmail-x.y.z/obj.systemtype/libmilter')
```

Compilez ensuite l'application Milter.

Lancer les applications Milter

Pour lancer une application Milter, reportez-vous à la documentation sur les filtres pour les options. Une ligne de commande typique ressemble à :

```
$ /chemin/vers/dk-filter -p inet:numeroDePort@localhost ...autres options...
```

Configurer Postfix

Comme Sendmail, Postfix dispose de nombreuses options de configuration qui contrôlent son dialogue avec les applications Milter. Initialement, de nombreuses options sont globales, c'est à dire qu'elles s'appliquent à toutes les applications Milter. Les versions futures de Postfix pourront supporter des timeouts, contrôles d'erreurs, etc. différenciés par Milter.

Contenu de ce paragraphe :

- [Applications Milter](#)
- [Interprétation des erreurs Milters](#)
- [Version du protocole Milter](#)
- [Timeouts du protocole Milter](#)
- [Macro émulation de Sendmail](#)

Applications Milter

Le protocole Milter a été initialement développé pour bloquer les messages indésirables arrivant du réseau. A cet effet, Postfix utilise les applications Milter listée dans le paramètre `smtpd_milters`. Vous devez indiquer les applications Milter par le nom de leur socket en écoute (les autres options Milter sont présentées ci-après). L'exemple ci-dessous montre une seule application Milter, mais Postfix peut en accepter plusieurs. Ils sont appelés dans l'ordre indiqué.

```
/etc/postfix/main.cf:
```

Documentation de Postfix en français

```
# Milters pour le courrier qui arrive via le serveur smtpd(8).  
# Voir ci-après pour la syntaxe des adresses de socket.  
smtpd_milters = inet:localhost:numéroDePort ...autres filtres...
```

Désormais, des applications Milter sont également développées pour signer des messages afin que les systèmes détectent le spam, le phishing, etc. Utiliser de tels Milters de signature est assez aisé pour le courrier qui arrive par SMTP, mais pas pour les soumissions arrivant par la commande sendmail de Postfix.

Pour rendre possible le filtrage Milter des messages non-SMTP, Postfix utilise les applications Milter listées dans le paramètre cleanup_milters. Ceci permet la signature des messages issus de la commande sendmail.

```
/etc/postfix/main.cf:  
# Milters pour les messages non-SMTP.  
# Voir ci-après pour la syntaxe des adresses de socket.  
cleanup_milters = inet:localhost:numéroDePort ...autres filtres...
```

Pour que les applications Milter acceptent les messages non-SMTP, le serveur cleanup(8) simule les événements de connexion SMTP, ehlo, mail from, rcpt to, et data, comme si le message arrivait en ESMTP depuis "localhost" avec un adresse IP "127.0.0.1". Toutefois, comme il n'y a pas réellement de client ESMTP, les applications Milter ne doivent pas rejeter ces événements SMTP simulés. Si elles le font, Postfix rapportera une erreur de configuration, mais aucun message ne sera pas perdu.

La syntaxe générale pour les sockets en écoute est :

unix:chemin

Se connect au serveur dans le domaine UNIX indiqué par le chemin. Si les processus smtpd(8) ou cleanup(8) fonctionnent en cage chroot, les chemins absolus seront interprétés relativement au répertoire des files d'attentes de Postfix.

inet:machine:port

Se connect au port TCP indiqué sur la machine indiquée (locale ou distante). Les machines et ports peuvent être indiqués sous la forme chiffrée ou symbolique.

Note : La syntaxe de Postfix diffère de la syntaxe Milter qui à la forme ***inet:port@host***.

Interprétation des erreurs Milters

Le paramètre milter default action indique la façon dont Postfix interprète les erreurs. Par défaut, il répond avec un statut d'erreur temporaire, ainsi le client réessaiera plus tard. Indiquez "accept" si vous voulez recevoir le courrier comme si le filtre n'existait pas, et "reject" pour rejeter le courrier avec un statut d'erreur permanent.

```
# Que faire en cas d'erreur ? Indiquez accept, reject, ou tempfail.  
milter default action = tempfail
```

Version du protocole Milter

Comme Postfix n'est pas compilé avec la librairie **libmilter** de Sendmail, vous devez également indiquer la version du protocole Milter que Postfix doit utiliser. La version par défaut est la n°2.

```
milter protocol = 2
```

Si le paramètre milter_protocol est trop petit, vous verrez une erreur du type :

```
application name: st_optionneg[xxxxx]: 0xyy does not fulfill action requirements 0xzz
```

Le remède est d'augmenter le numéro de version milter_protocol. Reportez-vous toutefois au paragraphe limitations ci-dessous pour les fonctionnalités non supportées par Postfix.

Si la paramètre milter_protocol est trop élevé, l'application Milter se déconnecte simplement, et vous verrez un message d'erreur du type :

```
postfix/smtpd[21045]: warning: milter inet:host:port: can't read packet header: Unknown er
```

Le remède est d'abaisser le numéro de version milter_protocol.

Timeouts du protocole Milter

Postfix utilise différentes limites de temps aux différentes étapes du protocole Milter. La table présentée ci-dessous montre quels délais limites sont utilisés et quand (FDET = fin des en-têtes; FDM = fin du message).

Paramètre	Délai limite	Étape du protocole
<u>milter_connect_timeout</u>	10s	CONNECT
<u>milter_command_timeout</u>	10s	HELO, MAIL, RCPT, DATA, UNKNOWN
<u>milter_content_timeout</u>	100s	HEADER, FDET, BODY, FDM

Attention : 10s est un délai court pour les applications qui effectuent plusieurs interrogations DNS. Toutefois, si vous augmentez trop les délais ci-dessus, les clients SMTP distants peuvent se déconnecter et le message peut être livré plusieurs fois. C'est un problème inhérent au filtrage avant mise en file d'attente.

Macro émulation Sendmail

Postfix émule un nombre limité de macros Sendmail, comme indiqué dans le tableau ci-dessous. Différentes macros sont disponibles aux différentes étapes du protocole (FDM = fin-du-message) ; leur disponibilité n'est pas toujours la même que dans Sendmail. Reportez-vous au paragraphe "contournement des problèmes" ci-dessous pour les solutions.

Nom	Disponibilité	Description
i	DATA, FDM	Identifiant de queue
j	Toujours	valeur de <u>myhostname</u>
{auth_authen}	MAIL, DATA, FDM	nom de login SASL
{auth_author}	MAIL, DATA, FDM	expéditeur SASL
{auth_type}	MAIL, DATA, FDM	méthode de login SASL
{client_addr}	Toujours	Adresse IP du client
{client_connections}	CONNECT	Nombre de connexions concurrentes pour ce client
{client_name}	Toujours	Nom de machine du client, "unknown" lorsque la consultation ou la vérification

		échoue
{client_ptr}	CONNECT, HELO, MAIL, DATA	Nom du client issu de la consultation DNS inverse, "unknown" lorsque la consultation échoue
{cert_issuer}	HELO, MAIL, DATA, FDM	Fournisseur du certificat TLS client
{cert_subject}	HELO, MAIL, DATA, FDM	"Subject" du certificat TLS client
{cipher_bits}	HELO, MAIL, DATA, FDM	Taille de la clef de session TLS
{cipher}	HELO, MAIL, DATA, FDM	chiffrement TLS
{daemon_name}	Toujours	valeur de <code>milter_daemon_name</code>
{mail_addr}	MAIL	Adresse d'expédition
{rcpt_addr}	RCPT	Adresse de destination
{tls_version}	HELO, MAIL, DATA, FDM	Version du protocole TLS

Postfix envoie des ensembles spécifiques de macros aux différentes étapes du protocole. Les ensembles sont configurés par les paramètres présentés dans le tableau ci-dessous.

Nom du paramètre	Version de protocole	Protocol stage
<code>milter_connect_macros</code>	2 ou plus	CONNECT
<code>milter_helo_macros</code>	2 ou plus	HELO/EHLO
<code>milter_mail_macros</code>	2 ou plus	MAIL FROM
<code>milter_rcpt_macros</code>	2 ou plus	RCPT TO
<code>milter_data_macros</code>	4 or higher	DATA
<code>milter_end_of_data_macros</code>	2 ou plus	FDM
<code>milter_unknown_command_macros</code>	3 ou plus	commande inconnue

Contournement des problèmes

Les applications Milter Sendmail ont été initialement développées pour la version 8 du MTA Sendmail, dont l'architecture est différente de celle de Postfix. En conséquence, certaines applications Milter font des requêtes qui ne sont pas disponibles en dehors de l'environnement Sendmail.

- Certaines applications Milter enregistrent un avertissement du type :

```
sid-filter[36540]: WARNING: sendmail symbol 'i' not available
```

et elles peuvent insérer un en-tête de message avec un "unknown-msgid" :

```
X-SenderID: Sendmail Sender-ID Filter vx.y.z machine.com <unknown-msgid>
```

Ceci arrive car les application Milter supposent que l'identifiant (queue ID) est connu *avant* que le MTA n'accepte la commande MAIL FROM (expéditeur). Postfix, d'autre part, ne crée pas de fichier en file d'attente *tant que* Postfix n'a pas accepté la première commande RCPT TO (destinataire).

Documentation de Postfix en français

Pour éviter cet en-tête de message, nous ajoutons un petit bout de code aux sources du Milter pour rechercher l'identifiant de file d'attente après que Postfix n'ait reçu la fin du message.

- ◆ Editez le fichier source du filter (généralement nommé `dk-filter/dk-filter.c` ou équivalent).
- ◆ Cherchez la fonction `mlfi_eom()` et ajoutez le code en **gras** ci-dessous vers le début du code de la fonction tel qu'indiqué ci-dessous :

```
cc = (connctx) smfi_getpriv(ctx);
assert(cc != NULL);

/*
** Determine the job ID for logging.
*/
if (sic->ctx_jobid == 0 || strcmp(sic->ctx_jobid, MSGIDUNKNOWN) == 0) {
    char *jobid = smfi_getsymval(ctx, "i");
    if (jobid != 0)
        sic->ctx_jobid = jobid;
}
```

Ceci ne supprime toutefois pas le message d'avertissement (WARNING).

Avec certaines applications Milters on peut régler l'avertissement et le "unknown-msgid" en repoussant l'appel de `mlfi_eoh()` (ou d'une autre routine générant l'avertissement) à la fin du message.

- ◆ Éditez le fichier source du filtre (généralement nommé `sid-filter/sid-filter.c` ou équivalent).
- ◆ Cherchez le tableau `smfilter` et remplacez `mlfi_eoh` (ou tout autre routine générant l'avertissement) par NULL.
- ◆ Cherchez la fonction `mlfi_eom()` et ajoutez le code en **gras** présenté ci-dessous au début du code avant l'appel à `mlfi_eoh()` :

```
assert(ctx != NULL);
#endif /* !DEBUG */

ret = mlfi_eoh(ctx);
if (ret != SMFIS_CONTINUE)
    return ret;
```

Ceci fonctionne avec `sid-milter-0.2.10`. D'autres applications Milters planteront si vous le faites.

Limites

Ce paragraphe liste les limites de l'implémentation Milter de Postfix. Certaines de ces limites seront corrigées au fur et à mesure que le support progresse. Bien sûr, les habituelles limitations du filtrage avant mise en file d'attente resteront vraies. Reportez-vous à la page [Inspection du contenu](#) pour plus de détails.

- Postfix ne supporte actuellement que les applications qui utilisent les versions 2 à 4 du protocole Milter de Sendmail 8. Le support pour les versions pourra être ajouté plus tard.
- Pour les applications écrites en C, vous devez utiliser la librairie `libmilter` de Sendmail. Une librairie Postfix pourra être fournie en remplacement dans le futur.
- Lorsque le message n'arrive pas via le serveur [smtpd\(8\)](#), le serveur [cleanup\(8\)](#) simule les événements SMTP de connexion, helo, mail from, rcpt to, et data. Toutefois, les applications Milters ne doivent pas rejeter ces événements SMTP simulés. Dans le cas contraire, Postfix rapportera une erreur de

configuration.

- Postfix n'applique pas actuellement le filtrage de contenu aux messages qui sont transférés par alias ou autre, ou générés en interne tels les notifications de rejet. Ceci peut poser problème lorsque vous souhaitez utiliser un Milter pour signer de tels messages.
- Lorsque vous utilisez un filtre avant mise enfile d'attente pour le courrier entrant par SMTP (voir [SMTPD_PROXY_README](#)), les applications Milters n'ont accès qu'aux informations issues des commandes SMTP ; elles n'ont pas accès aux en-têtes ou au contenu du message, et ne peuvent faire de modifications au message ou à l'enveloppe.
- Postfix 2.3 ne supporte pas les requêtes Milter remplaçant le corps du message. Les applications Milters qui le resuierent enregistreront un avertissement dans les journaux du type :

```
application name: st_optionneg[134563840]: 0x3d does not fulfill action requirements
```

La seule solution est d'utiliser (d'attendre) une version de Postfix qui supporte cette fonctionnalité manquante.

- La plupart des options de configuration Milter sont globales. Les versions futures de Postfix pourront proposer des timeouts, des contrôles d'erreurs, etc. par Milter.

Contrôle d'accès et de relais SMTP

avec Postfix

Introduction

Le serveur SMTP de Postfix reçoit le courrier depuis le réseau et est exposé à tous les virus et pourriels du l'Internet. Ce document présente les méthodes internes et externes qui contrôle quels messages SMTP accepter, quelles erreurs éviter, et comment tester votre configuration.

Sujets abordés dans ce document :

- Contrôle de relais, de pourriel et politique par utilisateur
- Restrictions à appliquer à tous les messages SMTP
- Sélectivité avec les listes de restriction d'accès
- Évaluation à posteriori des listes de restriction d'accès
- Utilisation dangereuse du paramètre `smtpd_recipient_restrictions`
- Tester les règles d'accès SMTP

Contrôle de relais, de pourriel et politique par utilisateur

A sa création, l'Internet était un réseau amical. Les serveurs de messagerie transféraient le courrier de n'importe qui vers n'importe quelle destination. Aujourd'hui, les spammers abusent des serveurs qui continuent de transférer le courrier sans contrôle et les systèmes ainsi abusés finissent dans les listes noires anti-spammer. Consultez par exemple le sites spécialisés comme <http://www.mail-abuse.org/>.

Par défaut, Postfix a une approche restrictive du relais. Il ne transfère que le courrier des clients du réseau sûr (réseau interne) ou des domaines de la liste des relais autorisés. Pour plus de détails sur la configuration par défaut, reportez-vous au paragraphe concernant le paramètre `smtpd_recipient_restrictions` dans la page de manuel [postconf\(5\)](#) et aux informations ci-dessous.

La plupart des contrôles d'accès du serveur SMTP de Postfix sont destinés à stopper le pourriel.

- Orientés protocole : certains contrôles du serveur SMTP bloquent le courrier en étant très strict sur le respect du protocole SMTP ; ceci élimine les logiciels de spam mal configurés ou mal implémentés comme les vers qui arrivent avec une implémentation de client SMTP non standard. Ces contrôles sont inefficaces face à des spammers et vers écrits en respectant les RFC.
- Orientés listes noires : certains contrôles du serveur SMTP interrogent des listes noires de sites connus pour être nocifs comme les relais SMTP ouverts, les proxies ouverts et certaines machines compromises et sous le contrôle de pirates. L'efficacité de ces listes noires dépend de la fréquence de leurs mises à jour.
- Orienté seuil : certains contrôles du serveur SMTP n'autorisent l'accès qu'après avoir fait effectuer un travail (listes grises) au client ou lui avoir demandé un élément non prévu (vérification de l'adresse de l'expéditeur/du destinataire et SPF). Les listes grises et politiques SPF sont externalisées et font l'objet de la page [SMTPD_POLICY_README](#). La vérification des adresses d'expédition/de destination font

l'objet de la page [ADDRESS VERIFICATION README](#).

Malheureusement, tous les contrôles peuvent parfois rejeter du courrier légitime. Ceci peut être un problème sur des sites avec des profils d'utilisateurs différents : certains trouveront inacceptable de recevoir un pourriel alors que pour d'autres la perte d'un message prend un caractère de fin du monde. Comme il n'y a pas de politique parfaite pour tous, Postfix supporte des restriction d'accès différentes par utilisateurs. Cette caractéristique est décrite à la page [RESTRICTION CLASS README](#).

Restrictions à appliquer à tous les messages SMTP

Outre les restrictions pouvant être configurées par utilisateur, Postfix en implémente quelques unes qui s'appliquent à tous les messages SMTP.

- Les restrictions internes sur le contenu [header checks](#) et [body checks](#) décrites à la page [BUILTIN FILTER README](#). Ceci s'applique lorsque Postfix reçoit le courrier avant stockage dans la [file d'attente entrante](#).
- Les restrictions sur le contenu à la volée et externalisées décrites à la page [SMTPD_PROXY README](#). Ceci s'applique lorsque Postfix reçoit le courrier avant stockage dans la [file d'attente entrante](#).
- L'obligation pour le client d'utiliser la commande HELO ou EHLO avant MAIL FROM ou ETRN. Ceci peut poser problème avec les applications maison qui envoient du courrier. Pour cette raison, cette fonctionnalité est désactivée par défaut ("[smtpd_helo_required](#) = no").
- Interdire les syntaxes non conformes dans les commandes MAIL FROM ou RCPT TO. Ceci peut poser problème avec les applications maison qui envoient du courrier et avec les vieux clients de messagerie. Pour cette raison, cette fonctionnalité est désactivée par défaut ("[strict_rfc821_envelopes](#) = no").
 - ◆ Interdire la syntaxe d'adresses [RFC 822](#) (exemple: "MAIL FROM: the dude <dude@example.com>").
 - ◆ Interdire les adresses non encapsulées dans <> (exemple: "MAIL FROM: dude@example.com").
- Rejeter le courrier provenant d'une adresse inexistante. Cette forme de filtrage aide à ralentir les vers et autres logiciels malfaisant mais peut poser problème avec les applications maison qui envoient du courrier avec une adresse inexistante. Pour cette raison, cette fonctionnalité est désactivée par défaut ("[smtpd_reject_unlisted_sender](#) = no").
- Rejeter le courrier à destination d'une adresse inexsistante. Cette forme de filtrage aide à garder la file d'attente vide des messages non livrables MAILER-DAEMON messages. Cette fonctionnalité est activée par défaut ("[smtpd_reject_unlisted_recipient](#) = yes").

Sélectivité avec les listes de restriction d'accès

Postfix vous permet de définir des listes de restrictions d'accès pour chaque type de session SMTP. Ces restrictions individuelles sont décrites à la page de manuel [postconf\(5\)](#).

Des exemples de listes de restrictions simples:

```
/etc/postfix/main.cf:
# Autorise les connexions depuis le réseau sûr seulement.
smtpd\_client\_restrictions = permit\_mynetworks, reject

# Ne pas communiquer avec les systèmes qui ne connaissent pas leur propre nom de machine.
```

Documentation de Postfix en français

```
# Avec Postfix < 2.3, utilisez reject unknown hostname
smtpd helo restrictions = reject unknown helo hostname

# Ne pas accepter de courrier des domaines qui n'existent pas.
smtpd sender restrictions = reject unknown sender domain

# Liste blanche: les clients locaux peuvent indiquer n'importe quelle destination, pas les au
smtpd recipient restrictions = permit mynetworks, reject unauth destination

# Bloquer les clients qui parlent trop tôt
smtpd data restrictions = reject unauth pipelining

# Contrôle les quotas de volume de message via un appel au service de politique
smtpd data end of restrictions = check policy service unix:private/policy
```

Chaque liste de restrictions est évaluée de la gauche vers la droite jusqu'à ce qu'une restriction produise un résultat parmi PERMIT, REJECT ou DEFER (essayer plus tard). La fin de liste est équivalente à PERMIT. En plaçant une restriction PERMIT avant une restriction REJECT vous pouvez faire des exceptions pour des clients ou utilisateurs particuliers. C'est ce qu'on appelle liste blanche ; le dernier exemple ci-dessus autorise le courrier depuis le réseau local mais rejete autrement les destinations arbitraires.

Le tableau ci-dessous résume les bouts poursuivis par chaque liste de restriction d'accès. Elles utilisent toutes la même syntaxe et ne diffèrent que par l'instant où elles sont appelées et l'effet d'un résultat REJECT ou DEFER.

Nom de liste de restriction	Status	Effet d'un REJECT ou DEFER
<u>smtpd client restrictions</u>	Optionel	Rejete toutes le commandes du client
<u>smtpd helo restrictions</u>	Optionel	Rejete les informations HELO/EHLO
<u>smtpd sender restrictions</u>	Optionel	Rejete l'information MAIL FROM
<u>smtpd recipient restrictions</u>	Obligatoire	Rejete l'information RCPT TO
<u>smtpd data restrictions</u>	Optionel	Rejete la commande DATA
<u>smtpd data end of restrictions</u>	Optionel	Rejete la commande END-OF-DATA
<u>smtpd etrn restrictions</u>	Optionel	Rejete la commande ETRN

Evaluation différée des listes de restriction d'accès SMTP

Les versions précédente de Postfix evaluaient les listes de restriction d'accès SMTP aussi vite que possible. La restriction basée sur le client était ainsi évaluée avant l'envoi de la bannière d'accueil "220 \$myhostname...", la restriction basée sur la commande HELO (EHLO) avant la réponse HELO (EHLO) et la restriction basée sur la commande MAIL FROM avant la réponse, etc. Cette approche devient difficile à utiliser.

La version actuelle de Postfix retarde l'évaluation du client, du HELO et des restrictions sur l'expéditeur jusqu'à la commande RCPT TO ou ETRN. Ce comportement est contrôlé par le paramètre smtpd delay reject. Les listes sont tout de même évaluées dans le bon ordre (client, helo, etrn) ou (client, helo, expéditeur, destinataire, data). Lorsqu'une de ces listes retourne REJECT ou DEFER, les suivantes sont pas évaluées.

A l'époque où le paramètre smtpd delay reject fut introduit, Postfix fut également modifié pour supporter des listes de restriction combinant les informations sur le client, la commande HELO, l'expéditeur, le destinataire et la commande ETRN.

Bénéfices de l'évaluation retardée des restriction et la combinaison des restrictions :

- Certains clients SMTP ne s'attendent pas à avoir une réponse négative en cours de session SMTP. Lorsque la réponse négative n'est envoyée qu'à la réponse RCPT TO, le client se déconnecte au lieu d'attendre jusqu'à la limite (timeout), ou pire d'entrer dans une boucle sans fin connexion–rejet–connexion.
- Postfix peut logger des informations plus utiles. Par exemple, lorsque Postfix rejete un nom de client ou une adresse mais attend la commande RCPT TO, il peut enregistrer les adresses d'expéditeur et de destination au lieu de ne logger que le nom de machine du client et l'adresse IP et ne pas savoir quel courrier a été bloqué.
- Mixing is needed for complex whitelisting policies. For example, in order to reject local sender addresses in mail from non-local clients, you need to be able to mix restrictions on client information with restrictions on sender information in the same restriction list. Without this ability, many per-user access restrictions would be impossible to express.

Utilisation dangereuse du paramètre `smtpd_recipient_restrictions`

A ce niveau le lecteur peut se demander pourquoi nous avons besoin des restrictions sur le client, HELO ou l'adresse de l'expéditeur quand leur évaluation est remise à la commande RCPT TO or ETRN. Certaines personnes recommandent de placer TOUTES les restrictions d'accès dans la liste `smtpd_recipient_restrictions`. Malheureusement, cela peut entraîner un accès trop permissif. Comment est-ce possible ?

Le but de la fonctionnalité `smtpd_recipient_restrictions` est de contrôler comment Postfix répond à la commande RCPT TO. Si la restriction répond REJECT ou DEFER, l'adresse du destinataire est rejetée; pas de surprise ici. Si le résultat est PERMIT, alors l'adresse de destination est acceptée, et c'est là qu'apparaissent les surprises.

Ci-dessous un exemple montrant qu'un résultat PERMIT peut entraîner un accès trop permissif :

```
1 /etc/postfix/main.cf:
2   smtpd_recipient_restrictions =
3       permit_mynetworks
4       check_helo_access hash:/etc/postfix/helo_access
5       reject_unknown_helo_hostname
6       reject_unauth_destination
7
8 /etc/postfix/helo_access:
9   localhost.localdomain PERMIT
```

La ligne 5 rejete le message des machines qui n'indiquent pas un nom correct dans la commande HELO (avec Postfix < 2.3, utilisez `reject_unknown_hostname`). Les lignes 4 et 9 font exception des machines s'annonçant "HELO localhost.localdomain".

Le problème de cette configuration est que le résultat de `smtpd_recipient_restrictions` est PERMIT pour toutes les machines s'annonçant comme "localhost.localdomain", faisant de Postfix un superbe relais pour de telles machines.

Pour éviter de telles surprises avec `smtpd_recipient_restrictions`, vous devez placer les restrictions ne concernant pas le destinataire APRÈS la restriction `reject_unauth_destination`, pas avant. Dans l'exemple ci-dessus, la restriction basée sur HELO devrait être placée APRÈS `reject_unauth_destination`, ou mieux, les

restrictions sur HELO devraient être placées dans le paramètre smtpd_helo_restrictions où elles ne peuvent faire aucun mal.

Tester les règles d'accès SMTP

Postfix possède différents dispositifs de test des règles d'accès :

soft bounce

C'est un filet de sécurité qui change les actions REJECT du serveur SMTP en actions DEFER (essayez plus tard). Ceci garde le message en file d'attente au lieu de le retourner à l'expéditeur. Indiquez "soft bounce = yes" dans le fichier main.cf pour prévenir les rejets permanents en changeant tous les codes SMTP 5xx en 4xx.

warn if reject

Il s'agit d'un mécanisme de protection différent qui change les actions REJECT du serveur SMTP en avertissements. Indiquez "warn if reject" dans une liste de restrictions d'accès avant la restriction que vous voulez tester sans rejeter les messages.

XCLIENT

Avec cette fonctionnalité Postfix 2.1, des clients SMTP autorisés peuvent passer pour d'autres systèmes, ainsi vous pouvez effectuer de réels tests. Des exemples de simulation d'autres systèmes pour des règles d'accès sont présentées à la fin de la page XCLIENT README.

Délégation de la politique d'accès

SMTP avec Postfix

But de la délégation de la politique d'accès SMTP

Le serveur SMTP de Postfix dispose de différents mécanismes intégrés pour bloquer ou accepter le courrier à différentes étapes du protocole SMTP. Depuis la version 2.1, Postfix peut déléguer ces décisions à un serveur extérieur à Postfix.

Avec ce mécanisme, une simple liste grise peut être implementée avec seulement une douzaine de lignes de Perl comme proposé à la fin de ce document. Un autre exemple de délégation de politique est le serveur SPF de Meng Wong disponible à l'adresse <http://spf.pobox.com/>. Des exemples de politiques peuvent être trouvées dans les sources de Postfix dans le répertoire `examples/smtpd-policy`.

La délégation de politique est maintenant la méthode préférée pour ajouter des politiques à Postfix. Il est plus aisé de développer une nouvelle fonctionnalité avec quelques lignes de Perl que d'essayer de faire la même chose en C. La différence en performance est imperceptible sauf dans les environnements les plus chargés.

Ce document couvre les sujets suivants :

- Description du protocole de politique
- Configuration de la politique client/serveur
- Exemple: serveur de politique liste grise
- Mettre en liste grise le courrier des domaines fréquemment forgés
- Mettre en liste grise tout le courrier
- Routine de maintenance des listes grises
- Exemple de serveur Perl de liste grise

Description du protocole

Le protocole de délégation de Postfix est vraiment simple. La requête client est une série de `nom=valeur` séparées par une nouvelle ligne et terminée par une ligne vide. La réponse du serveur est une ligne `nom=valeur` terminée elle aussi par une ligne vide.

Ci-dessous un exemple de tous les attributs que le serveur SMTP de Postfix envoie dans une requête de politique d'accès déléguée :

```
request=smtpd_access_policy
protocol_state=RCPT
protocol_name=SMTP
helo_name=un.domaine
queue_id=8045F2AB23
sender=foo@bar.tld
recipient=bar@foo.tld
recipient_count=0
client_address=1.2.3.4
```

Documentation de Postfix en français

```
client_name=un.autre.domaine.tld
reverse_client_name=un.autre.domaine.tld
instance=123.456.7
Postfix versions 2.2 et supérieures :
sasl_method=plain
sasl_username=vous
sasl_sender=
size=12345
ccert_subject=solaris9.porcupine.org
ccert_issuer=Wietse Venema
ccert_fingerprint=C2:9D:F4:87:71:73:73:D9:18:E7:C2:F3:C1:DA:6E:04
Postfix versions 2.3 et supérieures :
encryption_protocol=TLSv1/SSLv3
encryption_cipher=DHE-RSA-AES256-SHA
encryption_keysize=256
[ligne vide]
```

Notes :

- L'attribut "request" est obligatoire. Dans cet exemple, le type de requête est "smtpd_access_policy".
- L'ordre des attributs est sans importance. Le serveur de politique devrait ignorer celles qu'il ne comprend pas.
- Lorsqu'un même attribut est transmis plusieurs fois, le serveur ne devrait garder que la première ou la dernière valeur.
- Lorsqu'un attribut n'est pas disponible, le client ne l'envoie pas ou l'envoie avec une valeur vide ("name="), ou l'envoie avec une valeur 0 ("name=0") pour les attributs numériques.
- L'attribut "recipient" n'est disponible qu'à l'étape "RCPT TO", et aux étapes "DATA" et "END-OF-MESSAGE" lorsque Postfix n'accepte qu'un seul destinataire au message.
- L'attribut "recipient_count" (Postfix 2.3 et supérieurs) est non nul seulement aux étapes "DATA" et "END-OF-MESSAGE". Il indique le nombre de destinataires que Postfix a accepté pour le message en cours.
- L'adresse du client se présente sous la forme pointée IPv4 (1.2.3.4) ou sous la forme d'une adresse IPv6 (1:2:3::4:5:6).
- Pour les explications sur les différences entre les informations client_name inverse et vérifiée, reportez-vous au paragraphe reject_unknown_client_hostname de la page postconf(5).
- Un nom d'attribut ne doit pas contenir les signes "=", null ou "nouvelle ligne" et la valeur associée ne doit pas contenir le signe null ou "nouvelle ligne".
- La valeur de l'attribut "instance" peut être utilisée pour corréler différentes requêtes concernant la même livraison de message.
- La valeur de l'attribut "size" indique la taille du message que le client a indiqué dans la commande MAIL FROM command (zéro si rien n'est indiqué). Avec Postfix 2.2 et supérieurs, il indique la taille réelle du message lorsque le client envoie la commande END-OF-DATA.
- Les attributs "sasl_*" (Postfix 2.2 et supérieurs) présentent les informations sur la façon dont le client a été authentifié via SASL. Ces attributs sont vides s'il n'y a pas d'authentification SASL.
- Les attributs "ccert_*" (Postfix 2.2 et supérieurs) présentent les informations sur la façon dont le client a été authentifié via TLS. Ces attributs sont vides s'il n'y a pas d'authentification TLS.
- Les attributs "encryption_*" (Postfix 2.3 et supérieurs) présentent les informations sur la façon dont la connexion est chiffrée. Dans le cas des connexions en clair, les attributs concernant le protocole et le chiffrement sont vides et la taille de clef est fixée à 0.

Ce qui suit est spécifique aux requêtes de politique déléguées SMTPD

- Les noms de protocole (protocole_name) sont ESMTP ou SMTP.

- Les états du protocole (`protocol_state`) sont CONNECT, EHLO, HELO, MAIL, RCPT, DATA, VRFY ou ETRN; ce sont les différents états dans lesquels le serveur SMTP de Postfix prend des décisions OK/REJECT/HOLD/etc.

Le serveur de politique répond par une action autorisée dans une table [access\(5\)](#) du serveur SMTPD de Postfix. Exemple :

```
action=defer if permit Service temporarily unavailable  
[ligne vide]
```

Ceci entraîne le rejet de la requête avec un code d'erreur 450 et la mention "Service temporarily unavailable", si le serveur SMTP ne trouve aucune raison de rejeter la requête à titre permanent.

En cas de problème, le serveur ne doit pas renvoyer de réponse mais logger un avertissement et se déconnecter. Postfix retentera la requête ultérieurement.<

Configuration de la politique client/serveur

Le client de délégation de la politique de Postfix peut se connecter à une socket TCP ou UNIX. Exemples :

```
inet:127.0.0.1:9998  
unix:/chemin/absolu  
unix:chemin/relatif
```

Le premier exemple indique que le serveur écoute sur le port TCP 9998 de la boucle locale, le second sur une socket UNIX et le troisième sur une socket UNIX située dans le répertoire des files d'attente de Postfix (`/var/spool/postfix` en général). Utilisez ce dernier pour les serveurs de politiques contrôlés par le démon master de Postfix.

Pour créer un service de politique qui écoute une socket UNIX appelée "policy" et fonctionne sous le contrôle du démon [spawn\(8\)](#) de Postfix, vous pouvez configurer Postfix comme suit :

```
1 /etc/postfix/master.cf:  
2   policy unix -      n      n      -      -      spawn  
3     user=nobody argv=/chemin/vers/le/serveur/de/politique  
4  
5 /etc/postfix/main.cf:  
6   smtpd_recipient_restrictions =  
7     ...  
8     reject_unauth_destination  
9     check_policy_service unix:private/policy  
10    ...  
11   policy_time_limit = 3600
```

NOTES :

- Lignes 2 et 11 : le démon [spawn\(8\)](#) tue par défaut les processus après 1000 secondes. C'est trop court pour un démon qui doit fonctionner aussi longtemps que le client SMTP est connecté au serveur SMTP. Cette valeur par défaut est surchargée dans le fichier [main.cf](#) avec le paramètre "policy_time_limit". Le nom du paramètre est le nom de l'entrée du fichier [master.cf](#) ("policy") avec le suffixe "_time_limit".
- Lignes 8 et 9 : indiquez toujours "[check_policy_service](#)" APRÈS "[reject_unauth_destination](#)" sinon votre système peut devenir un relais ouvert.

- Les sockets UNIX Solaris ne sont pas fiables. Utilisez plutôt des sockets TCP :

```
1 /etc/postfix/master.cf:
2 127.0.0.1:9998 inet n n n - - spawn
3 user=nobody argv=/some/where/policy-server
4
5 /etc/postfix/main.cf:
6 smtpd_recipient_restrictions =
7 ...
8 reject_unauth_destination
9 check_policy_service inet:127.0.0.1:9998
10 ...
11 127.0.0.1:9998_time_limit = 3600
```

Les autres paramètres de configuration qui contrôlent le côté client du protocole de délégation de politique sont :

- smtpd_policy_service_max_idle (défaut: 300s) : le temps au bout duquel le serveur SMTP ferme une connexion inutilisée sur le serveur délégataire.
- smtpd_policy_service_max_ttl (défaut: 1000s) : le temps minimum avant que le serveur SMTP de Postfix ne ferme une connexion active au serveur délégataire.
- smtpd_policy_service_timeout (défaut: 100s) : le temps limite pour se connecter, envoyer ou recevoir d'un serveur délégataire.

Exemple: serveur de politique liste grise

Les listes grises sont une défense contre les courriers publicitaires non sollicités décrite à l'adresse <http://www.greylisting.org/>. Cette idée a été discutée sur la liste de diffusion postfix-users un an avant qu'elle ne soit popularisée.

Le fichier `exemples/smtpd-policy/greylist.pl` des sources de Postfix implémente un serveur simple de politique de liste grise. Il enregistre une marque de temps pour chaque tuple (client, expéditeur, destinataire). Par défaut, le message n'est accepté que si la marque de temps a plus de 60 secondes. Ceci arrête le pourriel venant d'adresses aléatoires et les messages envoyés et celui envoyé par un proxy ouvert sélectionné aléatoirement, ainsi que celui envoyé par des spammers qui changent leur adresse fréquemment.

Copiez `exemples/smtpd-policy/greylist.pl` dans le répertoire `/usr/libexec/postfix` ou un autre emplacement plus approprié à votre système.

Vous devez indiquer la l'emplacement du fichier de base de données de la liste grise dans le fichier `greylist.pl`, et combien de temps les messages doivent être retardés avant d'être acceptés. Les valeurs par défaut sont :

```
$database_name="/var/mta/greylist.db";
$greylist_delay=60;
```

Le répertoire `/var/mta` (ou celui que vous choisirez) doivent avoir les droits en écriture pour "nobody" ou pour l'utilisateur utilisé dans le fichier master.cf pour le service de politique.

Exemple :

```
# mkdir /var/mta
# chown nobody /var/mta
```

Note: ne créez PAS la base de données de la liste grise dans un répertoire positionné en écriture pour tout le monde tell /tmp ou /var/tmp et ne la créez PAS dans un système de fichiers susceptible d'être saturé. Postfix peut survivre dans des conditions de saturation de file d'attente ou de stockage des boîtes-aux-lettres mais pas avec une base corrompue. Dans ce dernier cas vous ne recevrez du courrier que lorsque vous aurez effacé ce fichier à la main.

Le script Perl `greylist.pl` peut être lancé sous le contrôle du démon `master.cf`. Par exemple pour le lancer sous le compte "nobody" en utilisant une socket UNIX accessible à Postfix, inscrivez simplement :

```
1 /etc/postfix/master.cf:
2   policy unix - n n - - spawn
3     user=nobody argv=/usr/bin/perl /usr/libexec/postfix/greylist.pl
4
5 /etc/postfix/main.cf:
6   policy_time_limit = 3600
```

Notes :

- Ligne 3 : utilisez "greylist.pl -v" pour avoir des logs plus bavards pour chaque requête ou réponse.
- Lignes 2 et 6 : le démon `spawn(8)` de Postfix tue par défaut tous les processus après 1000 secondes. C'est trop court pour un démon de politique qui doit fonctionner aussi longtemps qu'un client est connecté à un processus du serveur SMTP. Cette limite peut être outrepassée en renseignant le paramètre "policy_time_limit" du fichier `main.cf`. Le nom du paramètre est le nom de l'entrée dans le fichier `master.cf` ("policy") accolée au suffixe "_time_limit".

Sur Solaris vous devez utiliser des sockets TCP au lieu des sockets UNIX comme indiqué au paragraphe "[Policy client/server configuration](#)" ci-dessus.

```
1 /etc/postfix/master.cf:
2   127.0.0.1:9998 inet n n n - - spawn
3     user=nobody argv=/usr/bin/perl /usr/libexec/postfix/greylist.pl
4
5 /etc/postfix/main.cf:
6   127.0.0.1:9998_time_limit = 3600
```

Pour appeler ce service, vous devez indiquer "`check_policy_service inet:127.0.0.1:9998`".

Mettre en liste grise le courrier des domaines fréquemment forgés

Il est relativement sécurisé d'activer la mise en liste grise pour des domaines apparaissant souvent dans les messages forgés. Une liste des domaines fréquemment utilisés dans les commandes MAIL FROM est disponible à l'adresse suivante : <http://www.monkeys.com/anti-spam/filtering/sender-domain-validate.in>.

```
1 /etc/postfix/main.cf:
2   smtpd_recipient_restrictions =
3     reject_unlisted_recipient
4     ...
5     reject_unauth_destination
6     check_sender_access hash:/etc/postfix/sender_access
7     ...
8   restriction_classes = greylist
9   greylist = check_policy_service unix:private/policy
10
```

```
11 /etc/postfix/sender_access:
12     aol.com      greylist
13     hotmail.com  greylist
14     bigfoot.com  greylist
15     ... etc ...
```

NOTES :

- Ligne 9 : Sur Solaris vous devez utiliser des sockets réseau au lieu des sockets UNIX comme indiqué au paragraphe "Exemple: serveur de politique liste grise" ci-avant.
- Ligne 6 : Assurez-vous d'insérer "check_sender_access" APRÈS "reject_unauth_destination" sinon votre système risque de devenir un relais ouvert.
- Ligne 3 : Sur les version 2.0 et antérieures de Postfix, "reject_unlisted_recipient" est appelé "check_recipient_maps". Postfix 2.1 accepte les deux syntaxes.
- Ligne 3 : La base de données de la liste grise est rapidement polluée avec les adresses invalides. Protégez la liste grise avec d'autres restrictions qui rejettent les destinataires et expéditeurs inconnus.

Mettre en liste grise tout le courrier

Si vous activez la liste grise pour tout le courrier, vous souhaitez probablement faire des exceptions pour les listes de distributions qui utilisent une adresse d'expédition changeante car de telles listes peuvent polluer relativement rapidement la base de données de la liste grise.

```
1 /etc/postfix/main.cf:
2     smtpd_recipient_restrictions =
3         reject_unlisted_recipient
4         ...
5         reject_unauth_destination
6         check_sender_access hash:/etc/postfix/sender_access
7         check_policy_service unix:private/policy
8         ...
9
10 /etc/postfix/sender_access:
11     securityfocus.com OK
12     ...
```

NOTES :

- Ligne 7 : Sur Solaris vous devez utiliser des sockets TCP et non UNIX comme indiqué au paragraphe "Exemple: serveur de politique liste grise" ci-avant.
- Lignes 6 et 7 : Assurez vous d'insérer check_sender_access et check_policy_service APRÈS reject_unauth_destination sinon votre système risque de devenir un relais ouvert.
- Ligne 3 : La base de données de la liste grise est rapidement polluée avec les adresses invalides. Protégez la liste grise avec d'autres restrictions qui rejettent les destinataires et expéditeurs inconnus.

Routine de maintenance des listes grises

La base de données de la liste grise s'accroît au fil du temps car le serveur de politique ne retire jamais d'entrées. Sans surveillance, elle risque de saturer votre système.

Lorsque ce fichier dépasse un certain seuil, vous pouvez simplement le renommer ou le détruire sans effet nuisible; Postfix le recréera automatiquement. Dans le pire des cas, les nouveaux messages seront retardés d'une heure. Pour minimiser cet impact, effectuez cette opération au milieu de la nuit ou le week-end.

Exemple de serveur Perl de liste grise

Ci-dessous une sous-routine Perl qui implémente la politique liste grise citée en exemple. C'est une partie de l'exemple inclus dans les sources de Postfix (exemples/smtpd-policy/greylist.pl).

```
#
# greylist status database and greylist time interval. DO NOT create the
# greylist status database in a world-writable directory such as /tmp
# or /var/tmp. DO NOT create the greylist database in a file system
# that can run out of space.
#
$database_name="/var/mta/greylist.db";
$greylist_delay=60;

#
# Demo SMTPD access policy routine. The result is an action just like
# it would be specified on the right-hand side of a Postfix access
# table. Request attributes are available via the %attr hash.
#
sub smtpd_access_policy {
    my($key, $time_stamp, $now);

    # Open the database on the fly.
    open_database() unless $database_obj;

    # Lookup the time stamp for this client/sender/recipient.
    $key =
        lc $attr{"client_address"}."/".$attr{"sender"}."/".$attr{"recipient"};
    $time_stamp = read_database($key);
    $now = time();

    # If new request, add this client/sender/recipient to the database.
    if ($time_stamp == 0) {
        $time_stamp = $now;
        update_database($key, $time_stamp);
    }

    # The result can be any action that is allowed in a Postfix access(5) map.
    #
    # To label the mail, return ``PREPEND headername: headertext''
    #
    # In case of success, return ``DUNNO'' instead of ``OK'', so that the
    # check_policy_service restriction can be followed by other restrictions.
    #
    # In case of failure, return ``DEFER_IF_PERMIT optional text...',
    # so that mail can still be blocked by other access restrictions.
    #
    syslog $syslog_priority, "request age %d", $now - $time_stamp if $verbose;
    if ($now - $time_stamp > $greylist_delay) {
        return "dunno";
    } else {
        return "defer if permit Service temporarily unavailable";
    }
}
```

Vérification des adresses par

Postfix

ATTENTION ATTENTION ATTENTION

La fonctionnalité de vérification des adresses de destination/d'expédition décrite dans ce document n'est utilisable que sur des sites ayant une charge modérée. Elle décroît sensiblement les performances sous une charge élevée et risque de faire mettre votre site en liste noire de certains fournisseurs. Reportez-vous au paragraphe "Limites" ci-après pour plus de détails.

Quelles vérifications d'adresse Postfix peut-il faire pour vous

La vérification d'adresse est une fonctionnalité qui permet au serveur SMTP de Postfix de bloquer une adresse d'expédition (MAIL FROM) ou de destination (RCPT TO) jusqu'à ce que l'adresse ait été vérifiée comme livrable.

Cette technique a d'évidents intérêts pour rejeter le courrier non désirable venant d'une adresse inexistante.

Elle peut également être pratique pour bloquer le courrier des destinataires inexistants, par exemple sur une machine relais qui n'a pas de liste de destinataires valides. Ceci évite de faire entrer en file d'attente des messages non livrables évitant ainsi le transport de messages MAILER-DAEMON en retour.

Cette fonctionnalité est disponible sur les versions 2.1 et supérieures de Postfix.

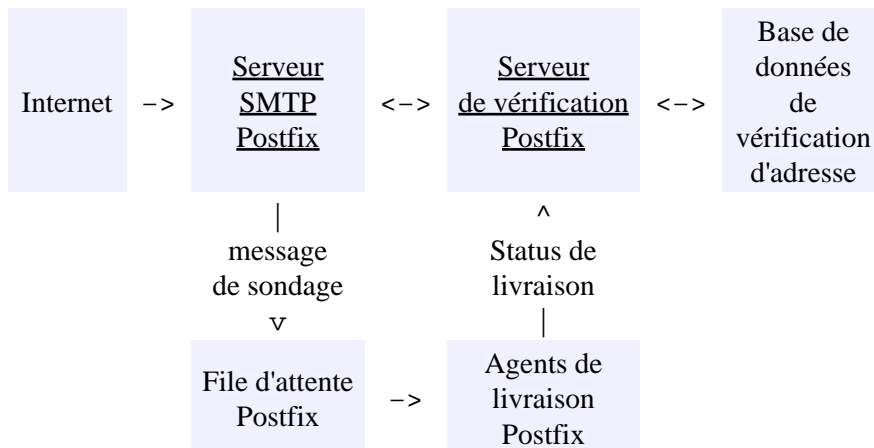
Sujets abordés dans ce document :

- Comment marche la vérification d'adresse
- Limites de la vérification d'adresse
- Vérification des adresses de destination
- Vérification des adresses d'expédition pour les domaines fréquemment forgés
- Vérification des adresses d'expédition pour tous le courrier
- Base de données de vérification d'adresse
- Gestion des base de données de vérification d'adresse
- Contrôle du routage des sondages de vérification d'adresse
- Exemples de routage forcé du sondage
- Limites du routage forcé du sondage

Comment marche la vérification d'adresse

Une adresse d'expédition ou de destination est vérifiée en sondant le MTA (agent de transfert de messages) suivant sans délivrer immédiatement le message. Celui-ci peut être Postfix lui-même ou être un MTA distant (SMTP interruptus). Les messages de sondages sont des messages normaux si ce n'est qu'ils ne sont jamais

livrés, retardés ou renvoyés; ils sont toujours jetés.



Avec la vérification d'adresse de Postfix activée, le courrier normal souffrira seulement d'un court délai de moins de 6 secondes lorsqu'une adresse est vérifiée pour la première fois. Une fois ce status connu, il est caché et Postfix répond immédiatement.

Lorsque la vérification prend trop de temps, le serveur SMTP retarde l'adresse d'expédition ou de destination avec une réponse 450. Les clients de messagerie normaux se reconnecteront après un certain délai. La vérification d'adresse est configurable avec les paramètres `address_verify_poll_count` et `address_verify_poll_delay` du fichier `main.cf`. Reportez-vous à la page [postconf\(5\)](#) pour plus de details.

Limites de la vérification d'adresse

- Pour vérifier une adresse distante, Postfix sonde le MTA suivant avec cette adresse, sans délivrer de message. Si le MTA suivant accepte l'adresse, alors Postfix suppose que l'adresse est livrable. En réalité, un message peut être rejeté APRÈS que le MTA ait accepté l'adresse de destination.
- Certains sites risquent de vous mettre en liste noire lorsque vous l'aurez sondé trop souvent (un sondage est une session SMTP qui ne livre aucun message) ou lorsque vous l'aurez sondé trop souvent avec des adresses inexistantes. C'est une des raisons pour laquelle vous ne devrez utiliser la vérification d'adresse avec parcimonie et si, bien sûr, votre site reçoit beaucoup de courrier.
- Normalement, les messages de sondage de vérification d'adresse suivent le même chemin que les messages réguliers. Toutefois, certains site envoient le courrier vers Internet au travers d'un relais, ce qui casse la vérification. Reportez-vous au paragraphe "[Contrôle du routage des sondages de vérification d'adresse](#)", pour voir comment surcharger le routage du message et pour les limites de cette approche.
- Postfix suppose qu'une adresse n'est pas livrable lorsque le MTA suivant rejette le sondage sans regarder la raison de ce rejet (client rejeté, HELO rejeté, MAIL FROM rejeté, etc.). Ainsi Postfix rejette le message lorsque le MTA de l'expéditeur rejette le courrier de votre machine. C'est une bonne chose.
- Malheureusement, certains sites majeurs tels YAHOO ne rejettent pas les adresses en réponse à la commande RCPT TO, mais reportent cette réponse à la fin de la commande DATA après que le message ait été transféré. La vérification ne fonctionnera pas avec ces sites.
- Par défaut, les messages de sonde de Postfix utilisent l'adresse "`postmaster@$myorigin`" comme adresse d'expédition. C'est plus sûr car Postfix ne rejette pas le courrier de ces adresses.

Vous pouvez la changer en adresse nulle ("`address_verify_sender =`"). Ce n'est PAS sûr car la vérification échouera avec les sites mal configurés qui rejettent MAIL FROM: <>, alors que les

sondages de "postmaster@\$myorigin" auraient réussi.

Vérification des adresses de destination

Comme indiqué ci-avant, la vérification des adresses de destination peut être intéressante pour bloquer le courrier des destinataires non joignables sur un serveur relais qui n'a pas une liste des adresses de destination valides. Ceci peut éviter d'encombrer les files d'attente avec les messages MAILER-DAEMON.

Cette vérification est relativement sûre et sans surprises. Si un sondage de destinataire échoue, Postfix rejette le courrier à destination de cette adresse. S'il réussit, Postfix accepte le courrier à destination de cette adresse.

```
/etc/postfix/main.cf:
  smtpd_recipient_restrictions =
    permit_mynetworks
    reject_unauth_destination
    ...
    reject_unknown_recipient_domain
    reject_unverified_recipient
    ...
```

La restriction "reject_unknown_recipient_domain" bloque le courrier des domaines inexistants. La mettre avant "reject_unverified_recipient" évite la surcharge du sondage.

Le paramètre unverified_recipient_reject_code (défaut 450) indique comment Postfix doit répondre lorsqu'une adresse est connue pour rebondir. Changez cette valeur en 550 lorsque vous faites confiance au jugement de Postfix.

Vérification des adresses d'expédition pour les messages de domaines fréquemment forgés

Il est relativement sûr d'activer la vérification des adresses d'expédition pour certains domaines qui apparaissent souvent dans les messages forgés.

```
/etc/postfix/main.cf:
  smtpd_sender_restrictions = hash:/etc/postfix/sender_access
  unverified_sender_reject_code = 550
  # Note 1: Prenez connaissance du paragraphe "Cache" ci-dessous!
  # Note 2: Evitez les fichiers hash ici. Utilisez plutôt btree.
  address_verify_map = btree:/var/mta/verify

/etc/postfix/sender_access:
  aol.com      reject_unverified_sender
  hotmail.com reject_unverified_sender
  bigfoot.com reject_unverified_sender
  ... etc ...
```

Une liste des domaines fréquemment utilisés dans les MAIL FROM des messages forgés peut être trouvée à l'adresse <http://www.monkeys.com/anti-spam/filtering/sender-domain-validate.in>.

NOTE: Une des premières choses à faire est d'activer la vérification d'adresse pour tous vos domaines.

Vérification des adresses d'expédition pour tous le courrier

Malheureusement, la vérification des adresses d'expédition ne peut être activée simplement pour tout le courrier – vous risquez de perdre le courrier légitime venant de systèmes mal configurés. Vous devrez sans doute utiliser des listes blanches pour des adresses spécifiques ou même des domaines entiers.

Pour découvrir comment la vérification des adresses d'expédition va se comporter, utilisez "warn if reject reject unverified sender", vous verrez ainsi quels messages auraient été bloqué :

```
/etc/postfix/main.cf:
    smtpd_sender_restrictions =
        permit mynetworks
        ...
        check_sender_access hash:/etc/postfix/sender_access
        reject_unknown_sender_domain
        warn if reject reject unverified sender
        ...
# Note 1: Prenez connaissance du paragraphe "Cache" ci-dessous!
# Note 2: Evitez les fichiers hash ici. Utilisez plutôt btree.
address_verify_map = btree:/var/mta/verify
```

C'est également une bonne idée pour remplir votre cache avec les résultats des vérifications avant de réellement rejeter le courrier.

La restriction sender_access est obligatoire pour les listes blanches d'adresses ou de domaines connus pour être fiables. Ainsi, Postfix ne marquera pas une adresse connue pour être fiable comme mauvaise après un sondage raté, il vaut mieux être sûr que désolé.

NOTE: Vous devrez inscrire en liste blanche les sites tels securityfocus.com qui gèrent des listes de diffusion utilisant différentes adresses à chaque envoi (VERP). De telles adresses polluent rapidement le cache de vérification des adresses et génèrent d'inutiles sondage.

```
/etc/postfix/sender_access
securityfocus.com OK
...
```

La restriction "reject_unknown_sender_domain" bloque le courrier des domaines inexistants. L'inscrire avant "reject_unverified_sender" évite une surcharge de génération de messages de sondage inutile.

Le paramètre unverified_sender_reject_code (défaut 450) indique comment Postfix doit répondre lorsqu'une adresse est connue pour rebondir. Changez ce paramètre en 550 lorsque vous faites confiance au jugement de Postfix.

Base de données de vérification d'adresse

NOTE : Par défaut, les informations de vérification d'adresse n'est pas stockée sur un fichier persistant. Vous devez en indiquer un dans le fichier main.cf (voir ci-dessous). Le stockage persistant est désactivé par défaut car il risque d'utiliser beaucoup d'espace disque.

Les informations de vérification d'adresse est cachée par le démon de vérification de Postfix. Postfix a différents paramètres pour contrôler le cache des réponses positives ou négatives. Reportez-vous à la page de manuel verify(8) pour plus de détails.

Le paramètre de configuration address_verify_map (NOTE : singulier) indique un fichier optionnel pour utiliser une base de données persistante des résultats des vérifications d'adresse d'expédition. Si vous ne renseignez pas ce paramètre, toutes les informations de vérification sont perdues après un "postfix reload" or "postfix stop".

Si votre système de fichiers /var dispose d'assez d'espace, essayez :

```
/etc/postfix/main.cf:
# Note: évitez les fichiers hash ici, utilisez plutôt btree.
address_verify_map = btree:/var/mta/verify
```

NOTE : N'installez pas cette base dans un système de fichier qui risque d'être saturé. Lorsque la table de vérification des adresses est corrompue, plus aucun message ne peut être reçu et vous devez MANUELLEMENT effectuer les réparations décrites au paragraphe suivant.

Le processus démon verify(8) créera la nouvelle base si elle n'existe pas et l'ouvrira avant d'entrer dans la cage chroot et avant de perdre ses privilèges root

Gestion des base de données de vérification d'adresse

La page de manuel verify(8) décrit les paramètres qui contrôlent combien de temps les informations restent en cache avant d'être rafraichies et combien de temps elles peuvent rester en cache sans être rafraichies avant d'expirer. Postfix utilise différents contrôles pour les résultats positifs (adresse acceptée) et négatifs (adresse rejetée).

Actuellement, aucun outil n'est fourni pour gérer la base des vérifications d'adresse. Si le fichier devient trop gros ou corrompu, vous devez l'effacer manuellement ou le renommer puis lancer "postfix reload". Le nouveau démon verify créera la nouvelle base de données.

Contrôle du routage des sondages de vérification d'adresse

Par défaut, Postfix sends address verification probe messages via the same route as regular mail, because that normally produces the most accurate result. It's no good to verify a local address by connecting to your own SMTP port; that just triggers all kinds of mailer loop alarms. The same is true for any destination that your machine is best MX host for: hidden domains, virtual domains, etc.

Toutefois, certains sites ont une infrastructure complexe où le courrier n'est pas directement envoyé sur Internet, mais est transmis à un relayhost intermédiaire. C'est un problème pour les vérifications d'adresses car elles ne peuvent être effectuées que si Postfix se connecte directement sur l'hôte de destination.

Pour cette raison, Postfix vous permet de redéfinir les paramètres de routage lorsqu'il livre un message de sondage.

En premier lieu, le paramètre address_verify_relayhost vous permet de surcharger le paramètre relayhost et le paramètre address_verify_transport_maps le paramètre transport_maps. Le paramètre address_verify_sender_dependent_relayhost_maps effectue la même chose pour la sélection du relayhost en fonction du destinataire.

De plus, chaque classe d'adresses peut avoir sa propre valeur pour le transport comme indiqué dans le tableau ci-dessous. Les classes d'adresses sont définies dans la page ADDRESS CLASS README.

liste de domaine	Transport régulier	Transport de vérification
<u>mydestination</u>	<u>local transport</u>	<u>address verify local transport</u>
<u>virtual alias domains</u>	(pas applicable)	(pas applicable)
<u>virtual mailbox domains</u>	<u>virtual transport</u>	<u>address verify virtual transport</u>
<u>relay domains</u>	<u>relay transport</u>	<u>address verify relay transport</u>
(pas applicable)	<u>default transport</u>	<u>address verify default transport</u>

Par défaut, les paramètres qui contrôlent la livraison des messages de sondage ont la même valeur que ceux contrôlant la livraison normale.

Exemples de routage forcé du sondage

Scénario typique : surcharge du paramètre relayhost pour les sondages de vérification d'adresse, le reste étant inchangé.

```
/etc/postfix/main.cf:
    relayhost = $mydomain
    address verify relayhost =
    ...
```

Les sites derrière un traducteur d'adresse réseau doivent utiliser un client SMTP différent qui envoie des informations de nom d'hôte correctes :

```
/etc/postfix/main.cf:
    relayhost = $mydomain
    address verify relayhost =
    address verify default transport = direct_smtp

/etc/postfix/master.cf:
    direct_smtp .. .. . smtp
    -o smtp helo name=nat.box.tld
```

Limites du routage forcé du sondage

Des incohérences peuvent arriver lorsque les messages ne suivent pas le même chemin que les messages normaux. Par exemple, un message peut être accepté lorsqu'il suit un chemin normal et être rejeté par la route forcée. L'inverse peut être vrai, mais plus rarement.

Contrôle d'accès par client,

utilisateur, etc.

Classes de restriction de Postfix

Le serveur SMTP de Postfix supporte différentes restrictions d'accès parmi les quels reject_rbl_client ou reject_unknown_client sur la partie droite des tables du serveur SMTP (access(5)). Ceci vous permet d'implémenter différentes restrictions contre les spams.

Devoir indiquer des listes de restriction d'accès pour chaque destinataire peut vite devenir fastidieux. Les classes de restrictions vous permettent de donner des noms faciles à retenir aux groupes de restrictions anti-spam (tel "permissif", "restrictif", ...).

La réelle raison de l'existence des classes de restrictions Postfix est plus pratique : vous ne pouvez pas indiquer une autre table de correspondance dans la partie droite des tables d'accès. C'est parce que Postfix doit ouvrir les tables au démarrage, mais le lecteur n'est probablement pas intéressé par ces détails.

Exemple :

```
/etc/postfix/main.cf :
    smtpd_restriction_classes = restrictive, permissive
    restrictive = reject_unknown_sender_domain reject_unknown_client ...
    permissive = permit

    smtpd_recipient_restrictions =
    permit_mynetworks
    reject_unauth_destination
        hash:/etc/postfix/recipient_access

/etc/postfix/recipient_access:
joe@my.domain      permissive
jane@my.domain     restrictive
```

Dans cet exemple, vous pouvez utiliser "restrictive" ou "permissive" sur la partie droite de vos table d'accès par client/helo/destinataire/expéditeur su serveur SMTPD.

La suite de ce document montre des exemples d'emploi des classes de restriction de Postfix :

- Interdire une liste de diffusion interne aux expéditeurs extérieurs,
- Prévenir les accès extérieurs par des utilisateurs internes.

Ces questions reviennent fréquemment et les exemples montrent clairement que ces classes de restriction ne sont pas réellement la bonne solution. Elles peuvent être utilisées pour ce pour quoi elles ont été conçues : différentes restriction anti-spam pour différents clients ou utilisateurs.

Protéger les listes de distribution internes

Nous voulons implémenter une liste de distribution interne `all@mon.domaine` qui contient tous les employés. Ma première idée fut d'utiliser les alias, mais cela n'interdit pas aux extérieurs de l'utiliser.

Postfix peut implémenter un contrôle d'accès par adresse. Ce qui suit est basé sur l'adresse IP du client SMTP et donc est sujette à l'IP spoofing.

```
/etc/postfix/main.cf :
    smtpd_recipient_restrictions =
        hash:/etc/postfix/access
        ...comme d'habitude...

/etc/postfix/access:
    all@my.domain    permit_mynetworks,reject
    all@my.hostname  permit_mynetworks,reject
```

Utilisez **dbm** au lieu de **hash** si votre système utilise des fichiers **dbm** au lieu de fichiers **db**. Pour connaître les types de tables supportées par Postfix, utilisez la commande **postconf -m**.

Cela peut être suffisant si votre machine reçoit le courrier d'Internet directement, mais pas si votre réseau dépasse le niveau d'une simple agence. Par exemple, vos MX de sauvegarde blanchiront l'adresse IP du client qui semblera venir d'un réseau sûr.

Dans le cas général, vous avez besoin de deux tables de correspondances : une table listant les destinations à protéger et une autre pour les domaines autorisés à envoyer du courrier aux destinations protégées.

Ce qui suit est basé sur l'adresse de l'expéditeur de l'enveloppe SMTP et donc sujette au spoofing de cette adresse.

```
/etc/postfix/main.cf :
    smtpd_recipient_restrictions =
        hash:/etc/postfix/protected_destinations
        ...contenu habituel...

    smtpd_restriction_classes = insiders_only
    insiders_only = check_sender_access hash:/etc/postfix/insiders, reject

/etc/postfix/protected_destinations:
    all@my.domain    insiders_only
    all@my.hostname  insiders_only

/etc/postfix/insiders:
    mon.domain       OK    correspond à mon.domain et ses sous-domaines
    autre.domaine    OK    correspond à un autre.domaine et ses sous-domaines
```

Usurper ce schéma est relativement aisé car la seule chose à faire est de modifier l'adresse de l'expéditeur SMTP.

Si la liste interne est de petite taille, peut-être est-il mieux de la modérer.

Restreindre les utilisateurs pouvant envoyer du courrier vers sites extérieurs

Comment puis-je configurer Postfix afin de restreindre la liste des utilisateurs pouvant envoyer du courrier vers les sites extérieurs ? Les utilisateurs n'ayant pas accès à Internet devant recevoir un message générique de rejet. Il ne s'agit pas ici de discuter de l'opportunité d'une telle décision, mais d'en étudier l'aspect technique.

Postfix dispose d'un support des restrictions par utilisateurs. Ces restrictions sont implémentées dans le serveur SMTP. Les utilisateurs violant cette politique verront leurs messages rejetés par le serveur SMTP comme suit :

```
554 <utilisateur@site.distant>: Access denied
```

Cette implémentation utilise deux tables de correspondances. L'une définit les utilisateurs qui sont restreints à l'envoi de messages locaux et l'autre les destinations considérées comme locales. Il s'agit d'un exercice que le lecteur pourra adapter s'il doit restreindre la majorité de ses utilisateurs en utilisant une table des utilisateurs autorisés.

Cet exemple utilise des fichiers DB/DBM, mais peut être adapté avec LDAP ou SQL.

```
/etc/postfix/main.cf :
    smtpd_recipient_restrictions =
        check_sender_access hash:/etc/postfix/restricted_senders
        ...suite...

    smtpd_restriction_classes = local_only
    local_only =
        check_recipient_access hash:/etc/postfix/local_domains, reject

/etc/postfix/restricted_senders:
    foo@domain      local_only
    bar@domain      local_only

/etc/postfix/local_domains:
    this.domain     OK      correspond à this.domain et ses sous-domaines
    that.domain     OK      correspond à that.domain et ses sous-domaines
```

Indiquez **dbm** au lieu de **hash** si votre système utilise des fichiers **dbm** au lieu de fichiers **db**. Pour connaître les types de bases supportées par Postfix, utilisez la commande **postconf -m**.

Note : ce schéma n'authentifie pas les utilisateurs et peut être contourné par plusieurs voies :

- En envoyant un message via un relais moins restrictif.
- En envoyant un message à quelqu'un d'autre qui a la permission d'envoyer ce message.

Postfix ETRN Howto

Purpose of the Postfix fast ETRN service

The SMTP ETRN command was designed for sites that have intermittent Internet connectivity. With ETRN, a site can tell the mail server of its provider to "Please deliver all my mail now". The SMTP server searches the queue for mail to the customer, and delivers that mail **by connecting to the customer's SMTP server**. The mail is not delivered via the connection that was used for sending ETRN.

Postfix versions before 1.0 (also known as version 20010228) implemented the ETRN command in an inefficient manner: they simply attempted to deliver all queued mail. This is slow on mail servers that queue mail for many customers.

As of version 1.0, Postfix has a fast ETRN implementation that does not require Postfix to examine every queue file. Instead, Postfix maintains a record of what queue files contain mail for destinations that are configured for ETRN service. ETRN service is no longer available for domains that aren't configured for the service.

This document provides information on the following topics:

- [Using the Postfix fast ETRN service](#)
- [How Postfix fast ETRN works](#)
- [Postfix fast ETRN service limitations](#)
- [Configuring the Postfix fast ETRN service](#)
- [Configuring a domain for ETRN service only](#)
- [Testing the Postfix fast ETRN service](#)

Other documents with information on this subject:

- [flush\(8\)](#), flush service implementation

Using the Postfix fast ETRN service

The following is an example SMTP session that shows how an SMTP client requests the ETRN service. Client commands are shown in bold font.

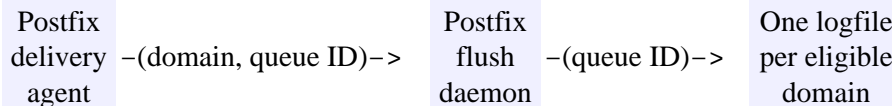
```
220 my.server.tld ESMTP Postfix
helo my.client.tld
250 Ok
etrn some.customer.domain
250 Queuing started
quit
221 Bye
```

As mentioned in the introduction, the mail is delivered by connecting to the customer's SMTP server; it is not sent over the connection that was used to send the ETRN command.

The Postfix operator can request delivery for a specific customer by using the command "sendmail -qR*destination*" and, with Postfix version 1.1 and later, "postqueue -s*destination*".

How Postfix fast ETRN works

When a Postfix delivery agent decides that mail must be delivered later, it sends the destination domain name and the queue file name to the flush(8) daemon which maintains per-destination logfiles with file names of queued mail. These logfiles are kept below `$queue_directory/flush`. Per-destination logfiles are maintained only for destinations that are listed with the `$fast_flush_domains` parameter and that have syntactically valid domain names.



When Postfix receives a request to "deliver mail for a domain now", the flush(8) daemon moves all deferred queue files that are listed for that domain to the incoming queue, and requests that the queue manager deliver them. In order to force delivery, the queue manager temporarily ignores the lists of undeliverable destinations: the volatile in-memory list of dead domains, and the list of message delivery transports specified with the defer transports configuration parameter.

Postfix fast ETRN service limitations

The design of the flush(8) server and of the flush queue introduce a few limitations that should not be an issue unless you want to turn on fast ETRN service for every possible destination.

- The flush(8) daemon maintains per-destination logfiles with queue file names. When a request to "deliver mail now" arrives, Postfix will attempt to deliver all recipients in the queue files that have mail for the destination in question. This does not perform well when queue files have recipients in many different domains.
- The flush(8) daemon maintains per-destination logfiles only for destinations listed with `$fast_flush_domains`. With other destinations it not possible to trigger delivery with "sendmail -qR*destination*" or, with Postfix version 1.1 and later, "postqueue -s*destination*".
- Up to and including early versions of Postfix version 2.1, the "fast flush" service may not deliver some messages if the request to "deliver mail now" is received while a deferred queue scan is already in progress. The reason is that the queue manager does not ignore the volatile in-memory list of dead domains, and the list of message delivery transports specified with the defer transports configuration parameter.

Configuring the Postfix fast ETRN service

The behavior of the flush(8) daemon is controlled by parameters in the main.cf configuration file.

By default, Postfix "fast ETRN" service is available only for destinations that Postfix is willing to relay mail to:

```

/etc/postfix/main.cf:
    fast_flush_domains = $relay_domains
    smtpd_etrn_restrictions = permit_mynetworks, reject
  
```

Notes:

- The relay_domains parameter specifies what destinations Postfix will relay to. For destinations that are not eligible for the "fast ETRN" service, Postfix replies with an error message.
- The smtpd_etrn_restrictions parameter limits what clients may execute the ETRN command. By default, any client has permission.

To enable "fast ETRN" for some other destination, specify:

```
/etc/postfix/main.cf:
    fast_flush_domains = $relay_domains, some.other.domain
```

To disable "fast ETRN", so that Postfix rejects all ETRN requests and so that it maintains no per-destination logfiles, specify:

```
/etc/postfix/main.cf:
    fast_flush_domains =
```

Configuring a domain for ETRN service only

While an "ETRN" customer is off-line, Postfix will make spontaneous attempts to deliver mail to it. These attempts are separated in time by increasing time intervals, ranging from \$minimal_backoff_time to \$maximal_backoff_time, and should not be a problem unless a lot of mail is queued.

To prevent Postfix from making spontaneous delivery attempts you can configure Postfix to always defer mail for the "ETRN" customer. Mail is delivered only after the ETRN command or with "sendmail -q", with "sendmail -qRdomain", or with "postqueue -sdomain"(Postfix version 1.1 and later only),

In the example below we configure an "etrn-only" delivery transport which is simply a duplicate of the "smtp" and "relay" mail delivery transports. The only difference is that mail destined for this delivery transport is deferred as soon as it arrives.

```
1 /etc/postfix/master.cf:
2  # =====
3  # service type  private unpriv  chroot  wakeup  maxproc command
4  #               (yes)   (yes)   (yes)   (never) (100)
5  # =====
6  smtp          unix  -      -      n      -      -      smtp
7  relay         unix  -      -      n      -      -      smtp
8  etrn-only     unix  -      -      n      -      -      smtp
9
10 /etc/postfix/main.cf:
11  relay_domains = customer.tld ...other domains...
12  defer_transports = etrn-only
13  transport_maps = hash:/etc/postfix/transport
14
15 /etc/postfix/transport:
16  customer.tld      etrn-only:[mailhost.customer.tld]
```

Translation:

- Line 8: The "etrn-only" mail delivery service is a copy of the "smtp" and "relay" service.
- Line 11: Don't forget to authorize relaying for this customer, either via relay_domains or with the permit_mx_backup feature.

- Line 12: The "etrn-only" mail delivery service is configured so that spontaneous mail delivery is disabled.
- Lines 13–16: Mail for the customer is given to the "etrn-only" mail delivery service.
- Line 16: The "[mailhost.customer.tld]" turns off MX record lookups; you must specify this if your Postfix server is the primary MX host for the customer's domain.

Testing the Postfix fast ETRN service

By default, "fast ETRN" service is enabled for all domains that match \$relay_domains. If you run Postfix with "fast ETRN" service for the very first time, you need to run "sendmail -q" once in order to populate the per-site deferred mail logfiles. If you omit this step, no harm is done. The logfiles will eventually become populated as Postfix routinely attempts to deliver delayed mail, but that will take a couple hours. After the "sendmail -q" command has completed all delivery attempts (this can take a while), you're ready to test the "fast ETRN" service.

To test the "fast ETRN" service, telnet to the Postfix SMTP server from a client that is allowed to execute ETRN commands (by default, that's every client), and type the commands shown in boldface:

```
220 my.server.tld ESMTP Postfix
helo my.client.tld
250 Ok
etrn some.customer.domain
250 Queuing started
```

where "some.customer.domain" is the name of a domain that has a non-empty logfile somewhere under \$queue_directory/flush.

In the maillog file, you should immediately see a couple of logfile records, as evidence that the queue manager has opened queue files:

```
Oct  2 10:51:19 myhostname postfix/qmgr[51999]: 682E8440A4:
    from=<whatever>, size=12345, nrcpt=1 (queue active)
Oct  2 10:51:19 myhostname postfix/qmgr[51999]: 02249440B7:
    from=<whatever>, size=4711, nrcpt=1 (queue active)
```

What happens next depends on whether the destination is reachable. If it's not reachable, the mail queue IDs will be added back to the some.customer.domain logfile under \$queue_directory/flush.

Repeat the exercise with some other destination that your server is willing to relay to (any domain listed in \$relay_domains), but that has no mail queued. The text in bold face stands for the commands that you type:

```
220 my.server.tld ESMTP Postfix
helo my.client.tld
250 Ok
etrn some.other.customer.domain
250 Queuing started
```

This time, the "ETRN" command should trigger NO mail deliveries at all. If this triggers delivery of all mail, then you used the wrong domain name, or "fast ETRN" service is turned off.

Finally, repeat the exercise with a destination that your mail server is not willing to relay to. It does not matter if your server has mail queued for that destination.

Documentation de Postfix en français

```
220 my.server.tld ESMTP Postfix
helo my.client.tld
250 Ok
etrn not.a.customer.domain
459 <not.a.customer.domain>: service unavailable
```

In this case, Postfix should reject the request as shown above.

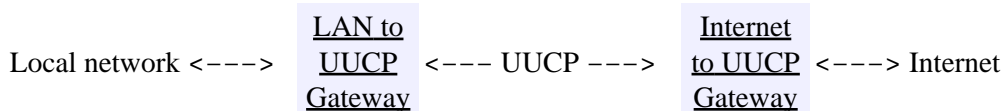
Postfix and UUCP

Using UUCP over TCP

Despite a serious lack of sex-appeal, email via UUCP over TCP is a practical option for sites without permanent Internet connections, and for sites without a fixed IP address. For first-hand information, see the following guides:

- Jim Seymour's guide for using UUCP over TCP at http://jimsun.LinxNet.com/jdp/uucp_over_tcp/index.html.
- Craig Sanders's guide for SSL-encrypted UUCP over TCP using stunnel at <http://taz.net.au/postfix/uucp/>.

Here's a graphical description of what this document is about:



And here's the table of contents of this document:

- [Setting up a Postfix Internet to UUCP gateway](#)
- [Setting up a Postfix LAN to UUCP gateway](#)

Setting up a Postfix Internet to UUCP gateway

Here is how to set up a machine that sits on the Internet and that forwards mail to a LAN that is connected via UUCP. See the [LAN to UUCP gateway](#) section for the other side of the story.

- You need an **rmail** program that extracts the sender address from mail that arrives via UUCP, and that feeds the mail into the Postfix **sendmail** command. Most UNIX systems come with an **rmail** utility. If you're in a pinch, try the one bundled with the Postfix source code in the **auxiliary/rmail** directory.
- Define a [pipe\(8\)](#) based mail delivery transport for delivery via UUCP:

```
/etc/postfix/master.cf:
    uucp      unix      -       n       -       -       pipe
               flags=F user=uucp argv=uux -r -n -z -a$sender - $nexthop!rmail ($recipient)
```

This runs the **uux** command to place outgoing mail into the UUCP queue after replacing \$nexthop by the next-hop hostname (the receiving UUCP host) and after replacing \$recipient by the recipients. The [pipe\(8\)](#) delivery agent executes the **uux** command without assistance from the shell, so there are no problems with shell meta characters in command-line parameters.

- Specify that mail for *example.com*, should be delivered via UUCP, to a host named *uucp-host*:

```
/etc/postfix/transport:
    example.com      uucp:uucp-host
```

```
.example.com      uucp:uucp-host
```

See the [transport\(5\)](#) manual page for more details.

- Execute the command "**postmap /etc/postfix/transport**" whenever you change the **transport** file.
- Enable **transport** table lookups:

```
/etc/postfix/main.cf:
    transport_maps = hash:/etc/postfix/transport
```

Specify **dbm** instead of **hash** if your system uses **dbm** files instead of **db** files. To find out what map types Postfix supports, use the command "**postconf -m**".

- Add *example.com* to the list of domains that your site is willing to relay mail for.

```
/etc/postfix/main.cf:
    relay_domains = example.com ...other relay_domains...
```

See the [relay_domains](#) configuration parameter description for details.

- Execute the command "**postfix reload**" to make the changes effective.

Setting up a Postfix LAN to UUCP gateway

Here is how to relay mail from a LAN via UUCP to the Internet. See the [Internet to UUCP gateway](#) section for the other side of the story.

- You need an **rmail** program that extracts the sender address from mail that arrives via UUCP, and that feeds the mail into the Postfix **sendmail** command. Most UNIX systems come with an **rmail** utility. If you're in a pinch, try the one bundled with the Postfix source code in the **auxiliary/rmail** directory.
- Specify that all remote mail must be sent via the **uucp** mail transport to your UUCP gateway host, say, *uucp-gateway*:

```
/etc/postfix/main.cf:
    relayhost = uucp-gateway
    default_transport = uucp
```

Postfix 2.0 and later also allows the following more succinct form:

```
/etc/postfix/main.cf:
    default_transport = uucp:uucp-gateway
```

- Define a [pipe\(8\)](#) based message delivery transport for mail delivery via UUCP:

```
/etc/postfix/master.cf:
    uucp      unix      -      n      n      -      -      pipe
               flags=F user=uucp argv=uux -r -n -z -a$sender - $nexthop!rmail ($recipient)
```

This runs the **uux** command to place outgoing mail into the UUCP queue. It substitutes the next-hop hostname (*uucp-gateway*, or whatever you specified) and the recipients before executing the command. The **uux** command is executed without assistance from the shell, so there are no problems with shell meta characters.

- Execute the command "**postfix reload**" to make the changes effective.

Introduction aux tables de

correspondances

Introduction

Ce document aborde les sujets suivants :

- Le modèle de table de correspondance de Postfix
- Listes et tables de Postfix
- Préparer Postfix pour les consultations LDAP ou SQL
- Maintenir les fichiers tables de correspondances de Postfix
- Mettre à jour les fichiers BD Berkeley en sûreté
- Types de tables de correspondances de Postfix

Le modèle de table de correspondance de Postfix

Postfix utilise les tables de correspondances pour stocker et lire les informations de contrôle d'accès, de réécriture d'adresses et pour le filtrage du contenu. Toutes ces tables sont indiquées dans le fichier `main.cf` sous la forme "type:table", où "type" est l'une des bases de données décrites au paragraphe "Postfix lookup table types" à la fin de ce document, et où "table" est le nom de la table de correspondance. La documentation de Postfix utilise les termes "base de données" et "table de correspondance" pour la même chose.

Exemples de tables de correspondances apparaissant souvent dans la documentation de Postfix :

```
/etc/postfix/main.cf:
alias_maps = hash:/etc/postfix/aliases          (alias locaux)
header_checks = regexp:/etc/postfix/header_checks (filtrage du contenu)
transport_maps = hash:/etc/postfix/transport    (table de routage)
virtual_alias_maps = hash:/etc/postfix/virtual  (réécriture des adresses)
```

Toutes les tables de correspondances de Postfix stockent les informations sous la forme de paires (clef, valeur). Cet interface peut paraître simpliste au premier abord, mais elle est très efficace. L'interface de requête (clef, valeur) masque complètement la complexité LDAP ou SQL de Postfix.

Bénéfices de l'interface de requête (clef, valeur) :

- Vous pouvez implémenter d'abord les tables de correspondances de Postfix avec des bases locales Berkeley (fichiers) et après basculer sur LDAP ou MySQL sans aucun impact sur la configuration de Postfix comme décrit au paragraphe "Préparer Postfix pour les consultations LDAP ou SQL" plus bas.
- Vous pouvez utiliser les fichiers Berkeley DB avec des correspondances fixes pour des opérations simples de réécriture d'adresses et utiliser des tables d'expressions rationnelles pour des travaux plus complexes.

Listes et tables de Postfix

Beaucoup de tables de correspondances de Postfix sont utilisées. Par exemple les réécritures d'adresses (la clef est l'ancienne adresse et la valeur la nouvelle) ou les contrôles d'accès (la clef est le client et la valeur correspond à l'action comme "reject").

Avec certaines tables, Postfix n'a besoin que de savoir si la clef existe. Le résultat de la consultation n'est pas utilisé lui-même. Par exemple, la table passée en paramètre à local_recipient_maps détermine les destinataires locaux que Postfix accepte dans le courrier issu du réseau, celle passée au paramètre mydestination les domaines livrés localement et celle passée à mynetworks les adresses IP considérées sûres. Techniquement, ce sont des listes et non des tables. En dépit de la différence, les listes de Postfix sont décrites ici car elles utilisent la même infrastructure que les tables de correspondances.

Préparer Postfix pour les consultations LDAP ou SQL

SQL et LDAP sont des systèmes complexes. Essayer de mettre en uvre simultanément Postfix et LDAP ou SQL n'est pas une bonne idée. Vous pouvez vous éviter de nombreux problèmes en implémentant d'abord Postfix avec des fichiers locaux type Berkeley DB. Ces fichiers cachent peu de surprises et sont faciles à déboguer avec la commande postmap(1) :

```
% postmap -q info@exemple.com hash:/etc/postfix/virtual
```

Une fois que vos fichiers locaux fonctionnent correctement, vous pouvez suivre les instructions des pages de manuel ldap_table(5), mysql_table(5) ou pgsql_table(5) et remplacer ces fichiers par des requêtes LDAP ou SQL. Dans ce cas, vous pouvez toujours utiliser la commande postmap(1) pour vérifier que les bases de données fournissent bien le même résultat :

```
% postmap -q info@exemple.com ldap:/etc/postfix/virtual.cf
```

Vérifiez toutes requêtes à base d'adresses partielles ou sous-domaines qui sont documentées au paragraphe "ordre de recherche dans les tables" dans la page de manuel correspondante : access(5), canonical(5), virtual(5), transport(5), ou dans la documentation du paramètre de configuration correspondant : mynetworks, relay_domains et parent_domain_matches_subdomains.

Maintenir les fichiers tables de correspondances de Postfix

Lorsque vous effectuez des changements dans la base de données lorsque le système de messagerie est en fonctionnement, il est souhaitable que Postfix sache que les informations ont changé. Le faire sans avoir à exécuter "Postfix reload" améliore nettement les performances car le lancement de cette commande ralentit sensiblement le serveur le temps du rechargement.

- Si vous changez une base en réseau comme LDAP, NIS ou SQL, il n'y a pas besoins de lancer "postfix reload". Les serveurs LDAP, NIS ou SQL gèrent les conflits de lecture/écriture et fournissent les nouvelles données à Postfix dès qu'elles sont disponibles.
- Si vous changez un fichier d'expressions rationnelles regex ou pcre, Postfix reliera peut-être le fichier immédiatement. C'est parce que Postfix lit le fichier entier lorsqu'il en a besoin et le garde en mémoire sans rexaminer le fichier.

- ◆ Si le fichier est utilisé par un programme lancé ponctuellement tel smtpd(8), cleanup(8) ou

- local(8), il n'est pas nécessaire de lancer "postfix reload" après modification.
- ◆ Si le fichier est utilisé par un processus lancé longtemps tel trivial-rewrite(8) sur un serveur chargé, il est nécessaire de lancer "postfix reload".
 - Si vous changez un fichier local type DBM ou Berkeley DB, il n'est pas nécessaire de lancer "postfix reload". Postfix utilise des verrous pour éviter les conflits d'écriture/lecture et chaque fois qu'un démon découvre qu'un fichier a changé, il se termine avant de prendre une nouvelle requête ainsi le nouveau processus peut utiliser la nouvelle base de données.

Mettre à jour les fichiers BD Berkeley en sûreté

Comme Postfix utilise des verrous de fichiers pour éviter les conflits lors des mises à jour des bases Berkeley DB ou des autres fichiers bases de données locaux, vous pouvez rencontrer un problème lors des mises à jour, car les commandes postmap(1) ou postalias(1) écrasent le fichier existant. Si la mise à jour rate au milieu alors la base sera inutilisable et Postfix arrêtera son travail. Ce problème ne se pose pas avec les bases de données de type CDB disponibles à partir de la version 2.2 de Postfix, car la reconstruction des bases CDB est atomique.

Avec des bases sur fichiers multiples comme DBM, il n'y a pas de solution simple. Avec les bases Berkeley DB et les autres bases à fichier unique, il est possible d'ajouter une sécurité en utilisant la commande "mv" pour remplacer un fichier existant au lieu de l'écraser :

```
# postmap access.in && mv access.in.db access.db
```

Ceci convertit le fichier "access.in" en une base "access.in.db" et remplace le fichier "access.db" seulement si la commande postmap(1) se termine avec succès. Pour améliorer ce dispositif, beaucoup utilisent "make" comme montré ci-dessous. Les entrées utilisateur correspondent aux caractères en gras.

```
# cat Makefile
all: aliases.db access.db virtual.db ...etc...

# Note 1: les commandes sont insérées après un caractère tabulation.
# Note 2: utilise postalias(1) for les alias locaux, postmap(1) pour le reste.
aliases.db: aliases.in
    postalias aliases.in
    mv aliases.in.db aliases.db

access.db: access.in
    postmap access.in
    mv access.in.db access.db

virtual.db: virtual.in
    postmap virtual.in
    mv virtual.in.db virtual.db

...etc...
# vi access.in
...session d'édition non montrée...
# make
postmap access.in
mv access.in.db access.db
#
```

La commande "make" ne met à jour que les fichiers qui ont changé. En cas d'erreur, elle s'arrête et n'appelle pas la commande "mv" ainsi Postfix continue de fonctionner avec l'ancienne base comme si de rien n'était.

Types de table de correspondances de Postfix

Pour obtenir la liste des types de bases de données que votre système Postfix supporte, utilisez la commande "postconf -m" command. Ci-dessous une liste des bases souvent supportées :

btree

Une structure en arbre balancé et trié. Ce n'est valable que si votre système supporte les bases Berkeley DB. Les fichiers sont créés avec les commandes postmap(1) ou postalias(1). Le nom de la table de correspondance utilisée dans "btree:table" est le nom du fichier base de données sans l'extension ".db".

cdb

Une structure optimisée pour la lecture sans support des mises à jour incrémentales. Les fichiers sont créés avec les commandes postmap(1) ou postalias(1). La nom de la table de correspondance utilisé dans "cdb:table" est le nom de du fichier de la base de données sans le suffixe ".cdb". Cette fonctionnalité est disponible à partir de la version 2.2de Postfix.

cidr

Une table qui associe des valeurs avec des expressions "Classless Inter-Domain Routing" (CIDR). Le format de ces tables est décrit à la page de manuel cidr_table(5).

dbm

Un fichier indexé basé sur des hachages. Ce n'est disponible que si votre système supporte les bases Berkeley DB. Les fichiers sont créés avec les commandes postmap(1) ou postalias(1). Le nom de la table de correspondance utilisée dans "dbm:table" est le nom du fichier base de données sans l'extension ".dir" ou ".pag".

environ

Le tableau de variables d'environnement UNIX. La clef est le nom de la variable. Le nom de la table de correspondances passé dans "environ:table" est ignoré.

hash

Un fichier indexé basé sur des hachages. Ce n'est disponible que si votre système supporte les bases Berkeley DB. Les fichiers sont créés avec les commandes postmap(1) ou postalias(1). Le nom de la table de correspondance utilisée dans "hash:table" est le nom du fichier base de données sans l'extension ".db".

ldap (*lecture seule*)

Recherche les correspondances en utilisant le protocole LDAP. Ce type de configuration est détaillé à la page de manuel ldap_table(5).

mysql (*lecture seule*)

Recherche les correspondances en utilisant une base de données MySQL. Ce type de configuration est détaillé à la page de manuel mysql_table(5).

netinfo (*lecture seule*)

Recherche les correspondances en utilisant les bases Netinfo.

nis (*lecture seule*)

Recherche les correspondances en utilisant les bases NIS.

nisplus (*lecture seule*)

Recherche les correspondances en utilisant les bases NIS+. Ce type de configuration est détaillé à la page de manuel nisplus_table(5).

pcre (*lecture seule*)

Une table de correspondances basée sur les expressions rationnelles compatibles Perl. Le format du fichier est décrit à la page de manuel pcre_table(5). Le nom de la table de correspondances utilisé dans "pcre:table" est celui du fichier.

pgsql (*lecture seule*)

Recherche les correspondances en utilisant une base de données PostgreSQL. Ce type de configuration est détaillé à la page de manuel [pgsql_table\(5\)](#).

proxy (*lecture seule*)

Accède aux informations via le service [proxymap\(8\)](#) de Postfix. La syntaxe du nom de la table de correspondances est "[proxy](#):type:table".

regexp (*lecture seule*)

Une table de correspondances basée sur les expressions rationnelles POSIX. Le format du fichier est décrit à la page de manuel [regexp_table\(5\)](#). Le nom de la table de correspondances utilisé dans "[regexp](#):table" est celui du fichier.

sdbm

Un type de fichier indexé par hachage. Ce n'est disponible que sur les systèmes qui supportent les bases de données SDBM. Les fichiers sont créés avec les commandes [postmap\(1\)](#) ou [postalias\(1\)](#). Le nom de la table de correspondance utilisée dans "sdbm:table" est le nom du fichier base de données sans l'extension ".dir" ou ".pag".

static (*lecture seule*)

Retourne toujours le nom de la table de correspondances comme résultat. Par exemple, la table de correspondances "static:foobar" retourne toujours la chaîne "foobar" comme résultat de la consultation.

tcp

Accède aux informations par un serveur TCP/IP. Le protocole est décrit à la page de manuel [tcp_table\(5\)](#). Le nom de la table de correspondances est "[tcp](#):machine:port" où "machine" indique un nom de machine ou une adresse IP et "port" un nom de service ou un numéro de port. Ce protocole n'est disponible qu'à partir de la version 2.1.

unix (*lecture seule*)

Une possibilité limitée pour interroger une base d'authentification UNIX. Les tables suivantes sont implémentées :

unix:passwd.byname

La table est la base de données de mots de passe UNIX. La clef est le nom de login. Le résultat est une entrée au format du fichier [passwd\(5\)](#).

unix:group.byname

La table est la base de données des groupes UNIX. La clef est le nom de groupe. Le résultat est une entrée au format du fichier [group\(5\)](#).

D'autres types de table de correspondances peuvent être disponible sur votre système suivant la manière dont à été compilé Postfix. Avec certaines distributions, la liste est extensible dynamiquement lorsque le support des tables de correspondances est lié dynamiquement à Postfix.

Howto CDB de Postfix

Introduction

CDB (Constant DataBase) est un format de fichier indexé inventé par Daniel Bernstein. CDB est optimisé exclusivement pour l'accès en lecture et garantit que chaque enregistrement sera lu en au plus deux accès au disque. Ceci est réalisé en excluant le support des mises à jour incrémentales : aucun enregistrement ou effacement n'est supporté. Les bases de données CDB ne peuvent être modifiées qu'en les reconstruisant complètement, d'où leur qualificatif de "constantes".

Les bases de données CDB de Postfix sont utilisées avec le préfixe "cdb:*nom*", où *nom* indique le nom du fichier CDB sans le suffixe ".cdb" (un autre suffixe, ".tmp", est utilisé temporairement tant que le fichier CDB file est en construction). Les bases de données CDB sont maintenues avec les commandes postmap(1) ou postalias(1). La page DATABASE_README présente les informations générales sur les bases de données de Postfix.

Le support CDB est disponible sur les versions 2.2 et supérieures de Postfix. Ce document montre comment compiler Postfix avec le support CDB.

Compiler Postfix avec CDB

Postfix est compatible avec deux implémentations de CDB :

- La librairie CDB originale de Daniel Bernstein, disponible sur <http://cr.yp.to/cdb.html>, et
- tinycdb (versions 0.5 et supérieures) de Michael Tokarev, disponible sur <http://www.corpit.ru/mjt/tinycdb.html>.

Tinycdb est préférable, car il est un peu plus rapide, dispose de fonctionnalités pratiques et est plus simple à utiliser.

Pour compiler Postfix après avoir installé tinycdb, utilisez quelque chose comme ça :

```
% make tidy
% CDB=../../../../tinycdb-0.5
% make -f Makefile.init makefiles "CCARGS=-DHAS_CDB -I$CDB" \
  "AUXLIBS=$CDB/libcdb.a"
% make
```

Autrement, pour la version de D.J.B. de CDB:

```
% make tidy
% CDB=../../../../cdb-0.75
% make -f Makefile.init makefiles "CCARGS=-DHAS_CDB -I$CDB" \
  "AUXLIBS=$CDB/cdb.a $CDB/alloc.a $CDB/buffer.a $CDB/unix.a $CDB/byte.a"
% make
```

Documentation de Postfix en français

Après que Postfix ait été compilé avec le support CDB, vous pouvez utiliser les tables "cdb" partout où pouvez utiliser en lecture seule les tables "hash", "btree" ou "dbm". Bien entendu, les commandes en ligne "**postmap -i**" (insertion incrémentale d'un enregistrement) et "**postmap -d**" (effacement incremental d'un enregistrement) ne sont pas disponibles. Pour les mêmes raisons, les correspondances type "cdb" ne peuvent être utilisées pour stocker le cache persistant des vérifications d'adresse du service verify(8).

Howto bases de données Berkeley

Introduction

Postfix utilise des bases de données de différentes sortes pour stocker et consulter des informations. Les bases de données de Postfix sont indiquées comme "type:nom". Les bases Berkeley implémentent les bases Postfix de type "hash" et "btree". Le nom d'une base Berkeley de Postfix est le nom de la base de données sans le suffixe ".db". Ces bases Berkeley sont maintenues avec la commande postmap(1).

Note : La version 4 des bases Berkeley n'est pas supportée sur les versions antérieures à la version 2.0 de Postfix.

Ce document décrit :

1. Comment compiler Postfix sur des systèmes sans librairie Berkeley DB.
2. Comment compiler Postfix sur les systèmes BSD ou Linux avec de multiples versions de Berkeley DB.
3. Comment optimiser les performances.
4. Problèmes en cas d'absence de la librairie pthread.

Comment compiler Postfix sur des systèmes sans librairie Berkeley DB

Beaucoup d'UNIX commerciaux sont vendus sans support Berkeley DB, comme Solaris, HP-UX, IRIX, UNIXWARE. Pour compiler Postfix avec le support Berkeley DB, vous devez télécharger et installer le code source depuis le site <http://www.sleepycat.com/>.

Attention : certains systèmes Linux utilisent Berkeley DB, ainsi que des bibliothèques tierces telles SASL. Si vous compilez Postfix avec différentes implémentation Berkeley DB, alors chaque programme de Postfix plantera à cause d'une librairie système, SASL ou Postfix lui même s'arrêtera en utilisant une mauvaise version.

Les versions les plus récentes de Berkeley DB disposent d'une option "--with-uniquename", permettant ainsi à plusieurs versions de Berkeley DB peuvent co-exister dans la même application. Bien qu'inutile, ce peut être le seul moyen d'éviter des incidents.

Pour compiler Postfix après avoir installé Berkeley DB depuis <http://www.sleepycat.com/>, utilisez :

```
% make tidy
% make makefiles CCARGS="-DHAS_DB -I/usr/local/BerkeleyDB/include" \
    AUXLIBS="-L/usr/local/BerkeleyDB/lib -ldb"
% make
```

Pour les systèmes Solaris :

```
% make tidy
```

```
% make makefiles CCARGS="-DHAS_DB -I/usr/local/BerkeleyDB/include" \
    AUXLIBS="-R/usr/local/BerkeleyDB/lib -L/usr/local/BerkeleyDB/lib -ldb"
% make
```

Le chemin exact dépend de la version que vous avez installé. Par exemple, la version 2 de Berkeley DB s'installe dans /usr/local/BerkeleyDB.

Attention : le format de fichier produit par la version 1 de Berkeley DB n'est pas compatible avec les versions 2 et 3 (ces dernières ont le même format). Si vous changez de version de base de données vous devrez recompiler tous vos fichiers de base de données de Postfix.

Attention : si vous utilisez les versions 2 et supérieures de Berkeley DB, n'activez pas le mode de compatibilité avec les bases 1.85, cela empêcherait le verouillage du fichier fcntl.

Attention : si vous utilisez Perl pour manipuler les fichiers Berkeley DB de Postfix, vous devez utiliser la même version dans Perl et Postfix.

Compiler Postfix sur les systèmes BSD avec de multiples versions de Berkeley DB

Certains systèmes BSD fonctionnent avec de multiples implémentations de Berkeley DB. Normalement, Postfix est compilé avec la version de base par défaut de votre système.

Pour compiler Postfix sur des systèmes BSD avec une version particulière de base de données, utilisez des commandes dérivant de :

```
% make tidy
% make makefiles CCARGS=-I/usr/include/db3 AUXLIBS=-ldb3
% make
```

Attention : le format de fichier produit par la version 1 de Berkeley DB n'est pas compatible avec les versions 2 et 3 (ces versions utilisent le même format). Si vous changez de version, vous devrez reconstruire toutes vos bases Postfix.

Attention : si vous utilisez les versions 2 ou supérieures de Berkeley DB, n'activez pas le mode de compatibilité DB 1.85, cela empêcherait le verouillage du fichier fcntl.

Attention : si vous utilisez Perl pour manipuler les fichiers Berkeley DB de Postfix, vous devez utiliser la même version dans Perl et Postfix.

Compiler Postfix sur les systèmes Linux avec de multiples versions de Berkeley DB

Certains systèmes Linux fonctionnent avec de multiples implémentations de Berkeley DB. Normalement, Postfix est compilé avec la version de base par défaut de votre système.

Attention : certaines bibliothèques du système Linux utilisent Berkeley DB. Si vous compilez Postfix avec une autre bibliothèque Berkeley DB que la bibliothèque par défaut, chaque programme de Postfix risque de planter à cause d'une bibliothèque système ou Postfix risque de s'arrêter lui même en utilisant la mauvaise version.

Sur Linux, vous devez éditer le script `makedefs` pour spécifier la librairie autre que par défaut car l'emplacement du fichier `include "db.h"` change en fonction des vendeurs et des versions.

Attention : le format de fichier produit par la version 1 de Berkeley DB n'est pas compatible avec les versions 2 et 3 (ces versions utilisent le même format). Si vous changez de version, vous devrez reconstruire toutes vos bases Postfix.

Attention : si vous utilisez les versions 2 ou supérieures de Berkeley DB, n'activez pas le mode de compatibilité DB 1.85, cela empêcherait le verouillage du fichier `fcntl`.

Attention : si vous utilisez Perl pour manipuler les fichiers Berkeley DB de Postfix, vous devez utiliser la même version dans Perl et Postfix.

Optimiser les performances

Postfix fournit deux paramètres de configuration qui contrôlent la quantité de mémoire utilisée par Berkeley DB.

- `berkeley_db_create_buffer_size` (défaut : 16 Mo par table). Ce paramètre est utilisé par les commandes qui maintiennent les fichiers Berkeley DB : `postalias(1)` et `postmap(1)`. Pour les fichiers "hash", les performances de création se dégradent rapidement à moins que la réserve de mémoire soit 0 (taille de fichier). Pour les fichiers "btree", ces performances sont bonnes avec des entrées triées même pour de petites réserves de mémoire, mais se dégradent vite avec des entrées non triées à moins que la réserve de mémoire soit 0 (taille de fichier). *(Note du traducteur : je ne suis vraiment pas sûr de la traduction de ce paragraphe. Une petite aide serait la bienvenue...)*
- `berkeley_db_read_buffer_size` (défaut : 128 ko par table). Ce paramètre est utilisé par tous les programmes de Postfix. La taille du buffer est adaptée à la lecture. Si le cache est plus petit que la table, la lecture aléatoire est très dépendante de la taille du cache, sauf avec les tables btree où la taille du cache doit être assez grande pour contenir le chemin entier du nud principal. Une évidence empirique montre que 64ko sont suffisant. Nous doublons cette taille pour fonctionner en toute sécurité et anticiper les changements dans l'implémentation.

Problèmes en cas d'absence de la librairie pthread

Lorsque la compilation de Postfix échoue avec un message du type :

```
undefined reference to `pthread_condattr_setpshared'
undefined reference to `pthread_mutexattr_destroy'
undefined reference to `pthread_mutexattr_init'
undefined reference to `pthread_mutex_trylock'
```

Ajoutez la librairie `"-lpthread"` à la commande `"make makefiles"`.

```
% make makefiles .... AUXLIBS="... -lpthread"
```

Pour plus d'information, reportez-vous à la page <http://www.sleepycat.com/>.

Postfix LDAP Howto

Support de LDAP dans Postfix

Postfix peut utiliser un annuaire LDAP comme source pour toutes ses correspondances : [aliases\(5\)](#), [virtual\(5\)](#), [canonical\(5\)](#), etc. Ceci vous permet de stocker les informations de votre système de messagerie dans une base de données protégée par un contrôle d'accès fin. En ne les stockant pas localement sur le serveur de messagerie, les administrateurs peuvent les maintenir depuis n'importe où et les utilisateurs peuvent accéder et modifier les informations que vous souhaitez. Vous pouvez également avoir de multiples serveurs utilisant la même information sans les conflits et délais de recopie sur chacun d'eux.

Sujets abordés dans ce document :

- [Compiler Postfix avec le support LDAP](#)
- [Configurer les correspondances par consultation LDAP](#)
- [Exemple : alias](#)
- [Exemple : domaines/adresses virtuels](#)
- [Autres utilisations des consultations LDAP](#)
- [Notes et éléments à prendre en compte](#)
- [Retour d'expérience](#)
- [Références](#)

Compiler Postfix avec le support LDAP

Note 1 : Postfix ne supporte plus la version 1 du protocole LDAP.

Note 2 : pour utiliser LDAP avec le Postfix de Debian GNU/Linux's, tout ce dont vous avez besoin est d'installer le package postfix-ldap. Il n'y a pas lieu de recompiler Postfix.

Vous devez disposer des bibliothèques et fichiers "include" sur votre système et configurer le Makefile de Postfix en fonction.

Par exemple pour compiler les bibliothèques d'OpenLDAP pour les utiliser avec Postfix (c'est à dire les codes client LDAP seulement), vous pouvez utiliser la commande suivante :

```
% ./configure --without-kerberos --without-cyrus-sasl --without-tls \  
--without-threads --disable-slapi --disable-slurpd \  
--disable-debug --disable-shared
```

Si vous utilisez les bibliothèques de la distribution UM (<http://www.umich.edu/~dirsvcs/ldap/ldap.html>) ou OpenLDAP (<http://www.openldap.org>), les commandes suivantes à la racine des sources de Postfix devraient suffire :

```
% make tidy  
% make makefiles CCARGS="-I/usr/local/include -DHAS_LDAP" \  
AUXLIBS="-L/usr/local/lib -lldap -L/usr/local/lib -llber"
```

Sur Solaris 2.x vous devrez spécifier le chemin des librairies sinon, ld.so ne trouvera pas les librairies partagées :

```
% make tidy
% make makefiles CCARGS="-I/usr/local/include -DHAS_LDAP" \
  AUXLIBS="-L/usr/local/lib -R/usr/local/lib -lldap \
    -L/usr/local/lib -R/usr/local/lib -llber"
```

La commande 'make tidy' n'est nécessaire que si vous avez précédemment compilé Postfix sans le support LDAP.

Au lieu de '/usr/local' indiquez la position de vos librairies LDAP et fichier include. Assurez-vous de ne pas mélanger des librairies et fichiers include de différentes versions!!

Si vos librairies LDAP sont compilées avec le support Kerberos, vous devrez également inclure vos librairies Kerberos sur cette ligne. Notez que les librairies KTH Kerberos IV peuvent entrer en conflit avec la librairie lib/libdns.a de Postfix qui définit dns_lookup. Si cela arrive, vous devrez probablement utiliser des librairies LDAP sans support de Kerberos pour compiler Postfix et il ne supportera pas les connexions Kerberos au serveur LDAP. Désolé...

Si vous utilisez un des SDK LDAP Netscape, vous devrez changer la ligne AUXLIBS pour pointer sur libldap10.so ou libldapssl30.so ou ce dont vous disposez, et vous devrez utiliser l'option de liage appropriée (-R) pour que l'exécutable la trouve au lancement.

Configurer les correspondances par consultation LDAP

Pour utiliser les correspondances LDAP, définissez une source LDAP comme table de correspondance dans le fichier main.cf, par exemple :

```
alias_maps = hash:/etc/aliases, ldap:/etc/postfix/ldap-aliases.cf
```

Le fichier /etc/postfix/ldap-aliases.cf peut contenir un grand nombre de paramètres, y compris les paramètres qui activent LDAP SSL et STARTTLS. Pour une description complète de ces possibilités, consultez la page de manuel ldap_table(5).

Exemple: alias locaux

Ci-dessous un exemple d'utilisation de LDAP pour les consultations d'alias locaux (local(8)). Supposons que dans le fichier main.cf, vous avez :

```
alias_maps = hash:/etc/aliases, ldap:/etc/postfix/ldap-aliases.cf
```

et dans ldap:/etc/postfix/ldap-aliases.cf :

```
server_host = ldap.my.com
search_base = dc=my, dc=com
```

A la réception d'un message à destination de l'adresse locale "ldapuser" qui n'est pas trouvée dans la base de données /etc/aliases, Postfix recherchera le serveur LDAP écoutant sur le port 389 de ldap.my.com. Il se connectera anonymement, cherchera toute entrée dont l'attribut mailacceptinggeneralid est "ldapuser", lira l'attribut "maildrop" des entrées trouvées et construira une liste de leur maildrops qui seront traitées comme

des adresses RFC822 à qui le message sera livré.

Exemple: domaines/adresses virtuelles

Si vous voulez stocker vos informations pour les consultations virtuelles dans votre annuaire, c'est seulement un petit peu plus compliqué. D'abord vous devez vous assurer de la configuration du domaine virtuel dans Postfix. Ensuite, vous devrez vous assurer que tous les attributs `mailacceptinggeneralid` de vos destinataires virtuels ont une forme correcte et dans le domaine virtuel. Finalement, si vous devez désigner une entrée de cet annuaire comme adresse par défaut pour le domaine virtuel, ajoutez simplement un champ `mailacceptinggeneralid` (ou équivalent dans votre annuaire) contenant "@domaine.virtuel". Si vous ne désirez pas d'adresse de collecte, omettez simplement cette étape et le courrier des utilisateurs inconnus sera retourné.

En résumé, l'enregistrement de l'utilisateur de collecte devrait ressembler à :

```
dn: cn=defaultrecipient, dc=fake, dc=dom
objectclass: top
objectclass: virtualaccount
cn: defaultrecipient
owner: uid=root, dc=someserver, dc=isp, dc=dom
1 -> mailacceptinggeneralid: fake.dom
2 -> mailacceptinggeneralid: @fake.dom
3 -> maildrop: realuser@real.dom
```

- 1: Postfix sait que fake.dom est un domaine virtuel valide lorsqu'il le cherche et obtient quelque chose (maildrop) en réponse.
- 2: Ceci implémente l'adresse de collecte : le courrier perdu est redirigé sur cette entrée ...
- 3: ... et va ensuite dans sa boîte-aux-lettres.

Les utilisateurs normaux auront simplement un `mailacceptinggeneralid` et un maildrop, à savoir "utilisateur.normal@fake.dom" et "utilisateur.normal@real.dom".

Autres utilisations des consultations LDAP

Les consultations LDAP peuvent être utilisées à d'autres fins y compris les réécritures d'adresses d'émission ou de destination avec les correspondances canoniques de Postfix. Par exemple, pour modifier les adresses "login@site.dom" en "prénom.nom@site.dom".

Notes et éléments à prendre en compte

- Les éléments de schéma et les noms d'attributs utilisés dans cette page sont juste des exemples. Seule remarque, certains sont des valeurs par défaut des configurations LDAP. Vous pouvez utiliser n'importe quel schéma et configurer Postfix en conséquence.
- Vous devrez probablement vous assurer que les `mailacceptinggeneralids` sont uniques et que les utilisateurs ne peuvent les modifier.
- Une entrée peut avoir plusieurs `mailacceptinggeneralids` ou maildrops. Les maildrops peuvent également contenir plusieurs adresses séparées par des virgules. Elles seront toutes retournées par la consultation. Par exemple, vous pouvez définir une entrée pour implémenter une liste de diffusion ressemblant à ceci (Attention! Ce schéma n'est construit que pour l'exemple) :

```
dn: cn=Accounting Staff List, dc=my, dc=com
cn: Accounting Staff List
```

```
o: my.com
objectclass: maillist
mailacceptinggeneralid: accountingstaff
mailacceptinggeneralid: accounting-staff
maildrop: mylist-owner
maildrop: an-accountant
maildrop: some-other-accountant
maildrop: this, that, theother
```

- Si vous utilisez les consultations LDAP pour autre chose que les alias, vous devrez vous assurer que ces consultations ont un sens. Dans le cas de correspondances virtuelles les enregistrement maildrop ne contenant pas d'adresse mail sont déconseillées car Postfix ne peut savoir sous quel utilisateur lancer le programme ou ouvrir le fichier de livraison. Votre requête query_filter devrait ressembler à :

```
query_filter = (&(mailacceptinggeneralid=%s)(!(|(maildrop="*"*) (maildrop="*:")))(mai
```

- Dans ce cas comme pour les alias, vous ne souhaitez probablement pas que les utilisateurs modifient leur enregistrement maildrop. Ceci peut être particulièrement pertinent sur un serveur "scellé" où ils n'ont pas de compte UNIX mais n'existent que dans LDAP et Cyrus. Vous souhaitez également que si le champ maildrop contient un programme et que l'objet n'appartient pas à "cn=root", rien ne soit retourné. Ceci requiert de l'attention de votre part pour implémenter ceci en toute sécurité, considérant les ramifications de ce type de livraison. Si vous décidez qu'il n'est intéressant d'autoriser tous ces non-sens dans les consultations LDAP, supprimez-les du filtre query_filter et utilisez d'autres artifices tels les listes Majordomo ou les bases de données locales d'alias.

```
query_filter = (&(mailacceptinggeneralid=%s)(!(|(maildrop="*"*) (maildrop="*:")))(mai
```

- Les requêtes LDAP sont plus lentes que les requêtes aux bases locales DB ou DBM. Pour la plupart des sites, ce ne sera pas un goulot d'entrangement, mais il est bon de savoir comment optimiser votre annuaire LDAP.
- Les correspondances LDAP multiples partagent la même connexion LDAP si elles ne diffèrent que dans les paramètres de leur requête : base, scope, query_filter, etc. Pour profiter de cet avantage, évitez les fausses différences dans vos définitions LDAP : ordre de selection des machines, version, connexion, paramètres tls, ... doivent être les mêmes pour le plus de correspondances possible.

Retours d'expérience

Si vous avez des questions, envoyez-les à postfix-users@postfix.org. N'oubliez pas les informations sur votre configuration : paramètres LDAP issus de postconf, quelles librairies LDAP vous avez utilisé et quel serveur d'annuaire vous utilisez. Si votre question porte sur le contenu des entrées de votre annuaire, incluez s'il vous plait quelques entrées;

Références

- Manuel Guesdon: Spotted a bug with the timeout attribute.
- John Hensley: Multiple LDAP sources with more configurable attributes.
- Carsten Hoeger: Search scope handling.
- LaMont Jones: Domain restriction, URL and DN searches, multiple result attributes.
- Mike Mattice: Alias dereferencing control.
- Hery Rakotoarisoa: Patches for LDAPv3 updating.
- Prabhat K Singh: Wrote the initial Postfix LDAP lookups and connection caching.
- Keith Stevenson: [RFC 2254](#) escaping in queries.
- Samuel Tardieu: Noticed that searches could include wildcards, prompting the work on [RFC 2254](#)

escaping in queries. Spotted a bug in binding.

- Sami Haahtinen: Referral chasing and v3 support.
- Victor Duchovni: ldap_bind() timeout. With fixes from LaMont Jones: OpenLDAP cache deprecation. Limits on recursion, expansion and query results size. LDAP connection sharing for maps differing only in the query parameters.
- Liviu Daia: Support for SSL/STARTTLS. Support for storing map definitions in external files (ldap:/path/ldap.cf) needed to securely store passwords for plain auth.
- Liviu Daia a révisé l'interface de configuration et ajouté les fonctionnalités de configuration dans main.cf.
- Liviu Daia avec quelques améliorations de Jose Luis Tallon et Victor Duchovni, a développé l'interface des requêtes courantes, du format des résultats (result_format), du domaine et des limites d'expansion pour LDAP, MySQL et PostgreSQL (*Ndlr: traduction approximative*).

Et bien sur Wietse.

Howto MySQL Postfix

Introduction

Le type de tables de correspondances "mysql" vous permettent connecter Postfix à une base de données MySQL. Cette implémentation autorise l'emploi de plusieurs bases de données : vous pouvez en utiliser une pour une table virtual(5), un pour une table d'accès(5) une pour les alias(5), etc. Vous pouvez spécifier plusieurs serveurs pour la même base de données, ainsi Postfix peut en changer en cas d'incident sur la première.

Les serveurs chargés utilisant les correspondances mysql généreront beaucoup de clients mysql concurrents, le serveur devra être conçu en conséquence. Vous pouvez réduire ce nombre de clients mysql en utilisant le service proxymap(8) de Postfix.

Compiler Postfix avec le support de MySQL

Note : pour utiliser mysql avec le serveur Postfix de Debian GNU/Linux, tout ce que vous avez à faire est d'installer le package postfix-mysql et c'est tout. Il n'est pas nécessaire de recompiler Postfix.

Le client MySQL de Postfix utilise la librairie cliente mysql qui peut être obtenue à l'adresse suivante :

<http://www.mysql.com/downloads/>
<http://sourceforge.net/projects/mysql/>

Pour compiler Postfix avec le support des correspondances mysql, vous devrez ajouter `-DHAS_MYSQL` et `-I` pointant sur le répertoire contenant les headers de mysql et la librairie mysqlclient (et libm) à `AUXLIBS`, par exemple :

```
% make -f Makefile.init makefiles \  
  'CCARGS=-DHAS_MYSQL -I/usr/local/mysql/include' \  
  'AUXLIBS=-L/usr/local/mysql/lib -lmysqlclient -lz -lm'
```

Lancez simplement 'make'. Ceci nécessite la librairie de compression libz. Les implementations plus anciennes de MySQL se compilent sans libz.

Utiliser des tables MySQL

Une fois que Postfix est compilé avec le support MySQL, vous pouvez l'utiliser dans main.cf comme suit :

```
alias_maps = mysql:/etc/postfix/mysql-aliases.cf
```

Le fichier /etc/postfix/mysql-aliases.cf peut contenir beaucoup d'informations indiquant à Postfix comment référencer la base de données MySQL. Pour une description complète, reportez-vous à la page de manuel mysql_table(5).

Exemple: alias locaux

```
#
# fichier de configuration MySQL pour les consultations des aliases(5)
# par le démon local(8)
#

# Le nom d'utilisateur et le mot de passe de connexion au serveur mysql
user = someone
password = some_password

# Le nom de la base de données
dbname = customer_database

# Pour Postfix 2.2 et supérieurs, le modèle de requête SQL.
# Voir mysql_table(5) pour les détails
query = SELECT forw_addr FROM mxaliases WHERE alias='%s' AND status='paid'

# Pour les versions de Postfix antérieures à la 2.2.
# Voir mysql_table(5) pour les détails
select_field = forw_addr
table = mxaliases
where_field = alias
# N'oubliez pas le "AND" !
additional_conditions = AND status = 'paid'
```

Notes complémentaires

L'installation de l'interface de configuration de MySQL permet l'emploi de plusieurs bases de données : vous pouvez en utiliser une pour la table virtual, une pour la table d'accès et une pour les alias si vous le souhaitez.

Puisque les sites qui ont besoin de multiples serveurs de messagerie apprécient la possibilité d'utiliser une base de données en réseau mais ne veulent pas avoir un point de fragilité en ayant qu'une base, nous avons inclus le possibilité de référencer plusieurs machines pour accéder à la même table. Ceci fonctionnera si les sites mettent en place au moins une base miroir. Chaque fois qu'une requête échouera, les autres serveurs seront utilisés dans un ordre aléatoire. Si aucun serveur MySQL n'est disponible, le courrier sera retardé jusqu'à ce qu'un serveur soit joignable.

Références

- Scott Cotton et Joshua Marcus (IC Group, Inc) ont contribué à la version initiale.
- Liviu Daia a révisé l'interface de configuration interface et ajouté les paramètres de configuration du fichier main.cf.
- Liviu Daia avec quelques améliorations de Jose Luis Tallon et Victor Duchovni, a développé l'interface des requêtes courantes, du format des résultats (result_format), du domaine et des limites d'expansion pour LDAP, MySQL et PostgreSQL (*Ndlr: traduction approximative*).

Support PCRE de Postfix

Support des expressions rationnelles PCRE (Perl Compatible Regular Expressions)

Les expressions rationnelles PCRE vous permettent d'utiliser la syntaxe des expressions rationnelles de Perl tel `\s` pour un espace ou `\S` pour le contraire. Cependant, le principal bénéfice est que les consultations PCRE sont souvent plus rapides que les `regex` car les implémentations PCRE sont souvent plus efficaces que les implémentations des expressions rationnelles POSIX que vous trouvez sur vos systèmes.

Vous trouverez une description de l'emploi des tables PCRE comprenant des exemples à la page de manuel `pcre_table(5)`. Pour plus d'information sur PCRE lui-même, reportez-vous à la page <http://www.pcre.org/>.

Compiler Postfix avec le support PCRE

Note : pour utiliser PCRE avec le serveur Postfix de Debian GNU/Linux, tout ce que vous avez à faire est d'installer le package `postfix-pcre`. Il n'y a pas à recompiler Postfix.

Dans de prochaines versions, Postfix disposera d'une interface permettant d'ajouter des types de tables par `plug-in`. Jusque là, vous devez compiler le support PCRE dans Postfix.

En premier lieu, vous devez disposer de la librairie PCRE (Perl Compatible Regular Expressions), qui peut être trouvée à l'adresse suivante :

<ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/>.

NOTE : les versions antérieures à la version 2.06 de `pcre` ne peuvent pas être utilisées.

Pour compiler Postfix avec le support PCRE vous devez ajouter `-DHAS_PCRE` et un `-I` pointant sur le répertoire include de PCRE, et ajouter le chemin vers la librairie PCRE à `AUXLIBS`, par exemple :

```
make -f Makefile.init makefiles \
    "CCARGS=-DHAS_PCRE -I/usr/local/include" \
    "AUXLIBS=-L/usr/local/lib -lpcre"
```

Solaris peut avoir besoin d'informations sur le chemin du `run-time` :

```
make -f Makefile.init makefiles \
    "CCARGS=-DHAS_PCRE -I/usr/local/include" \
    "AUXLIBS=-L/usr/local/lib -R/usr/local/lib -lpcre"
```

Éléments à connaître

- Lorsque Postfix cherche une table de correspondances `pcre`: ou `regex`: chaque expression est appliquée à la chaîne entière en entrée. Suivant l'application, cette chaîne est nom de client, une

Documentation de Postfix en français

adresse IP ou une adresse de messagerie. Ainsi, aucune recherche n'est effectuée sur le domaine parent : une adresse "utilisateur@domaine" n'est pas coupée en partie utilisateur et domaine, de même pour "utilisateur+foo".

- Les tables d'expressions rationnelles telles pcre: ou regexp: ne sont pas autorisées à faire des substitutions \$nombre dans les résultats de la consultation qui pourrait être sensibles du point de vue de la sécurité : couramment, cette restriction s'applique à la base de données locale des alias(5) ou aux tables de l'agent de livraison virtual(8).

Howto PostgreSQL avec Postfix

Introduction

Le type de table `pgsql` de Postfix vous permet d'utiliser une base de données PostgreSQL avec Postfix. Cette implémentation autorise plusieurs bases `pgsql` : vous pouvez en utiliser une pour une table [virtual\(5\)](#), une pour une table d'[accès](#) et une pour une table d'[alias](#) si vous le souhaitez. Vous pouvez indiquer de multiple serveurs pour la même base de données, ainsi Postfix peut changer de base si une est défaillante.

Les serveurs de messagerie chargés utilisant des tables `pgsql` généreront beaucoup de clients `pgsql` concurrents, le(s) serveur(s) `pgsql` devront être lancés en prenant ceci en considération. Vous pouvez réduire le nombre de clients `pgsql` concurrent en utilisant le service [proxymap\(8\)](#).

Compiler Postfix avec le support PostgreSQL

Note : pour utiliser `pgsql` avec le Postfix de Debian GNU/Linux's, tout ce que vous avez à faire est d'installer le package `postfix-pgsql`. Il n'y a pas à recompiler Postfix.

Pour compiler Postfix avec le support `pgsql`, indiquez `-DHAS_PGSQL`, le répertoire contenant les fichiers "include" de PostgreSQL et l'emplacement de la librairie `libpq`.

Par exemple :

```
% make tidy
% make -f Makefile.init makefiles \
    'CCARGS=-DHAS_PGSQL -I/usr/local/include/pgsql' \
    'AUXLIBS=-L/usr/local/lib -lpq'
```

Lancez ensuite 'make'.

Configurer les tables de correspondances PostgreSQL

Une fois Postfix compilé avec le support `pgsql`, vous pouvez indiquer un type de correspondance dans [main.cf](#) comme ceci :

```
/etc/postfix/main.cf :
    alias\_maps = pgsql:/etc/postfix/pgsql-aliases.cf
```

Le fichier `/etc/postfix/pgsql-aliases.cf` contient de nombreuses informations indiquant à Postfix comment référencer la base de données `pgsql`. Pour une description complète, reportez-vous à la page de manuel [pgsql table\(5\)](#).

Exemple: local aliases

```
#
# fichier de configuration pgsql pour les alias de local\(8\)
```



```
#

#
# les machines auxquelles Postfix doit se connecter
hosts = host1.mon.domaine host2.mon.domaine

# Le nom d'utilisateur et le mot-de-passe pour se connecter
user = quelquun
password = mot-de-passe

# Le nom de la base de données
dbname = customer_database

# Postfix 2.2 et supérieurs utilisent le modèle de requête SQL. Voir pgsql_table(5).
query = SELECT adresse FROM alias_mx WHERE alias='%s' AND status='payé'

# Pour les versions < 2.2. Voir pgsql_table(5) pour plus de détails.
select_field = adresse
table = alias_mx
where_field = alias
# N'oubliez pas l'indispensable "AND" !
additional_conditions = AND status = 'payé'
```

Utiliser des bases de données miroir

Les sites qui ont besoin de plusieurs échangeurs de messagerie apprécient la possibilité d'utiliser une base de données en réseau, mais ne veulent pas introduire ainsi un point vulnérable dans leur système.

Pour cette raison, nous avons inclut la possibilité d'indiquer plusieurs machines à Postfix pour accéder à la même base pgsql. Ceci fonctionne si les sites mettent en oeuvre des bases en miroir sur plusieurs machines.

Si une requête échoue sur une machine, les autres seront essayées dans un ordre aléatoire. Si aucun serveur pgsql n'est joignable, le message est retardé jusqu'à ce que l'un d'entre eux soit joignable.

Credits

- Ce code est basé sur le code mysql de Postfix écrit par Scott Cotton et Joshua Marcus, de "IC Group, Inc".
- L'adaptation PostgreSQL a été réalisée par Aaron Sethman.
- La mise à jour pour Postfix 1.1.x et PostgreSQL 7.1+ et le support de l'appel aux procédures stockées a été ajouté par Philip Warner.
- LaMont Jones a été le premier mainteneur de Postfix pgsql.
- Liviu Daia a révisé l'interface de configuration et ajouté la fonctionnalité de configuration de main.cf.
- Liviu Daia avec plusieurs améliorations de Jose Luis Tallon et Victor Duchovni a développé les interfaces de requête commune, result_format, domain et expansion_limit pour LDAP, MySQL et PostgreSQL.

Postfix VERP Howto

Postfix VERP support

Les version 1.1 et supérieures de Postfix supportent les chemins d'adresses de retour variables dans l'enveloppe. Lorsque la livraison type VERP est demandée, chaque destinataire d'un message reçoit une copie personnalisée de ce message, avec sa propre adresse de destination encodée dans l'adresse d'expédition de l'enveloppe.

Par exemple, lorsque la livraison type VERP demandée, Postfix délivre le message avec comme expéditeur "nom-de-la-liste@origine" pour un destinataire "user@domain", avec l'adresse d'expédition encodant le destinataire comme suit :

```
owner-listname+user=domain@origin
```

Ainsi, le courrier non livrable peut révéler l'adresse de destination non livrable sans imposer au propriétaire de la liste de traiter les rejets.

Le concept VERP a été popularisé par la MTA qmail MTA et par le gestionnaire de listes ezmlm. Lisez <http://cr.yp.to/proto/verp.txt> pour connaître les idées ayant initié ce concept.

Sujets abordés par ce document :

- Paramètres de configuration VERP de Postfix
- Utiliser VERP avec les gestionnaires de liste Majordomo etc...
- Le support VERP dans le serveur SMTP de Postfix
- Le support VERP dans la commande sendmail de Postfix
- Le support VERP dans le serveur QMOP de Postfix

Paramètres de configuration VERP de Postfix

Avec Postfix, l'ensemble du processus est contrôlé par quatre paramètres de configuration.

default_verp_delimiters (défaut : +=)

Caractères utilisés par Postfix comme délimiteur VERP lorsque ce type de livraison est requis sans que soit précisé les délimiteurs.

verp_delimiter_filter (défaut : -+=)

Caractères acceptés par Postfix comme délimiteurs VERP sur la ligne de la commande sendmail et dans les commandes SMTP. De nombreux caractères ne doivent pas être utilisés comme délimiteur VERP, soit parce qu'ils ont déjà un sens particulier dans les adresses mail (tels @ et %), soit parce qu'ils sont utilisés comme partie des noms d'utilisateurs ou des noms de domaines (tels les caractères alphanumériques), soit parce qu'ils sont des caractères non-ASCII ou de contrôle. Et comme chacun sait, certains caractères peuvent engendrer des bugs dans les logiciels vulnérables, et nous ne souhaitons pas que cela arrive.

smtpd_authorized_verp_clients (défaut : aucun)

Clients SMTP autorisés à demander la livraison type VERP. Le serveur QMQP de Postfix utilise son propre mécanisme de contrôle d'accès et les soumissions locales (via /usr/sbin/sendmail etc.) sont toujours autorisées. Pour autoriser une machine, listez son nom, adresse IP, sous-réseau (réseau/masque) ou son .domaine parent.

Avec les versions 1.1 et 2.0 de Postfix, ce paramètre est appelé authorized_verp_clients (défaut : \$mynetworks).

disable_verp_bounces (défaut : no)

Détermine si Postfix envoie un rapport de rejet global pour les messages VERP multi-destinataires, ou un rapport par destinataire. Par défaut, un rapport par destinataire comme demandé par ezmlm.

Utiliser VERP avec les gestionnaires de liste Majordomo etc...

Pour utiliser VERP avec les listes de diffusions Majordomo etc., vous devrez configurer le gestionnaire de liste pour soumettre les messages sous l'une de ces deux formes :

Postfix 2.3 et supérieurs :

```
% sendmail -V -f owner-listname other-arguments...  
% sendmail -V+= -f owner-listname other-arguments...
```

Postfix 2.2 et antérieurs (Postfix 2.3 comprend l'ancienne syntaxe à des fins de compatibilité, mais il enregistre un avertissement qui vous rappelle que vous utilisez une syntaxe obsolète) :

```
% sendmail -V -f owner-listname other-arguments...  
% sendmail -V+= -f owner-listname other-arguments...
```

La première forme utilise les caractères de délimitation VERP par défaut de main.cf. La seconde forme vous autorise à indiquer ces caractères. L'exemple montre les valeurs recommandées.

Ce texte suppose que vous avez configuré un alias propriétaire des listes qui route le courrier non livrable à une personne réelle :

```
/etc/aliases:  
owner-listname: votre-nom+nom-de-liste
```

Pour traiter les rejets nous allons faire une utilisation extensive des tours d'extensions d'adresses.

Vous devez indiquer à Postfix que + est le séparateur entre une adresse et ces extensions optionnelles, que les extensions d'adresses sont ajoutées au noms des fichiers .forward file names, et que les extensions d'adresses sont ignorées lors des substitutions d'alias :

```
/etc/postfix/main.cf:  
recipient_delimiter = +  
forward_path = $home/.forward${recipient_delimiter}${extension},  
$home/.forward  
propagate_unmatched_extensions = canonical, virtual
```

(les deux derniers paramètres ont la valeur par défaut).

Vous devez créer un fichier nommé `.forward+listname` avec les commandes qui traitent tous les messages envoyés à l'adresse du propriétaire de la liste :

```
~/.forward+listname:
    "/some/where/command ..."
```

Avec cette configuration, le courrier non livrable à destination de `user@domain` sera retourné à l'adresse suivante :

```
owner-listname+user=domain@votre.domaine
```

qui est traitée par votre commande située dans le fichier `.forward+listname`. Ce message devrait contenir, entre autres, un en-tête `To:` avec l'adresse d'expédition encapsulée dans l'adresse de destination :

```
To: owner-listname+user=domain@your.domain
```

Il est laissé comme exercice au lecteur l'extraction des valeurs `user=domain` de l'adresse de destination de l'en-tête `To:`.

Le support VERP dans le serveur SMTP de Postfix

Le serveur SMTP de Postfix implémente une commande `XVERP` pour activer la livraison type VERP. La syntaxe autorise deux formes :

```
MAIL FROM:<expéditeur@domaine> XVERP

MAIL FROM:<expéditeur@domaine> XVERP+=
```

La première forme utilise les délimiteurs VERP par défaut de `main.cf`, la seconde forme les surcharge explicitement. Les valeurs montrées sont celles recommandées.

Le support VERP dans la commande `sendmail` de Postfix

La commande `sendmail` de Postfix dispose d'une option `-V` pour demander la livraison type VERP. Indiquez l'une des deux formes suivantes :

Postfix 2.3 et supérieurs

```
% sendmail -XV -f owner-listname ....

% sendmail -XV+= -f owner-listname ....
```

Postfix 2.2 et antérieurs (Postfix 2.3 comprend l'ancienne syntaxe à des fins de compatibilité, mais il enregistre un avertissement qui vous rappelle que vous utilisez une syntaxe obsolète) :

```
% sendmail -V -f owner-listname ....

% sendmail -V+= -f owner-listname ....
```

La première forme utilise les délimiteurs VERP par défaut de `main.cf`, la seconde forme les surcharge explicitement. Les valeurs montrées sont celles recommandées.

Le support VERP dans le serveur QMQP de Postfix

Lorsque le serveur QMQP de Postfix reçoit le message avec une adresse d'expédition dans l'enveloppe sous la forme :

```
listname-@your.domain-@[ ]
```

Postfix génère les adresses d'expédition "listname-user=domain@your.domain", en utilisant "-=" comme délimiteurs VERP car gmail/ezmlm les imposent.

Plus généralement, une adresse d'expédition of "prefix@origin-@[]" requiert une livraison type VERP avec des adresses d'expéditions sous la forme "prefixuser=domain@origin". Toutefois, Postfix autorise seulement les délimiteurs VERP qui sont indiqués avec le paramètre verp_delimiter_filter. En particulier, le délimiteur "=" est requis pour la compatibilité gmail (lisez la page de manuel addresses(5) de gmail pour plus de détails).

Postfix et Linux

Éléments de configuration des bases Berkeley DB

Attention : si vous ne pouvez pas compiler Postfix parce que le fichier "db.h" n'est pas trouvé, vous DEVEZ installer la package de développement Berkeley DB (nom de package : db???-dev???) fourni par votre distribution. Seul ce package contient les fichiers correspondant à la version de Berkeley DB utilisée par les librairies du systèmes.

NE téléchargez PAS de versions de Berkeley DB du réseau. Chaque programme de Postfix plantera s'il est compilé avec une version différente de Berkeley DB que celle utilisée par les librairies du système. Lisez le fichier DB_README pour plus d'informations.

Éléments de configuration de Procmail

Sur RedHat Linux 7.1 **procmail** ne dispose pas des droits d'écriture sur le répertoire /var/spool/mail. Contournement : `chmod 1777 /var/spool/mail`.

Performances de Syslogd

Le démon **syslogd** de Linux utilise l'écriture synchrone par défaut. Pour cette raison, **syslogd** peut parfois utiliser plus de ressources système que Postfix. Pour éviter ceci, désactivez l'écriture synchrone du fichier journal du courrier en éditant /etc/syslog.conf et en faisant précéder d'un – ce fichier :

```
/etc/syslog.conf:
mail.*                -/var/log/mail.log
```

Envoyez un "**kill –HUP**" à **syslogd** pour activer ces changements.

Postfix et NFS

Cette question a été demandée sur la liste de diffusion postfix-users il y a quelques temps :

En outre, comment sont gérés le verrouillage des fichiers et autres problèmes potentiels lorsque Postfix fonctionne avec une boîte type Netapp pour la livraison dans /var/mail ? Il est connu que FreeBSD a un verrouillage de fichier NFS défaillant (clients et serveurs tous deux ?) mais je (Wietse) ne sais pas si Postfix peut le contourner.

Postfix passe par plusieurs étapes pour traiter les problèmes spécifiques NFS. Ainsi, Postfix sur NFS est légèrement moins fiable que sur un disque local. Ceci n'est pas un problème de Postfix mais de NFS et affecte les autres MTA également.

Pour le verrouillage des files d'attente, NFS ne pose pas de problèmes car on ne peut partager les files d'attente de Postfix entre différentes instances.

Pour obtenir le verrouillage des fichiers boîtes-aux-lettres sur NFS, vous devez tout configurer pour utiliser les verrous fcntl() (ou passer à la livraison type maildir, qui ne nécessite pas de contrôle niveau applicatif des verrous).

Pour activer les verrous fcntl() sur les fichiers boîtes-aux-lettres avec Postfix, indiquez :

```
/etc/postfix/main.cf:  
    virtual mailbox lock = fcntl  
    mailbox delivery lock = fcntl
```

Malheureusement, cette approche n'est utilisable seulement si tous les autres logiciels accédant aux boîtes-aux-lettres utilisent les verrous fcntl().

Vous pouvez également "jouer sûr" et utiliser les fichiers *utilisateur.lock* files :

```
/etc/postfix/main.cf:  
    virtual mailbox lock = fcntl, dotlock  
    mailbox delivery lock = fcntl, dotlock
```

C'est une combinaison que beaucoup d'applications terminales utilisent.

Postfix and Ultrix

Postfix on Ultrix

This document is probably only of historical value, because Ultrix version 4 dates from the early 1990s. However, as long as Wietse keeps Postfix alive for SunOS 4, it is likely to run on Ultrix 4 with very little change. Feedback is welcome if anyone actually still uses Postfix on any version of Ultrix.

The source of this document is an email message by Christian von Roques that was sent on Jun 2, 1999.

I've upgraded the MTA of our DECstation-3100 running Ultrix4.3a to postfix-19990317-pl05 and am sending you the patches I needed to get it running under Ultrix.

...

One of the bugs of Ultrix's /bin/sh is that shell-variables set in arguments of ':' expand to garbage if expanded in here-documents. Using a different shell helps. I needed to replace all calls of ``sh ../makedefs" by ``\$(SHELL) ../makedefs" in all the Makefile.in and am now able to use ``make SHELL=/bin/sh5" or zsh.

...

Ultrix's FD_SET_SIZE is 4096, but getdtablesize() returns 64 by default, if not increased when building a new kernel. getrlimit() doesn't know RLIMIT_NOFILE. This makes event_init() always log the warning: 'could allocate space for only 64 open files'.

I just reduced the threshold from 256 to 64, but this is not good. The initial problem still remains: How to disable this warning on Ultrix without making the source ugly?

To work around the first problem, all the Makefile.in files have been updated to use '\$(SHELL)' instead of 'sh'. So you only need to supply a non-default shell in order to eliminate Ultrix shell trouble.

To work around the latter, util/sys_defs.h was updated for Ultrix, with a default FD_SETSIZE of 100. This should be sufficient for a workstation. Even in 1999, no-one would run a major mail hub on Ultrix 4.

Postfix + Maildrop Howto

Introduction

Ce document présente différentes façons d'utiliser l'agent de livraison maildrop dans Postfix :

- Livraison directe sans utiliser l'agent local de livraison
- Livraison indirecte via l'agent local de livraison
- Références

Livraison directe sans utiliser l'agent local de livraison

Postfix peut être configuré pour livrer le courrier directement à maildrop sans utiliser l'agent de livraison local(8) comme intermédiaire. Ceci signifie que vous n'aurez pas les substitutions des alias locaux ou le traitement des fichiers \$HOME/.forward. Vous pourrez typiquement utiliser pour les domaines hébergés avec des destinataires qui n'ont pas de répertoires individuels UNIX (\$HOME).

Les exemples suivants montrent comment utiliser maildrop pour un.domaine et pour un.autre.domaine.

```
1 /etc/postfix/main.cf:
2     maildrop_destination_recipient_limit = 1
3     virtual_mailbox_domains = un.domaine un.autre.domaine
4     virtual_transport = maildrop
5     virtual_mailbox_maps = hash:/etc/postfix/virtual_mailbox
6     virtual_alias_maps = hash:/etc/postfix/virtual_alias
7
8 /etc/postfix/virtual_mailbox:
9     user1@un.domaine      ...texte inutilisé ici...
10    user2@un.domaine      ...texte inutilisé ici...
11    user3@un.autre.domaine ...texte inutilisé ici...
12
13 /etc/postfix/virtual_alias:
14    postmaster@un.domaine      postmaster
15    postmaster@un.autre.domaine postmaster
```

- La ligne 2 est nécessaire pour que Postfix appelle l'agent de livraison maildrop pour chaque destinataire individuellement.
- La ligne 3 informe Postfix que un.domaine et un.autre.domaine sont des domaines virtuels de boîtes-aux-lettres. Au lieu de lister ces noms dans le fichier main.cf, vous pouvez également utiliser un fichier ; reportez-vous à la page virtual_mailbox_domains pour plus de détails.
- La ligne 4 indique que le courrier pour les domaines virtuels de boîtes-aux-lettres (un.domaine et un.autre.domaine ici) doivent être livrés via l'agent de livraison maildrop.
- Lignes 5 et 8 à 11 : indiquez les destinataires que le serveur SMTP de Postfix doit accepter comme destinataire. Ceci évite l'encombrement des files d'attente par des messages non livrables. Indiquez une valeur vide ("virtual_mailbox_maps =") pour désactiver cette fonctionnalité.
- Lignes 6 et 13 à 15 : redirige le courrier du postmaster des domaines virtuels vers le postmaster local. La RFC 821 requiert une adresse postmaster par domaine.

L'UID vmail utilisé ci-dessous est le compte sous lequel doit être lancé maildrop. Il doit être le propriétaire des boîtes-aux-lettres virtuelles si elles ont le même propriétaire. Si maildrop a le bit suid (voir la documentation de maildrop), alors maildrop changera automatiquement de compte pour livrer le courrier.

Note : N'utilisez pas l'utilisateur postfix pour maildrop.

```
/etc/postfix/master.cf:
maildrop unix -      n      n      -      -      pipe
      flags=DRhu user=vmail argv=/path/to/maildrop -d ${recipient}
```

Si vous voulez supporter les adresses du style user+extension@domain, utilisez ce qui suit à la place :

```
/etc/postfix/master.cf:
maildrop unix -      n      n      -      -      pipe
      flags=DRhu user=vmail argv=/path/to/maildrop
      -d ${user}@${nexthop} ${extension} ${recipient} ${user} ${nexthop}
```

Le courrier est livré à \${user}@\${nexthop} (clef correspondant pour la recherche dans la base des utilisateurs maildrop). L'\${extension} et les autres composants sont disponibles pour les règles maildrop sous \$1, \$2, \$3, ... et peut être ommit dans master.cf ou ignoré par maildrop s'ils ne sont pas utilisés.

Livraison indirecte via l'agent local de livraison

Postfix peut être configuré pour livrer le courrier à maildrop via l'agent local de livraison. Ceci est légèrement moins efficace que l'approche "directe" décrite ci-dessus, mais présente l'intérêt des substitutions d'[alias](#) locaux et de l'exploitation des fichiers \$HOME/.forward. Vous pourrez typiquement l'utiliser pour les domaines listés dans [mydestination](#) et qui n'ont pas de compte du système UNIX.

Pour configurer la livraison maildrop pour tous les comptes du système UNIX :

```
/etc/postfix/main.cf:
      mailbox_command = /path/to/maildrop -d ${USER}
```

Note : \${USER} est épilé en majuscules.

Pour activer la livraison maildrop pour certains utilisateurs seulement, vous pouvez utiliser la fonctionnalité [mailbox_command_maps](#) de l'agent [local\(8\)](#) de livraison :

```
/etc/postfix/main.cf:
      mailbox_command_maps = /etc/postfix/mailbox_commands

/etc/postfix/mailbox_commands:
vous      /path/to/maildrop -d ${USER}
```

La livraison maildrop pour certains utilisateurs est également possible en l'invoquant depuis les fichiers \$HOME/.forward :

```
/home/vous/.forward:
" | /path/to/maildrop -d ${USER} "
```

Références

- Le texte original a été fourni par Russell Mosemann.
- Victor Duchovni a fourni des éléments pour supporter les adresses utilisateur+extension@domaine.
- Tonni Earnshaw a contribué au texte sur la livraison via l'agent de livraison local(8).

Présentation de l'architecture de

Postfix

Introduction

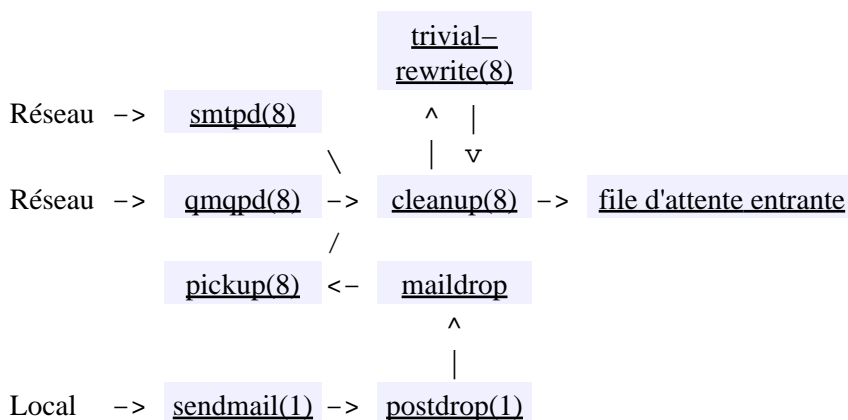
Ce document présente un aperçu de l'architecture de Postfix et les liens vers chacune des commandes ou serveurs de Postfix. Il montre le contexte général dans lequel chaque commande ou serveur est utilisé et indique les pages dans lesquelles vous trouverez plus d'information.

Sujets abordés par ce document :

- [Comment Postfix reçoit le courrier](#)
- [Comment Postfix livre le courrier](#)
- [Ce qui se passe en coulisse](#)
- [Commandes de Postfix](#)

Comment Postfix reçoit le courrier

Lorsqu'un message entre dans le système de messagerie Postfix, le premier point d'arrêt est la [file d'attente entrante](#). Le schéma ci-dessous montre les principaux processus invoqués à l'arrivée d'un nouveau message. Les noms suivis par un nombre désignent des commandes ou programmes de Postfix, et les noms sans chiffre dans une aire grisée représentent des files d'attente.



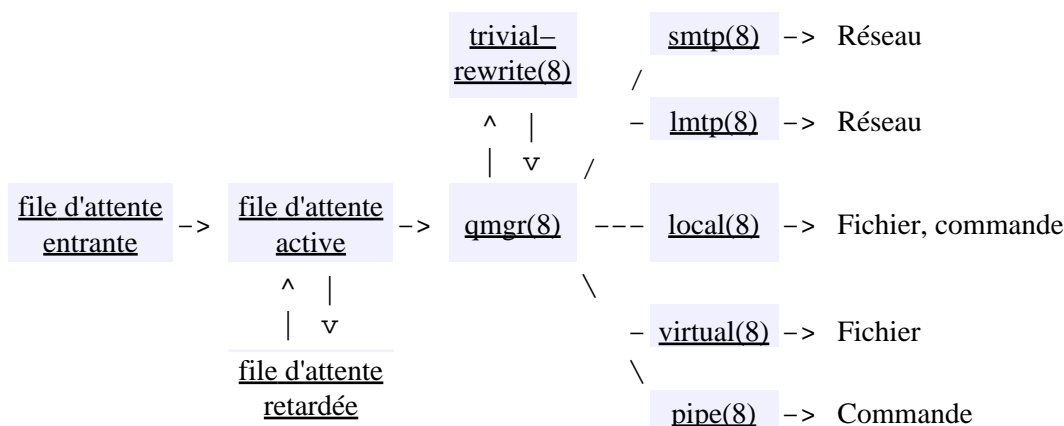
- Les messages du réseau entrent dans Postfix via les serveurs `smtpd(8)` ou `qmqpd(8)`. Ces derniers retirent les enveloppes protocolaires (SMTP ou QMQP), effectuent quelques contrôles de sécurité pour protéger Postfix, et donnent l'expéditeur, les destinataires et le contenu du message au serveur `cleanup(8)`. Le serveur `smtpd(8)` peut être configuré pour bloquer les messages indésirables comme décrit à la page [SMTPD_ACCESS_README](#).
- Les sousmissions locales sont reçues par la commande compatible `sendmail(1)` de Postfix et sont stockés dans la `file d'attente maildrop` par la commande privilégiée `postdrop(1)`. Ce dispositif fonctionne même lorsque le système de messagerie Postfix est arrêté. Le serveur local `pickup(8)` reprend ces sousmissions locales, effectue quelques contrôles de sécurité pour protéger Postfix et

donne l'expéditeur, les destinataires et le contenu du message au serveur [cleanup\(8\)](#).

- Les messages issus de sources internes sont donnés directement au serveur [cleanup\(8\)](#). Ces sources ne sont pas montrées dans le schéma et incluent : les messages transférés par l'agent de livraison [local\(8\)](#) (voir paragraphe suivant), les messages retournés à l'expéditeur par le serveur [bounce\(8\)](#) (voir le deuxième paragraphe suivant) et les notifications au Postmaster à propos des problèmes rencontrés par Postfix.
- Le serveur [cleanup\(8\)](#) implemente le processus final avant que les messages ne soient mis en file d'attente. Il ajoute les From: manquants et d'autres en-têtes de message et transforme les adresses comme décrit à la page [ADDRESS REWRITING README](#). De plus, le serveur [cleanup\(8\)](#) peut être configuré pour effectuer une inspection de contenu légère avec des expressions rationnelles tel que décrit à la page [BUILTIN FILTER README](#). Le serveur [cleanup\(8\)](#) place le résultat dans un fichier dans la [file d'attente entrante](#) et notifie au gestionnaire des files d'attente (voir paragraphe suivant) l'arrivée d'un nouveau message.
- Le serveur [trivial-rewrite\(8\)](#) réécrit les adresses sous la forme standard "utilisateur@domaine.qualifié", comme décrit à la page [ADDRESS REWRITING README](#). Postfix n'implémente pas actuellement de langage de réécriture mais beaucoup peut être fait avec les tables de correspondances et, si besoin, les expressions rationnelles.

Comment Postfix livre le courrier

Une fois le message arrivé dans la [file d'attente entrante](#), l'étape suivante est la livraison. Le schéma ci-dessous montre les principaux composants de l'appareil de livraison de Postfix. Les noms suivis par un nombre désignent des commandes ou programmes de Postfix, et les noms sans chiffre dans une aire grisée représentent des files d'attente.



- Le gestionnaire des files d'attente (le processus serveur [qmgr\(8\)](#) du schéma) est le cur de la livraison de message de Postfix. Il contacte les agents de livraison [smtp\(8\)](#), [lmtp\(8\)](#), [local\(8\)](#), [virtual\(8\)](#), [pipe\(8\)](#), ou [error\(8\)](#), et envoie une requête de livraison pour une ou plusieurs adresses de destination. L'agent de livraison [error\(8\)](#) est special : il déclare toujours le message non livrable. Il n'est pas montré dans le schéma ci-dessus.

Le gestionnaire des files d'attente maintient une [file d'attente active](#) aussi petite que possible avec les messages ouverts pour livraison. La [file d'attente active](#) agit comme une fenêtre limitée sur une [file d'attente entrante](#) ou des [files d'attente retardées](#) potentiellement encombrées. La limitation de la [file d'attente active](#) évite au gestionnaire des files d'attente de fonctionner avec trop de mémoire.

Le gestionnaire des files d'attente maintient une [file d'attente retardée](#) séparée pour les messages qui

ne peuvent être livrés évitant ainsi le ralentissement de l'accès normal aux files d'attente. La stratégie du gestionnaire des files d'attente pour estimer le temps d'attente est décrite aux pages [QSHAPE README](#) et [TUNING README](#).

- Le serveur [trivial-rewrite\(8\)](#) résout chaque adresse de destination suivant ses classes d'adresses locales et distantes, comme définit à la page [ADDRESS CLASS README](#). Des informations complémentaires de routage peuvent être ajoutées à la table optionnelle [transport\(5\)](#). Le serveur [trivial-rewrite\(8\)](#) interroge éventuellement la table [relocated\(5\)](#) pour connaître les destinataires dont l'adresse a changé ; le courrier de ces destinataires est retourné à l'expéditeur avec une explication.
- Le client [smtp\(8\)](#) consulte une liste des échangeurs de messagerie pour obtenir la machine de destination host, la trie par ordre de préférence et essaye chaque serveur jusqu'à ce qu'il trouve un serveur qui lui répond. Il encapsule ensuite l'expéditeur, le destinataire et le contenu du message comme exigé par le protocole SMTP ; ce qui inclut la conversion de l'encodage 8-bit MIME vers l'encodage 7-bit.
- Le client [lmtp\(8\)](#) parle un protocole similaire à SMTP qui est optimisé pour la livraison aux serveurs de boîtes-aux-lettres tels Cyrus. L'avantage de cette configuration est qu'une machine Postfix peut alimenter plusieurs serveurs de boîtes-aux-lettres via LMTP. L'opposé est vrai aussi : un serveur de boîtes-aux-lettres peut être alimenté via LMTP par de multiples machines Postfix. La page [LMTP README](#) montre des exemples de l'emploi du client [lmtp\(8\)](#).
- L'agent de livraison [local\(8\)](#) sait utiliser les boîtes-aux-lettres type UNIX, les fichiers de courrier maildir de qmail, les bases de données d'[alias\(5\)](#) type Sendmail et les fichiers .forward style Sendmail des utilisateurs. Plusieurs agents de livraisons "local" peuvent fonctionner en parallèle, mais la livraison simultanée au même utilisateur est limitée.

L'agent de livraison [local\(8\)](#) peut utiliser des formes de livraison locales alternatives : vous pouvez le configurer pour livrer dans des fichiers boîtes-aux-lettres des répertoires des utilisateurs, pour déléguer la livraison à un programme externe tel procmail ou déléguer la livraison à un agent de livraison de Postfix différent.

- L'agent de livraison [virtual\(8\)](#) est un agent de livraison qui ne livre que dans les boîtes-aux-lettres type UNIX ou les fichiers de courrier maildir de qmail. Cet agent de livraison peut livrer le courrier de plusieurs domaines ce qui le rend particulièrement intéressant pour l'hébergement de nombreux petits domaines sur la même machine. Ceci est décrit à la page [VIRTUAL README](#).
- L'agent [pipe\(8\)](#) est l'interface de sortie vers les autres systèmes de traitement du courrier (la commande Postfix [sendmail\(1\)](#) constitue l'interface d'entrée). Cette interface est compatible UNIX : elle fournit des informations sur la ligne de commande et sur l'entrée standard et s'attend à ce que le processus termine avec un code de statut tel que définit dans <sysexit.h>. Des exemples de livraison via l'agent [pipe\(8\)](#) sont proposés aux pages [MAILDROP README](#) et [UUCP README](#).

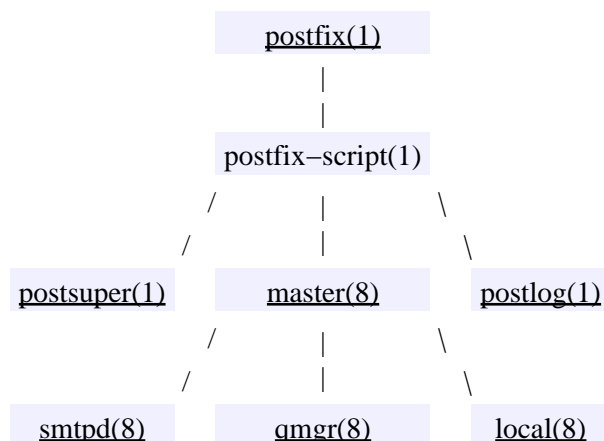
Ce qui se passe en coulisse

Les paragraphes précédents ont montré un aperçu de la réception et de la livraison du courrier par Postfix. Ces processus serveurs sont reliés à d'autres qui effectuent des actions en coulisse. Ce paragraphe tente de montrer chaque service dans son contexte. Comme précédemment, les noms suivis par un nombre désignent des commandes ou programmes de Postfix, et les noms sans chiffre dans une aire grisée représentent des files d'attente.

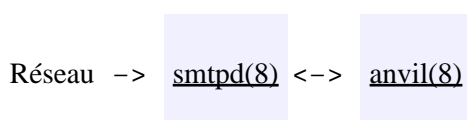
- Le serveur résident [master\(8\)](#) est le superviseur qui garde un œil sur le bon fonctionnement du système de messagerie Postfix. Il est typiquement lancé au démarrage du système avec la commande "postfix start" et reste actif jusqu'à l'arrêt de ce dernier. Le serveur [master\(8\)](#) est responsable du lancement des processus serveur de Postfix pour la réception et la livraison du courrier, et du relancement des serveurs se terminant prématurément suite à un problème. Le serveur [master\(8\)](#) est également

Documentation de Postfix en français

responsable du respect des limites du nombre de processus serveur indiqué dans le fichier de configuration **master.cf**. Le schéma ci-dessous montre la hiérarchie des programmes au lancement de Postfix. Seuls quelques processus démons manipulant le courrier sont montrés.

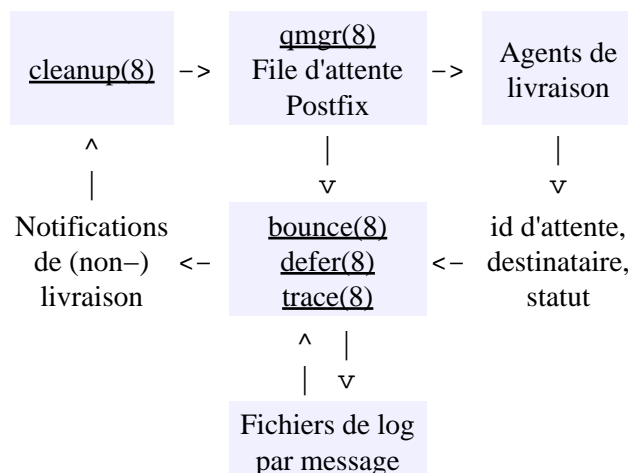


- Le serveur anvil(8) implémente la connexion client et la limite du taux pour tous les serveurs smtpd(8). La page [TUNING README](#) fournit un guide pour l'échange avec des clients SMTP se comportant mal. Le service anvil(8) est disponible dans les versions 2.2 et supérieures de Postfix.

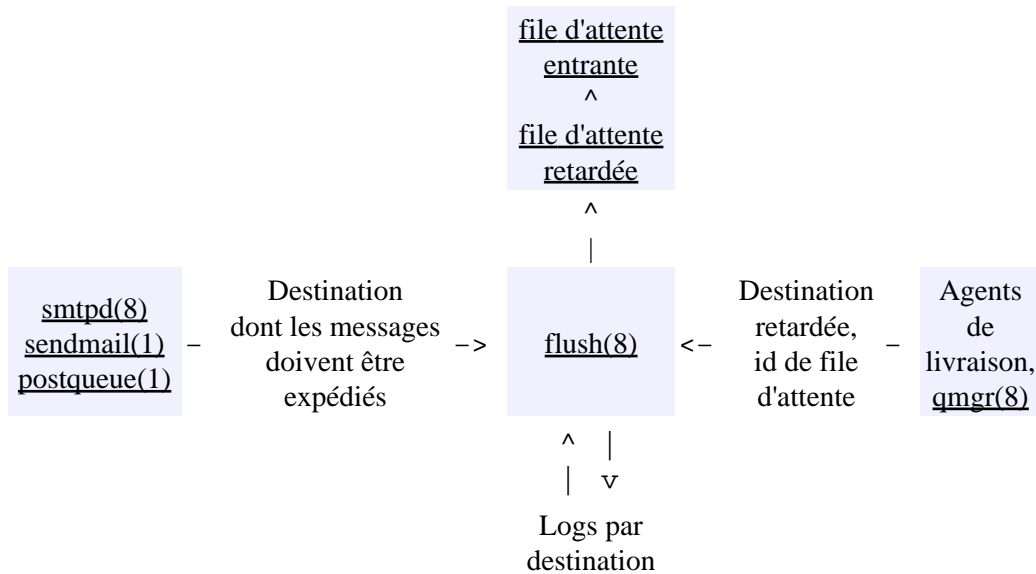


- Les serveurs bounce(8), defer(8) et trace(8) maintiennent chacun leur propre arbre de répertoires de file d'attente avec des fichiers de log par message. Ces informations sont utilisées pour envoyer les notifications de livraison ou de non-livraison à l'expéditeur.

Le service trace(8) implémente le support des commandes "sendmail -bv" et "sendmail -v" qui produisent des rapports sur la livraison de message par Postfix et est disponible sur les versions 2.1 et supérieures de Postfix. Reportez-vous à la page [DEBUG README](#) pour les exemples.



- Les serveurs flush(8) maintiennent les logs par destination et implémentent ETRN et la commande "sendmail -qRdestination", tel que décrit à la page [ETRN README](#). Ils déplacent les fichiers stockés dans les files d'attente retardées dans la file d'attente entrante et demande leur livraison. Le service flush(8) est disponible dans les versions 1.0 et supérieures de Postfix.

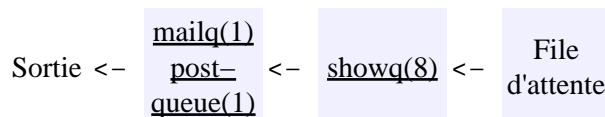


- Les serveurs proxymap(8) fournissent un service de consultation des tables de correspondance aux processus de Postfix. Ceci permet les restrictions chroot, et réduit le nombre de tables de correspondances ouvertes en partageant l'accès à de multiples processus.
- Le serveur scache(8) maintient le cache des sessions pour le client smtp(8) de Postfix. Lorsque le cache des sessions est activé pour des destinations choisies, le client smtp(8) ne se déconnecte pas immédiatement après une transaction, mais donne cette connexion au serveur de cache. Le client smtp(8) continue avec d'autres requêtes de livraison. Parallèlement, le cache de session maintient la connexion ouverte pour un temps limité. Pendant ce temps, tout processus smtp(8) peut demander au serveur scache(8) cette session cachée et l'utiliser pour la livraison d'un message. Postfix limite le temps pendant lequel une connexion peut être réutilisée.

Lors de la livraison vers des destinations disposant de plusieurs serveurs, le cache des connexions peut permettre d'exclure un serveur ne répondant pas, ce qui augmente sensiblement la vitesse de livraison.



- Les serveurs showq(8) listent les statuts des files d'attente de Postfix. Il s'agit du service de listage qui effectue le travail des commandes mailq(1) et postqueue(1).

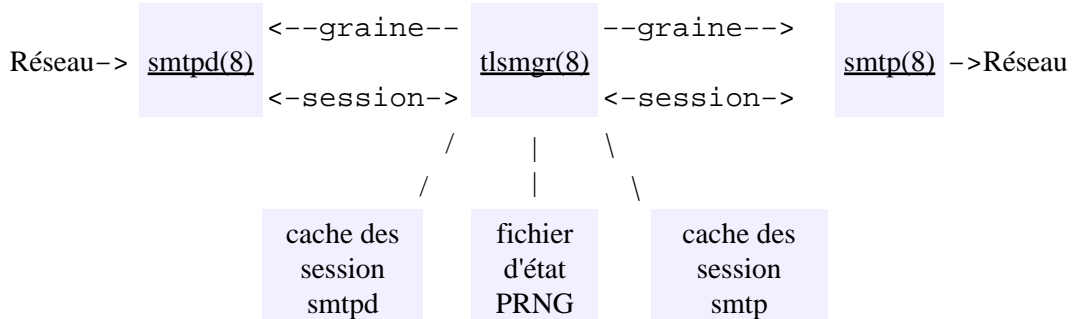


- Les serveurs spawn(8) lancent les commandes non-Postfix à la demande avec les clients connectés par des sockets ou FIFO à l'entrée, la sortie et le flux d'erreur standard de la commande. Vous pouvez trouver des exemples de cet usage à la page SMTPD_POLICY_README.
- Le serveur tlsmgr(8) fonctionne lorsque TLS (Transport Layer Security, connu sous le nom SSL) est activé dans le client smtp(8) ou le serveur smtpd(8). Ce processus poursuit deux buts :
 - ◆ maintenir un générateur de nombres pseudo-aléatoire (PRNG) utilisé pour égrainer les moteurs TLS des processus du client smtp(8) client ou du serveur smtpd(8) de Postfix. L'état du PRNG est régulièrement sauvegardé dans un fichier et est lu au démarrage de tlsmgr(8),

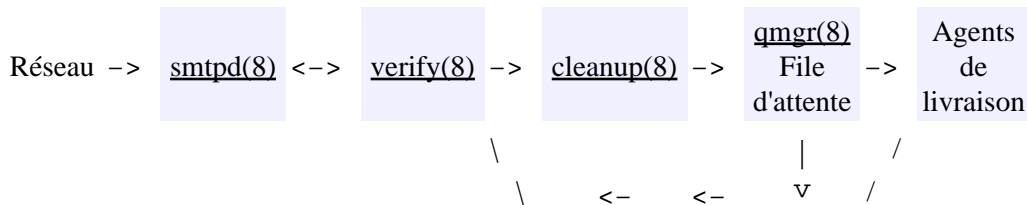
Documentation de Postfix en français

- ◆ maintenir l'optionnel cache des clefs de sessions TLS du client smtp(8) ou du serveur smtpd(8) de Postfix. La sauvegarde des clefs peut améliorer les performances en réduisant le surplus de trafic et de calcul lié au démarrage d'une session TLS.

Le support TLS est disponible à partir de la version 2.2 de Postfix. Les informations sur l'implémentation TLS de Postfix se trouvent à la page [TLS README](#).



- Le serveur verify(8) vérifie que l'adresse d'expédition ou de destination est livrable avant que le serveur smtpd(8) l'accepte. Le serveur verify(8) injecte les messages approuvés dans la file d'attente Postfix et procède à la mise à jour des statuts des agents de livraisons et/ou du gestionnaire des files d'attente. Ce processus est décrit à la page [ADDRESS VERIFICATION README](#). Le service verify(8) est disponible sur les versions 2.1 et supérieures de Postfix.



Commandes de Postfix

Cet aperçu de l'architecture de Postfix se termine avec un résumé des utilitaires en ligne de commande pour l'utilisation au quotidien du système de messagerie Postfix. En plus des commandes compatibles Sendmail sendmail(1), mailq(1), et newaliases(1), le système Postfix fournit sa propre collection d'utilitaires en ligne de commande. Pour plus de consistance, elles sont nommées *postsomething*.

- La commande postfix(1) contrôle les opérations du système de messagerie. C'est l'interface pour démarrer, arrêter et redémarrer le système de messagerie, ainsi que d'autres opérations administratives. Cette commande est réservée au super-utilisateur.
- La commande postalias(1) maintient les bases de données type alias(5) de Postfix. Il s'agit du programme qui effectue le travail de la commande newaliases(1).
- La commande postcat(1) expose le contenu des fichiers de files d'attente de Postfix. Il s'agit d'un utilitaire préliminaire et limité. Ce programme est généralement surclassé par d'autres plus puissants pouvant également éditer les files d'attente.
- La commande postconf(1) expose ou met à jour les paramètres du fichier `main.cf` de Postfix et montre les informations dépendantes du système sur les méthodes de verrouillage et les types de tables de correspondance supportées.
- La commande postdrop(1) est l'utilitaire d'envoi de courrier qui est lancé par la commande Postfix sendmail(1) afin de déposer le courrier dans la file d'attente maildrop.
- La commande postkick(1) crée des canaux internes de communication utilisables, par exemple, avec les scripts shell.

Documentation de Postfix en français

- La commande postlock(1) fournit des verrous de boîtes–aux–lettres compatibles Postfix utilisables, par exemple, avec les scripts shell.
- La commande postlog(1) permet aux scripts shell d'enregistrer des logs compatibles Postfix.
- La commande postmap(1) maintient les tables de correspondances de Postfix telles les tables canoniques(5), virtuelles(5) et autres. C'est un cousin de la commande UNIX makemap.
- La commande postqueue(1) est la commande privilégiée utilisée par les commandes sendmail(1) et mailq(1) de Postfix pour vider ou lister les files d'attente des messages.
- La commande postsuper(1) maintient la file d'attente Postfix. Elle supprime les vieux fichiers temporaires et déplace les fichiers de file d'attente dans les bons répertoires après un changement dans la profondeur des répertoires des files d'attente. Elle est lancée au démarrage du système de messagerie startup et lorsque Postfix est relancé.

Rejeter les destinataires locaux

inconnus avec Postfix

Introduction

Depuis la version 2.0 de Postfix, le serveur SMTP rejette le courrier des destinataires inconnus dans les domaines locaux (domaines qui correspondent à \$mydestination ou adresses IP dans \$inet_interfaces ou \$proxy_interfaces) avec le message "User unknown in local recipient table". Cette fonctionnalité était optionnelle avec les versions antérieures.

La bonne nouvelle est que ceci évite de mettre en file d'attente le courrier non livrable et ne génère ainsi pas de messages MAILER-DAEMON.

La mauvaise nouvelle est que celà rejete le courrier lorsque vous montez d'une version antérieure qui n'était pas configuré pour rejeter le courrier des destinataires locaux inconnus.

Ce document décrit les étapes nécessaires au rejet correct des destinataires locaux inconnus.

- Configurer local_recipient_maps dans main.cf
- Quand devez-vous changer la paramètre local_recipient_maps dans main.cf
- Format de la table des destinataires locaux

Configurer local_recipient_maps dans main.cf

Le paramètre local_recipient_maps indique les tables de correspondances contenant tous les noms ou adresses des destinataires locaux. Une adresse de destination est locale lorsque le domaine correspond à \$mydestination, \$inet_interfaces ou \$proxy_interfaces. Si un nom d'utilisateur local ou une adresse n'est pas dans la liste \$local_recipient_maps, le serveur SMTP de Postfix rejettera l'adresse avec la mention "User unknown in local recipient table".

La valeur par défaut, montrée ci-dessous, suppose que vous utilisez l'agent de livraison local(8) de Postfix avec lequel les destinataires sont des alias locaux ou ont un compte UNIX :

```
/etc/postfix/main.cf:  
local_recipient_maps = proxy:unix:passwd.byname $alias_maps
```

Pour désactiver le rejet des destinataires locaux inconnus, indiquez :

```
/etc/postfix/main.cf:  
local_recipient_maps =
```

C'est à dire une valeur nulle. Avec ce paramètre, le serveur SMTP ne rejettera pas le courrier avec la mention "User unknown in local recipient table".

Quand devez-vous changer la paramètre `local_recipient_maps` dans `main.cf`

- Problème : vous n'utilisez pas l'agent de livraison `local(8)` par défaut de Postfix pour les domaines correspondant à `$mydestination`, `$inet_interfaces` ou `$proxy_interfaces`. Par exemple, vous avez redéfini le paramètre "`local_transport`" dans le fichier `main.cf`.

Solution : votre paramètre `local_recipient_maps` doit indiquer une base de données qui liste tous les noms d'utilisateurs ou adresses connus par cet agent de livraison. Par exemple, si vous livrez les utilisateurs des domaines `$mydestination` etc. via l'agent de livraison `virtual(8)`, indiquez :

```
/etc/postfix/main.cf
mydestination = $myhostname localhost.$mydomain localhost ...
local_transport = virtual
local_recipient_maps = $virtual_mailbox_maps
```

Si vous utilisez un agent de livraison différent pour les domaines `$mydestination` etc., reportez-vous au paragraphe "[Format de la table des destinataires locaux](#)" ci-dessous pour voir comment renseigner cette table.

- Problème : vous utilisez la fonctionnalité `mailbox_transport` ou `fallback_transport` de l'agent de livraison `local(8)` pour livrer le courrier des comptes non-UNIX.

Solution : vous devez ajouter la base de données qui liste les utilisateurs non-UNIX :

```
/etc/postfix/main.cf
local_recipient_maps = proxy:unix:passwd.byname, $alias_maps,
    <la base de données des comtes non-UNIX>
```

Reportez-vous au paragraphe "[Format de la table des destinataires locaux](#)" ci-dessous pour voir comment la table doit être renseignée.

- Problème : vous utilisez la fonctionnalité `luser_relay` de l'agent local de livraison.

Solution : vous devez désactiver complètement la fonctionnalité `local_recipient_maps` pour que Postfix accepte le courrier des adresses locales :

```
/etc/postfix/main.cf
local_recipient_maps =
```

Format de la table des destinataires locaux

Si vous utilisez des fichiers locaux au format `postmap(1)`, `local_recipient_maps` respecte le format de table suivant :

- Coté gauche, indiquez un nom d'utilisateur nu, une carte blanche "`@domain.tld`" ou indiquez une adresse complète "`utilisateur@domain.tld`".
- Vous devez spécifier quelque chose sur la partie droite de la table mais cette valeur sera ignorée.

Si vous utilisez des tables de correspondances basées sur NIS, LDAP, MYSQL, ou PGSQL, alors `local_recipient_maps` effectue la même requête que pour un fichier local au format `postmap(1)` et espère le même résultat.

Documentation de Postfix en français

Avec les tables d'expressions rationnelles, Postfix n'interroge qu'avec l'adresse complète de destination et non avec l'utilisateur nu ou "@domaine.tld".

NOTE : une table de correspondance doit toujours retourner un résultat lorsque l'adresse existe et doit toujours retourner "non trouvé" lorsque l'adresse n'existe pas. En particulier, les résultats de taille nulle comptent comme un résultat "non trouvé".

Classes d'adresses Postfix

Introduction

La version 2.0 de Postfix a introduit la notion de classes d'adresses. Il s'agit d'une possibilité de grouper les adresses de destination par leur méthode de livraison, qui provient d'une discussion avec Victor Duchovni. Bien que les classes d'adresse présentent quelques incompatibilités, elles améliorent la manipulation des domaines hébergés et des destinataires inconnus.

Ce document fournit des informations sur les sujets suivants :

- A quoi servent les classes d'adresses ?
- Quelles classes d'adresses Postfix implémente-t-il ?
- Améliorations par rapport à Postfix 1.1
- Incompatibilités with Postfix 1.1

A quoi servent les classes d'adresses ?

En quoi devez-vous vous intéresser aux classes d'adresses ? C'est avec elles que Postfix décide quels messages il doit accepter et comment les livrer. En d'autres mots, les classes d'adresses sont très importantes pour les opérations de Postfix.

Une classe d'adresse est définie par trois champs.

- La liste des domaines membres de la classe : par exemple, tous les local domains et relay domains.
- La méthode de livraison par défaut. Par exemple, l'agent de livraison local ou smtp. Cela permet de maintenir une configuration simple de Postfix.
- La liste des adresses valides pour cette classe d'adresses. Le serveur SMTP de Postfix rejette les destinataires invalides avec le message "User unknown in <nom de la classe d'adresse ici> table". Ceci contribue à éviter de stocker des messages MAILER-DAEMON non livrables.

Quelles classes d'adresses Postfix implémente-t-il ?

Initialement, la liste des classes d'adresses est implémentée en dur dans le code, mais elle est conçue pour devenir extensible. Le résumé ci-dessous décrit le but principal de chaque classe et les paramètres de configuration *ad hoc*.

La classe des domaines locaux.

- Cible : destination finale des comptes du système UNIX et des alias style Sendmail traditionnels. Elle est typiquement utilisée pour les domaines canoniques de la machine. Pour plus d'explications sur la différence entre les domaines canoniques, les domaines hébergés et les autres domaines, reportez-vous à la page VIRTUAL READMEfile.
- Les noms de domaine sont listés dans le paramètre mydestination. Cette classe de domaine inclut également le courrier à destination de *utilisateur@[adresseIP]* lorsque l'adresse IP est listée dans les

paramètres [inet_interfaces](#) ou [proxy_interfaces](#).

- Les adresses de destination valides sont listées dans le paramètre [local_recipient_maps](#) tel que décrit à la page [LOCAL_RECIPIENT_README](#). Le serveur SMTP de Postfix rejette les destinataires invalides avec la mention "User unknown in local destinataire table". Si la valeur du paramètre [local_recipient_maps](#) est vide alors le serveur SMTP de Postfix accepte toutes les adresses de la classe [local_domain](#).
- Le transporteur du message est indiqué par le paramètre [local_transport](#). La valeur par défaut est **local:\$myhostname** pour la livraison via l'agent de livraison [local\(8\)](#).

La classe des domaines d'alias virtuels.

- Cible : destination finale des [domaines hébergés](#) où chaque adresse de destination est un alias d'un compte local du système UNIX ou d'une machine distante. Un [exemple d'alias virtuel](#) est présenté à la page [VIRTUAL_README](#).
- Les noms de domaine sont listés dans [virtual_alias_domains](#). La valeur par défaut est [\\$virtual_alias_maps](#) pour la compatibilité avec Postfix 1.1.
- Les adresses de destination valides sont listées via le paramètre [virtual_alias_maps](#). Le serveur SMTP de Postfix rejette les destinataires invalides avec la mention "User unknown in virtual alias table". La valeur par défaut est [\\$virtual_maps](#) pour la compatibilité avec Postfix 1.1.
- Il n'y a pas de paramètre de transport pour la livraison des messages, chaque adresse devant correspondre à un alias vers une autre adresse.

La classe de domaines virtual mailbox .

- Cible : destination finale pour les [domaines hébergés](#) où chaque adresse de destination peut avoir sa propre boîte-aux-lettres, et où les utilisateurs n'ont pas nécessairement un compte du système UNIX. Un [exemple de boîtes-aux-lettres virtuelles](#) est présenté à la page [VIRTUAL_README](#).
- Les noms de domaine sont listés via le paramètre [virtual_mailbox_domains](#). La valeur par défaut est [\\$virtual_mailbox_maps](#) pour la compatibilité avec Postfix 1.1.
- Les adresses de destination valides sont listées via le paramètre [virtual_mailbox_maps](#). Le serveur SMTP de Postfix rejette les destinataires invalides avec la mention "User unknown in virtual mailbox table". Si cette valeur est vide, le serveur SMTP de Postfix accepte tous les destinataires des domaines listés dans [\\$virtual_mailbox_domains](#).
- Le transport pour la livraison des messages est indiqué par le paramètre [virtual_transport](#). La valeur par défaut est **virtual** afin de livrer avec l'agent de livraison [virtual\(8\)](#).

La classe de domaines relay .

- Cible : courrier transféré à une destination distante qui liste votre système comme champ MX primaire ou secondaire. Pour plus de détails sur ce type de configuration standard, reportez-vous à la page [BASIC_CONFIGURATION_README](#). Pour les explications sur la différence entre les domaines canoniques, les [domaines hébergés](#) et les autres domaines, reportez-vous à la page [VIRTUAL_README](#).
- Les noms de domaine sont listés via le paramètre [relay_domains](#).
- Les adresses de destination valides sont listées via le paramètre [relay_recipient_maps](#). Le serveur SMTP de Postfix rejette les destinataires invalides avec la mention "User unknown in relay destinataire table". Si la valeur de ce paramètre est vide, le serveur SMTP de Postfix accepte tous les destinataires des domaines listés via le paramètre [relay_domains](#).
- Le transport pour la livraison du courrier est indiqué par le paramètre [relay_transport](#). La valeur par défaut est **relay** qui est un clone de l'agent de livraison [smtp\(8\)](#).

La classe de domaines par défaut.

- Cible : messages transféré vers Internet provenant de clients autorisés. Pour plus de détails sur ce type de configuration standard, reportez-vous à la page [BASIC CONFIGURATION README](#). Pour les explications sur la différence entre les domaines canoniques, les [domaines hébergés](#) et les autres domaines, reportez-vous à la page [VIRTUAL README](#).
- Cette classe n'a pas de table de domaines de destination.
- Cette classe n'a pas de table des adresses valides.
- Le transport pour la livraison des messages est indiqué via le paramètre [default_transport](#). La valeur par défaut est **smtp** pour livrer avec l'agent de livraison [smtp\(8\)](#).

Améliorations par rapport à Postfix 1.1

Les classes d'adresse de Postfix 2.0 ont amené les améliorations suivantes aux versions antérieures :

- Vous n'avez plus à spécifier tous les domaines [virtuels](#) de boîtes-aux-lettres dans la table de transport. L'agent de livraison [virtual\(8\)](#) est devenu un citoyen de première classe comme [local\(8\)](#) ou [smtp\(8\)](#).
- Sur des passerelles de messagerie, les classes d'adresses fournissent une séparation entre le trafic de courrier entrant ([\\$relay_transport](#)) et le trafic sortant ([\\$default_transport](#)). Ceci élimine le problème où la livraison du courrier entrant pouvait être gênée en présence d'un volume important de courrier sortant.
- Le serveur SMTP rejette les destinataires inconnus d'une manière plus consistante qu'avec Postfix version 1. C'est particulièrement utile pour éviter de garder le courrier non-livrabable (et les renvois) en file d'attente. Ce comportement est contrôlé par le paramètre [smtpd_reject_unlisted_recipient](#).
- Depuis la version 2.1 de Postfix, le serveur SMTP rejette également les adresses d'expédition inconnues (c'est à dire les adresses qu'il rejetterait comme destinataires inconnus). Le "filtrage de sortie" des expéditeurs peut aider à ralentir les explosions de vers. Ce comportement est contrôlé par le paramètre [smtpd_reject_unlisted_sender](#).

Incompatibilités avec Postfix 1.1

Les classes d'adresses Postfix 2.0 introduisent quelques changements incompatibles documentés ci-dessous. Pour faciliter les transitions, de nouveaux paramètres ont des valeurs par défaut rétro-compatibles.

- Le paramètre [virtual_maps](#) est remplacé par [virtual_alias_maps](#) (pour les recherches d'adresses) et par [virtual_alias_domains](#) (pour les noms de domaines qui étaient formellement nommés "domaines virtuels style Postfix").

Pour la compatibilité avec Postfix version 1.1, le nouveau paramètre [virtual_alias_maps](#) vaut par défaut [\\$virtual_maps](#), et le nouveau paramètre [virtual_alias_domains](#) vaut par défaut [\\$virtual_alias_maps](#).

- Le paramètre [virtual_mailbox_maps](#) a désormais un paramètre compagnon nommé [virtual_mailbox_domains](#) (pour les noms de domaines servis par l'agent de livraison virtual). Le paramètre [virtual_mailbox_maps](#) n'est maintenant utilisé que pour les recherches d'adresses.

Pour la compatibilité avec Postfix version 1.1, le nouveau paramètre [virtual_mailbox_domains](#) vaut par défaut [\\$virtual_mailbox_maps](#).

- Introduction du paramètre [relay_recipient_maps](#). Le serveur SMTP de Postfix peut l'utiliser pour bloquer le courrier des destinataires relayés qui n'existent pas. Cette liste est vide par défaut, ce qui

signifie que tous les destinataires sont acceptés.

- La fonctionnalité local_recipient_maps est maintenant activée par défaut. Le serveur SMTP de Postfix l'utilise pour rejeter le courrier des destinataires locaux inconnus. Lisez la page LOCAL_RECIPIENT_README pour plus de détails.
- Introduction du transporteur de messages "relay" dans le fichier master.cf. Ceci évite les problèmes d'ordonnancement de livraison des messages entrant lorsqu'il y a beaucoup de courrier sortant, mais peut requérir que vous mettiez à jour votre paramètre "defer_transports".

Postfix Connection Cache

Introduction

This document describes the Postfix connection cache implementation, which is available with Postfix version 2.2 and later.

Topics covered in this document:

- [What SMTP connection caching can do for you](#)
- [Connection cache implementation](#)
- [Connection cache configuration](#)
- [Connection cache safety mechanisms](#)
- [Connection cache limitations](#)
- [Connection cache statistics](#)

What SMTP connection caching can do for you

With SMTP connection caching, Postfix can deliver multiple messages over the same SMTP connection. By default, Postfix 2.2 reuses an SMTP connection automatically when a destination has high volume of mail in the [active queue](#).

SMTP Connection caching is a performance feature. Whether or not it actually improves performance depends on the conditions:

- SMTP Connection caching can greatly improve performance when delivering mail to a destination with multiple mail servers, because it can help Postfix to skip over a non-responding server.
- Otherwise, the benefits of SMTP connection caching are minor: it eliminates the latency of the TCP handshake (SYN, SYN+ACK, ACK), plus the latency of the SMTP initial handshake (220 greeting, EHLO command, EHLO response).
- SMTP Connection caching gives no gains with respect to SMTP session tear-down. The Postfix [smtp\(8\)](#) client normally does not wait for the server's reply to the QUIT command, and it never waits for the TCP final handshake to complete.
- SMTP Connection caching introduces some overhead: the client needs to send an RSET command to find out if a connection is still usable, before it can send the next MAIL FROM command.

For other potential issues with SMTP connection caching, see the discussion of [limitations](#) at the end of this document.

Connection cache implementation

For an overview of how Postfix delivers mail, see the Postfix architecture [OVERVIEW](#) document.

The Postfix connection cache is shared among Postfix mail delivering processes. This maximizes the opportunity to reuse an open connection. Other MTAs such as Sendmail or exim have a non-shared

connection cache. Here, a connection can be reused only by the mail delivering process that creates the connection. To get the same performance improvement as with a shared connection cache, non-shared connections need to be kept open for a longer time.



The scache(8) server, introduced with Postfix version 2.2, maintains the shared connection cache. With Postfix version 2.2, only the smtp(8) client has support to access this cache.

When SMTP connection caching is enabled (see next section), the smtp(8) client does not disconnect after a mail transaction, but gives the connection to the scache(8) server which keeps the connection open for a limited amount of time.

After handing over the open connection to the scache(8) server, the smtp(8) client continues with some other mail delivery request. Meanwhile, any smtp(8) client process can ask the scache(8) server for that cached connection and reuse it for mail delivery.

The connection cache can be searched by destination domain name (the right-hand side of the recipient address) and by the IP address of the host at the other end of the connection. This allows Postfix to reuse a connection even when the remote host is mail server for domains with different names.

Connection cache configuration

The Postfix smtp(8) client supports two connection caching strategies:

- On-demand connection caching. This is enabled by default, and is controlled with the smtp_connection_cache_on_demand configuration parameter. When this feature is enabled, the Postfix smtp(8) client automatically saves a connection to the connection cache when a destination has a high volume of mail in the active queue.

Example:

```

/etc/postfix/main.cf:
    smtp_connection_cache_on_demand = yes
  
```

- Per-destination connection caching. This is enabled by explicitly listing specific destinations with the smtp_connection_cache_destinations configuration parameter. After completing delivery to a selected destination, the Postfix smtp(8) client *always* saves the connection to the connection cache.

Specify a comma or white space separated list of destinations or pseudo-destinations:

- ♦ if mail is sent without a relay host: a domain name (the right-hand side of an email address, without the [] around a numeric IP address),
- ♦ if mail is sent via a relay host: a relay host name (without the [] or non-default TCP port), as specified in main.cf or in the transport map,
- ♦ a /file/name with domain names and/or relay host names as defined above,
- ♦ a "type:table" with domain names and/or relay host names on the left-hand side. The right-hand side result from "type:table" lookups is ignored.

Examples:

```
/etc/postfix/main.cf:
smtp_connection_cache_destinations = $relayhost
smtp_connection_cache_destinations = hotmail.com, ...
smtp_connection_cache_destinations = static:all (not recommended)
```

Connection cache safety mechanisms

Connection caching must be used wisely. It is anti-social to keep an unused SMTP connection open for a significant amount of time, and it is unwise to send huge numbers of messages through the same connection. In order to avoid problems with SMTP connection caching, Postfix implements the following safety mechanisms:

- The Postfix scache(8) server keeps a connection open for only a limited time. The time limit is specified with the smtp_connection_cache_time_limit and with the connection_cache_ttl_limit configuration parameters. This prevents anti-social behavior.
- The Postfix smtp(8) client reuses a session for only a limited number of times. This avoids triggering bugs in implementations that do not correctly handle multiple deliveries per session.

With Postfix 2.2 the use count is limited with the smtp_connection_cache_reuse_limit configuration parameter. With Postfix 2.3 this is replaced by a time limit which is specified with the smtp_connection_reuse_time_limit parameter. In addition, Postfix 2.3 logs the use count of multiply-used connections, as shown in the following example:

```
Nov  3 16:04:31 myname postfix/smtp[30840]: 19B6B2900FE:
to=<wietse@test.example.com>, orig_to=<wietse@test>,
relay=mail.example.com[1.2.3.4], conn_use=2, delay=0.22,
delays=0.04/0.01/0.05/0.1, dsn=2.0.0, status=sent (250 2.0.0 Ok)
```

- The connection cache explicitly labels each cached connection with destination domain and IP address information. A connection cache lookup succeeds only when the correct information is specified. This prevents mis-delivery of mail.

Connection cache limitations

Postfix SMTP connection caching conflicts with certain applications:

- The Postfix shared connection cache cannot be used with TLS, because saved TLS session information can be used only when a new connection is created (this limitation does not exist in connection caching implementations that reuse a connection only in the process that creates it). For this reason, the Postfix smtp(8) client always closes the connection after completing an attempt to deliver mail over TLS.
- Postfix connection caching currently does not support multiple SASL accounts per mail server. Specifically, Postfix connection caching assumes that a SASL credential is valid for all hostnames or domain names that deliver via the same mail server IP address and TCP port, and assume that the SASL credential does not depend on the message originator.

Connection cache statistics

The scache(8) connection cache server logs statistics about the peak cache size and the cache hit rates. This information is logged every connection_cache_status_update_time seconds, when the process terminates after the maximal idle time is exceeded, or when Postfix is reloaded.

Documentation de Postfix en français

- Hit rates for connection cache lookups by domain will tell you how useful connection caching is.
- Connection cache lookups by network address will always fail, unless you're sending mail to different domains that share the same MX hosts.
- No statistics are logged when no attempts are made to access the connection cache.

Support DSN de Postfix

Introduction

La version 2.3 de Postfix a introduit le support des notifications de statut de livraison (Delivery Status Notifications : DSN) tel que décrit dans la [RFC 3464](#). Elles donnent à l'expéditeur les informations sur le succès ou l'échec de la livraison.

Plus précisément, le support DSN donne à l'expéditeur du message la capacité à indiquer :

- les notifications envoyées : succès, échec, retard, ou rien. Normalement, Postfix n'informe l'expéditeur que lorsque la livraison est retardée ou lorsqu'elle échoue.
- le contenu retourné en cas d'échec : seul les en-têtes du message ou le message complet.
- un identificateur renvoyé avec les notifications de statut de livraison. Ceci identifie la transaction de *soumission* du message, et ne doit pas être confondu avec l'identifiant du message qui identifie le *contenu* du message.

L'implémentation du support DSN implique l'ajout de quelques paramètres aux commandes SMTP MAIL FROM et RCPT TO, ainsi que quelques nouvelles options de ligne de commande pour la commande sendmail de Postfix qui fournissent un sous-ensemble de fonctions correspondant aux paramètres des commandes SMTP.

Ce document aborde les sujets suivants :

- [Restreindre le champ des notifications de "succès"](#)
- [Interface de la ligne de commande sendmail de Postfix](#)
- [Compatibilité avec le support VERP de Postfix](#)

Restreindre le champ des notifications de "succès"

Comme les rapports de message non livrable, les rapports DSN de *succès* de livraison peuvent fournir plus d'information sur l'infrastructure interne que souhaité. Malheureusement, désactiver les requêtes de notification de "succès" requiert la désactivation des autres requêtes DSN. Les RFC n'offrent pas la possibilité de négocier des sous-ensembles de fonctionnalités.

Ce n'est pas si négatif qu'il n'y parait. Lorsque vous arrêtez le DSN pour le courrier extérieur entrant, les expéditeurs extérieurs supportant le DSN seront tout de même informés que leurs messages ont atteint votre passerelle Postfix ; ils ne leur manquera que les notifications de succès venant de votre système interne. Les expéditeurs extérieurs perdent peu : ils ne peuvent pas indiquer comment Postfix doit rapporter les échecs et retards de livraison.

Utilisez la fonctionnalité [smtpd_discard_ehlo_keyword_address_maps](#) si vous souhaitez autoriser les requêtes DSN pour les clients agréés mais pas pour les autres étrangers (voir plus bas comment les désactiver pour tous) :

```
/etc/postfix/main.cf :  
    smtpd_discard_ehlo_keyword_address_maps =  
        cidr:/etc/postfix/esmtp_access  
  
/etc/postfix/esmtp_access :  
    # N'autorise les requêtes DSN que pour le réseau local  
    192.168.0.0/28      silent-discard  
    0.0.0.0/0          silent-discard, dsn  
    ::/0               silent-discard, dsn
```

Si vous voulez désactiver l'usage des requêtes DSN pour le réseau, utilisez la fonctionnalité smtpd_discard_ehlo_keywords :

```
/etc/postfix/main.cf:  
    smtpd_discard_ehlo_keywords = silent-discard, dsn
```

Interface de la ligne de commande sendmail de Postfix

Postfix propose deux options de ligne de commande compatible Sendmail pour le support DSN.

- La première option indique quelles notifications sont envoyées pour les messages soumis via la commande en ligne sendmail(1) de Postfix :

```
$ sendmail -N success,delay,failure ... (une ou plusieurs de celles-ci)  
$ sendmail -N never ...                (ou juste celle-ci)
```

La valeur par défaut correspond à "delay,failure".

- La seconde option indique un identifiant d'enveloppe qui doit être rapporté dans les notifications de statut de livraison pour les messages soumis par la commande sendmail(1) de Postfix :

```
$ sendmail -V envelope-id ...
```

Note : ce conflit avec le support VERP dans les versions antérieures de Postfix, est présenté dans le paragraphe suivant.

Compatibilité avec le support VERP de Postfix

Avec les versions de Postfix antérieures à la 2.3, la commande sendmail(1) utilise l'option `-V` pour requérir des livraisons de type VERP. Pour requérir des livraisons de type VERP avec les versions 2.3 et supérieures de Postfix, vous devez utiliser `-XV` au lieu de `-V`.

Toutefois, la commande sendmail(1) de Postfix 2.3 identifie les tentatives de livraisons type VERP soumises avec `-V`. Elle exécutera bien la livraison VERP et vous avertira de la nouvelle syntaxe.

Guidelines for Package Builders

Purpose of this document

This document has hints and tips for those who manage their own Postfix distribution for internal use, and for those who maintain Postfix distributions for general use.

General distributions: please provide a small default main.cf file

The installed main.cf file must be small. PLEASE resist the temptation to list all 400+ parameters in the main.cf file. Postfix is supposed to be easy to configure. Listing all 400+ in main.cf defeats the purpose. It is an invitation for hobbyists to make random changes without understanding what they do, and gets them into endless trouble.

General distributions: please include README or HTML files

Please provide the applicable README or HTML files. They are referenced by the Postfix manual pages and by other files. Without README or HTML files, Postfix will be difficult if not impossible to configure.

Postfix Installation parameters

Postfix installation is controlled by a dozen installation parameters. See the postfix-install and post-install files for details. Most parameters have system-dependent default settings that are configurable at compile time, as described in the INSTALL file.

Preparing a pre-built package for distribution to other systems

You can build a Postfix package on a machine that does not have Postfix installed on it. All you need is Postfix source code and a compilation environment that is compatible with the target system.

You can build a pre-built Postfix package as an unprivileged user.

First compile Postfix. After successful compilation, execute:

```
% make package
```

With Postfix versions before 2.2 you must invoke the post-install script directly (% **sh post-install**).

You will be prompted for installation parameters. Specify an `install_root` directory other than `/`. The `mail_owner` and `setgid_group` installation parameter settings will be recorded in the `main.cf` file, but they won't take effect until the package is unpacked and installed on the destination machine.

If you want to fully automate this process, specify all the non-default installation parameters on the command line:

```
% make non-interactive-package install_root=/some/where...
```

With Postfix versions before 2.2 you must invoke the post-install script directly (`% sh post-install-non-interactive install_root...`).

Begin Security Alert

When building an archive for distribution, be sure to archive only files and symbolic links, not their parent directories. Otherwise, unpacking a pre-built Postfix package may mess up permission and/or ownership of system directories such as `/etc` `/usr` `/usr/bin` `/var` `/var/spool` and so on. This is especially an issue if you executed `postfix-install` (see above) as an unprivileged user.

End Security Alert

Thus, to tar up the pre-built package, take the following steps:

```
% cd INSTALL_ROOT
% rm -f SOMEWHERE/outputfile
% find . \! -type d -print | xargs tar cf SOMEWHERE/outputfile
% gzip SOMEWHERE/outputfile
```

This way you will not include any directories that might cause trouble upon extraction.

Installing a pre-built Postfix package

- To unpack a pre-built Postfix package, execute the equivalent of:

```
# umask 022
# gzip -d <outputfile.tar.gz | (cd / ; tar xvpf -)
```

The `umask` command is necessary for getting the correct permissions on non-Postfix directories that need to be created in the process.

- Create the necessary `mail_owner` account and `setgid_group` group for exclusive use by Postfix.
- Execute the `postfix` command to set ownership and permission of Postfix files and directories, and to update Postfix configuration files. If necessary, specify any non-default settings for `mail_owner` or `setgid_group` on the `postfix` command line:

```
# postfix set-permissions upgrade-configuration \
    setgid_group=xxx mail_owner=yyy
```

With Postfix versions before 2.1 you achieve the same result by invoking the `post-install` script directly.

Ordonnancement de la file d'attente

Objectif de ce document

Ceci est le début de la documentation de l'algorithme du gestionnaire des files d'attente de Patrik Rak. Depuis longtemps, ce code était disponible sous le nom "nqmgr(8)" (nouveau gestionnaire des files d'attente), comme module optionnel. Depuis Postfix 2.1, il est le gestionnaire des files d'attente par défaut, qui est toujours appelé "qmgr(8)". L'ancien gestionnaire des files d'attente restera disponible un certain temps sous le nom "oqmgr(8)".

Pourquoi avoir remplacé l'ancien gestionnaire des files d'attente

L'ancien ordonnaceur avait des limitations sévères dues à de mauvais choix dans sa conception.

1. Selection Round-robin par destination pour le courrier délivré via le même transporteur de messages. La stratégie round-robin avait été choisie avec l'intention d'éviter à un simple site (destination) d'utiliser trop de ressources de livraison de messages. Toutefois, cette stratégie pénalise le courrier entrant sur des passerelles bi-directionnelles. Ces destinations étant sélectionnées seulement 1 fois par nombre de destinations, même s'il elles ont plus de courrier que les autres destinations, et ainsi le courrier peut prendre du retard.

Victor Duchovni a trouvé un contournement : utiliser différents transporteurs de messages, et ainsi éviter le problème. L'ordonnaceur de Patrik Rak résout ce problème en utilisant une sélection FIFO (premier entré – premier sorti).

2. Une seconde limitation de l'ancien ordonnaceur était que la livraison des messages en bloc pouvait bloquer les autres livraisons, causant de fort retards. L'ordonnaceur de Patrik Rak's autorise les messages avec peu de destinataires de dépasser les messages en bloc de manière élégante.

Comment fonctionne le gestionnaire des files d'attente

Le texte suivant a été écrit par Patrik Rak et doit être lu en parallèle de la page de manuel postconf(5) qui décrit chaque paramètre de configuration en détail.

Du point de vue de l'utilisateur, oqmgr(8) et qmgr(8) sont les mêmes, si ce n'est que le message suivant est choisit lorsque l'agent de livraison devient disponible. Vous savez déjà que oqmgr(8) utilise un round-robin par destination alors que qmgr(8) utilise un simple FIFO, à l'exception de certaines priorités magiques. La page de manuel postconf(5) documente toutes les ficelles que l'utilisateur peut utiliser pour contrôler cette préemption magique – il n'y a que les simples conditions décrites ci-dessous pour la préemption.

Comme une documentation niveau programmeur, elle a été extraite de tous les messages que nous avons échangé avec Wietse [Arg ! J'espérais que Patrik fasse le travail pour moi — Wietse] mais je pense que peu de choses manquent de ce que nous avons mentionné dans nos conversations.

Toutefois, même du point de vue du programmeur, il n'y a rien à ajouter à l'idée de l'ordonnancement des messages elle-même. Quelques éléments peuvent la faire paraître plus compliquée qu'elle n'est, l'algorithme est le même que celui perçu par les utilisateurs. En résumé, les différences entre les points de vue utilisateur et programmeur sont :

1. Simplification des termes pour les utilisateurs : l'utilisateur n'a connaissance que des messages et des destinataires. Le programme lui-même travaille avec des jobs (un message est coupé en plusieurs jobs, un par transporteur utilisé pour livrer le message) et les entrées de la file d'attente (chaque entrée peut regrouper plusieurs destinataires pour la même destination). Ensuite il y a la structure introduite par qmgr(8) qui est simplement analogue aux jobs de la structure de la file d'attente.
2. Traitement des limites de simultanéité : l'implémentation actuelle est compliquée par le fait que les messages (resp. jobs) peuvent ne pas être livrés exactement dans le même ordre en raison des limites de concurrence. Il est nécessaire d'ignorer certains jobs "bloquant" lorsque la limite de concurrence est atteinte et de les relancer lorsque les limites le permette.
3. Traitement des limites de ressources : l'implémentation actuelle est compliquée par le fait que tous les destinataires du corps peuvent ne pas être traités en même temps. Ainsi chaque message a des destinataires dans le corps et d'autres pouvant résulter de fichiers. Ceci signifie que a) l'algorithme préemptif doit travailler avec une estimation du nombre de destinataires au lieu du nombre exact, b) il y a du code supplémentaire qui nécessite de manipuler les groupes de destinataires par transporteur qui peuvent être lus dans le corps au même moment, et c) il y a du code supplémentaire qui nécessite d'être apte à lire les destinataires dans le corps en arrière-plan et qui est déclenché au moment approprié.
4. Faire les choses efficacement : toutes les choses importantes sont faites dans le temps le plus court possible (soit directement soit après pour les traitements complexes), mais choisir le job meilleur candidat pour la préemption requiert une recherche linéaire dans tous les jobs du transporteur (le pire cas théorique – la réalité est meilleure). Comme ceci est fait chaque fois que l'entrée suivante dans la file doit être livré, il semble raisonnable d'ajouter un cache qui minimise les délais. La maintenance de ce cache obscurcit les choses.

Les points 2 et 3 sont ceux qui rendent (l'apparence de) l'implémentation compliquée, mais j'espère que la compréhension de l'algorithme d'ordonnancement lui-même (qui demeure le travail réel) reste aisée.

Note du traducteur : je ne suis pas très fier de la traduction de cette page, une petite aide serait la bienvenue.

Howto XCLIENT

But de l'extension SMTP XCLIENT

La commande XCLIENT règle les problèmes suivants :

1. Tests de contrôle d'accès. Les règles d'accès au serveur SMTP sont difficiles à vérifier lorsque les décisions ne sont communiquées qu'au clients distants. Pour faciliter les tests de règles d'accès, un programme client de test client SMTP a besoin de surcharger ses informations de nom de machine, d'adresse réseau etc... perçues par le serveur SMTP pour toute la durée de la session SMTP :
2. Les logiciels clients qui téléchargent les messages d'un serveur de messagerie amont pour les injecter dans un MTA local via SMTP. Pour conserver les avantages des règles d'accès du serveur SMTP local, le logiciel client a besoin de surcharger les informations perçues par le serveur SMTP sur le nom du client, son adresse et autres informations. Ces informations peuvent typiquement être extraites de l'en-tête Received: du serveur amont.
3. Les journaux et contrôles d'accès post-filtrage. Avec des applications de filtrage de contenu type Internet->filter->MTA, le filtre peut être simplifié s'il peut déléguer les décisions concernant le relayage et autres contrôles d'accès du MTA. C'est particulièrement pratique lorsque le filtre agit comme proxy transparent pour les commandes SMTP. Ceci requiert que le filtre puisse surcharger les informations précitées.

XCLIENT Command syntax

Des exemples de conversations client-serveur sont présentés à la fin de ce document.

Dans les réponses SMTP EHLO du serveur, le mot-clé associé à cette extension est XCLIENT. Il est suivi des noms des attributs que l'implémentation XCLIENT supporte.

La commande XCLIENT peut être transmise à tout moment sauf au lieu d'une transaction de livraison de message (i.e. entre MAIL et POINT). La commande XCLIENT peut être enchaînée lorsque le serveur supporte l'enchaînement des commandes ESMTP.

La syntaxe des requêtes XCLIENT est décrite ci-dessous. Les majuscules et chaînes encadrées représentent les terminaisons, les chaînes en minuscules représentent des terminaisons meta, et SP est un espace. Bien que les noms des commandes et attributs sont montrés en majuscule, ils sont dans la réalité insensibles à la casse.

commande-xclient = XCLIENT 1*(SP nom-attribut="valeur-attribut)

nom-attribut = (NAME | ADDR | PROTO | HELO)

- L'attribut NAME indique un nom de machine client SMTP (pas une adresse client SMTP), [UNAVAILABLE] lorsque la résolution du nom de machine du client échoue en raison d'une erreur permanente, ou [TEMPUNAVAIL] lorsque l'erreur est temporaire.
- L'attribut ADDR indique l'adresse réseau numérique IPv4 du client SMTP, une adresse IPv6 préfixée par "IPV6:", ou [UNAVAILABLE] lorsque cette information n'est pas disponible. Cette information

n'est pas encadrée par [].

- L'attribut PROTO indique SMTP ou ESMTP.
- L'attribut HELO indique la valeur du paramètre SMTP HELO, ou la valeur [UNAVAILABLE] lorsque cette information n'est pas disponible.

Note 1 : les attributs syntaxiquement corrects ne peuvent dépasser 255 caractères. Le client ne doit pas envoyer de commandes XCLIENT excédant la limite de 512 caractères des commandes SMTP. Pour éviter le dépassement, le client peut envoyer l'information en plusieurs commandes XCLIENT.

Note 2: [UNAVAILABLE], [TEMPUNAVAIL] et IPV6 : peuvent être indiqués en majuscules, minuscules ou casse mixée.

Les codes de réponse XCLIENT du serveur sont les suivants :

Code	Signification
250	succès
501	syntaxe incorrecte
503	transaction de message en cours
421	impossible de procéder, déconnexion

Exemples XCLIENT

Dans le premier exemple, le client simule un message issu d'un système en passant toutes les informations de session SMTP via la commande XCLIENT. Les informations envoyées par le client sont affichées en gras.

```
220 server.exemple.com ESMTP Postfix
client EHLO.exemple.com
250-server.exemple.com
250-PIPELINING
250-SIZE 10240000
250-VERFY
250-ETRN
250-XCLIENT NAME ADDR PROTO HELO
250 8BITMIME
XCLIENT NAME=spike.porcupine.org ADDR=168.100.189.2 PROTO=ESMTP
250 Ok
XCLIENT HELO=spike.porcupine.org
250 Ok
MAIL FROM:<wietse@porcupine.org>
250 Ok
RCPT TO:<user@exemple.com>
250 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
. . .contenu du message. . .
.
250 Ok: queued as 763402AAE6
QUIT
221 Bye
```

Dans le second exemple, le client simule un message issu d'un système en passant la commande XCLIENT avant la commande EHLO ou HELO. Ceci augmente le réalisme, mais requiert que le client connaisse à l'avance les options XCLIENT que supporte le serveur.

```
220 server.exemple.com ESMTP Postfix
XCLIENT NAME=spike.porcupine.org ADDR=168.100.189.2
250 Ok
HELO spike.porcupine.org
250 server.exemple.com
MAIL FROM:<wietse@porcupine.org>
250 Ok
RCPT TO:<user@exemple.com>
250 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
. . .contenu du message. . .
.
250 Ok: queued as CF1E52AAE7
QUIT
221 Bye
```

Securité

La commande XCLIENT change les traces d'audit et/ou les permissions d'accès du client SMTP. L'utilisation de cette commande doit être restreinte aux clients autorisés. Sinon, la commande XCLIENT ne doit pas surcharger son mécanisme de contrôle d'accès.

SMTP connection caching

Les attributs XCLIENT persistent jusqu'à la fin de la session SMTP. Si une session est utilisée pour livrer du courrier de différents clients SMTP, les attributs XCLIENT ont besoin d'être réinitialisés si nécessaire avant chaque commande MAIL FROM.

Howto XFORWARD

But de l'extension SMTP XFORWARD

La commande XFORWARD règle le problème suivant :

- Journalisation après filtrage de contenu basé sur SMTP. Avec le déploiement d'applications de filtrage de contenu type Internet->MTA1->filtre->MTA2, la journalisation des informations d'identification des clients et messages change lorsque le MTA1 donne le message au filtre de contenu. Pour simplifier l'interprétation des journaux du MTA2, il serait souhaitable que le MTA1 transfère les informations d'identification du client distant et/ou du message au MTA2 au travers du filtre de contenu, ainsi les informations peuvent être enregistrées comme faisant partie de la même transaction de message.

Cette extension est implementée en tant que commande séparée et peut être utilisée pour transmettre les attributs du client ou du message successivement. Elle n'est pas implementée en passant des paramètres additionnels via la commande MAIL FROM, car faire ainsi pourrait imposer d'étendre la limite de longueur de la commande MAIL FROM de plus de 600 caractères au delà de l'espace déjà nécessaire au support des autres extensions telles AUTH.

Syntaxe de la commande XFORWARD

Un exemple de conversation client-serveur est présenté à la fin de ce document.

Dans les réponses EHLO du serveur SMTP EHLO, le mot-clef associé à cette extension est XFORWARD. Ce mot-clef est suivi des noms des attributs que l'implémentaion de XFORWARD supporte.

Le client peut envoyer la requête XFORWARD à tout moment excepté au milieu d'une transaction de livraison de message (i.e. entre MAIL et POINT). La commande peut être enchaînée lorsque le serveur supporte l'enchaînement des commandes ESMTP.

La syntaxe des requêtes XFORWARD est décrite ci-dessous. Les majuscules et chaînes encadrées représentent les terminaisons, les chaînes en minuscules représentent des terminaisons meta, et SP est un espace. Bien que les noms des commandes et attributs sont montrés en majuscule, ils sont dans la réalité insensibles à la casse.

```
xforward-command = XFORWARD 1*( SP attribute-name="attribute-value )
```

```
attribute-name = ( NAME | ADDR | PROTO | HELO | SOURCE )
```

- L'attribut NAME indique le nom de machine, ou [UNAVAILABLE] lorsque cette information n'est pas disponible. Le nom de machine peut être un nom de machine non-DNS.
- L'attribut ADDR indique l'adresse réseau, ou [UNAVAILABLE] lorsque cette information n'est pas disponible. Cette information n'est pas encadrée par []. L'adresse peut être une adresse non-IP.

- L'attribut **PROTO** indique le protocole de réception du message depuis le client extérieur. Ce peut être un protocole autre que SMTP d'au plus 64 caractères ou [UNAVAILABLE] lorsque cette information n'est pas disponible.
- L'attribut **HELO** indique the nom de machine annoncé par le client extérieur lui-même (pas nécessairement via la commande SMTP HELO), ou [UNAVAILABLE] lorsque cette information n'est pas disponible. Le nom de machine peut ne pas être un nom DNS.
- L'attribut **SOURCE** indique LOCAL lorsque le message est issu d'une source locale, REMOTE pour les autres ou [UNAVAILABLE] lorsque cette information n'est pas disponible. Le MTA aval peut décider d'activer le traitement des en-têtes et la qualification des adresses avec les messages des sources locales.

Note 1 : les valeurs des attributs ne peuvent dépasser 255 caractères (certains attributs peuvent imposer des longueurs plus courtes), ne doivent pas contenir de caractères de contrôle, non-ASCII, des espaces, ou d'autres caractères particuliers aux en-têtes de message. Les futurs attributs pourront violer cette règle en utilisant l'encodage xtext comme décrit par la [RFC 1891](#).

Note 2 : les noms de machines DNS peuvent atteindre 255 caractères au plus. L'implémentation XFORWARD cliente ne doit pas envoyer de commandes XFORWARD excédant les 512 caractères, limite des commandes SMTP.

Note 3 : [UNAVAILABLE] peut être indiqué en majuscules, minuscules ou casse mixte.

Note 4 : l'implémentation XFORWARD serveur ne doit pas mélanger les informations de la session SMTP courante avec les informations transmises depuis la session amont.

Les codes de réponse XFORWARD du serveur sont les suivants :

Code	Signification
250	succès
501	syntaxe incorrecte
503	transaction de message en cours
421	impossible de procéder, déconnexion

XFORWARD Exemple

Dans l'exemple suivant, les informations envoyées par le client sont montrées en gras.

```
220 server.exemple.com ESMTP Postfix
client EHLO.exemple.com
250-server.exemple.com
250-PIPELINING
250-SIZE 10240000
250-VERFY
250-ETRN
250-XFORWARD NAME ADDR PROTO HELO
250 8BITMIME
XFORWARD NAME=spike.porcupine.org ADDR=168.100.189.2 PROTO=ESMTP
250 Ok
XFORWARD HELO=spike.porcupine.org
250 Ok
MAIL FROM:<wietse@porcupine.org>
250 Ok
```



```
RCPT TO:<user@example.com>
250 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
. . .contenu du message. . .
.
250 Ok: queued as 3CF6B2AAE8
QUIT
221 Bye
```

Securité

La commande XFORWARD change les traces d'audit. L'utilisation de cette commande doit être restreinte aux clients autorisés.

Cache des connexions SMTP

Le cache des connexions SMTP permet de livrer de multiples messages dans la même session SMTP. Les attributs XFORWARD sont remis à zéro après chaque accomplissement d'une commande MAIL FROM, ainsi il n'y a pas de risque de mélange d'informations.

Paramètres de Configuration de

Postfix

Format du fichier main.cf de Postfix

Le fichier de configuration main.cf de Postfix renseigne un petit sous-ensemble des paramètres qui contrôlent les opérations du système de messagerie Postfix. Les paramètres non explicitement renseignés sont initialisés avec leur valeur par défaut.

Ci-dessous le format général du fichier main.cf :

- Chaque ligne logique est sous la forme "parametre = valeur". Les espaces autour du signe "=" sont ignorés, comme ceux situés à la fin de la ligne logique.
- Les lignes vides ou constituées seulement d'espaces sont ignorées ainsi que celles dont le premier caractère autre qu'un espace est `#`.
- Une ligne logique ne démarre pas avec un espace. Une ligne démarrant avec un espace continue la ligne logique.
- La valeur d'un paramètre peut faire référence à d'autres paramètres.
 - ◆ Les expressions "\$paramètre", "\${paramètre}" ou "\$(paramètre)" sont récursivement remplacées par la valeur du paramètre.
 - ◆ L'expression "\${paramètre?valeur}" est remplacée par "valeur" lorsque "\$paramètre" est non vide.
 - ◆ L'expression "\${paramètre:valeur}" est remplacée par "valeur" lorsque "\$paramètre" est vide.
- Lorsque le même paramètre est définie plusieurs fois, seule la dernière occurrence est enregistrée.
- L'ordre de définition des paramètres du fichier main.cf est sans importance.

Ce document décrit tous les paramètres de configuration de Postfix. Les valeurs par défaut sont indiquées après le nom du paramètre entre parenthèses, et peuvent être obtenues avec la commande **postconf -d**.

Note : ce n'est pas une invitation à modifier les paramètres de configuration de Postfix. Les changements non nécessaires pourraient gêner le fonctionnement du système de messagerie.

2bounce_notice_recipient (défaut : postmaster)

Le destinataire du courrier non livrable qui ne peut être retourné à l'expéditeur. Cette fonctionnalité est activée par le paramètre notify_classes.

access_map_reject_code (défaut : 554)

Le code numérique que le serveur SMTP de Postfix renvoie dans sa réponse lorsqu'un client est rejeté par une table de restriction (access(5)).

Ne changez pas ce paramètre avant d'avoir complètement compris la RFC 821.

address_verify_default_transport (défaut : \$default_transport)

Surcharge la valeur du paramètre default_transport pour la vérification des adresses.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

address_verify_local_transport (défaut : \$local_transport)

Surcharge la valeur du paramètre local_transport pour la vérification des adresses.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

address_verify_map (défaut : vide)

Table de correspondance optionnelle pour le stockage des status de vérification des adresses. Cette table est maintenue par le service verify(8) et est ouverte avant que le processus ne limite ses privilèges.

Par défaut, cette information est gardée en mémoire volatile et est perdue après "**postfix reload**" ou "**postfix stop**".

Indiquez un emplacement dans un système de fichier qui ne risque pas d'être saturé. Si la base de données est corrompue, le système ne peut plus recevoir de courrier. Pour réparer, effacez le fichier et lancez "**postfix reload**".

Exemples :

```
address_verify_map = hash:/etc/postfix/verify  
address_verify_map = btree:/etc/postfix/verify
```

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

address_verify_negative_cache (défaut : yes)

Active la mise en cache des résultats négatifs de vérification d'adresse. Lorsque cette fonctionnalité est activée, le cache peut vite être pollué avec des adresses inexistantes. Lorsque cette fonctionnalité est désactivée, Postfix génère un sondage d'adresse pour chaque consultation.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

address_verify_negative_expire_time (défaut : 3d)

Délai d'expiration du cache des résultats négatifs de vérification d'adresse.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines).

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

address_verify_negative_refresh_time (défaut : 3h)

Délai à partir duquel un résultat négatif de vérification d'adresse doit être rafraîchi.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines).

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

address_verify_poll_count (défaut : 3)

Combien de fois interroger le service verify(8) avant d'achever une vérification d'adresse en cours.

Le nombre par défaut est 3.

Indiquez 1 pour implémenter une forme brute de liste grise, c'est à dire toujours retarder la première livraison pour une adresse jamais vue.

Exemple :

```
address_verify_poll_count = 1
```

Documentation de Postfix en français

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

address_verify_poll_delay (défaut : 3s)

Le délai entre deux requêtes d'une vérification d'adresse en cours.

Le délai par défaut est de 3 secondes.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines).

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

address_verify_positive_expire_time (défaut : 31d)

Délai après lequel un résultat positif de vérification d'adresse expire du cache.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines).

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

address_verify_positive_refresh_time (défaut : 7d)

Délai après lequel un résultat positif de vérification d'adresse doit être rafraîchi. Le statut de la vérification n'est pas mis à jour lorsqu'une vérification échoue (cache optimiste).

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines).

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

address_verify_relay_transport (défaut : \$relay_transport)

Surcharge la valeur du paramètre relay_transport pour la vérification des adresses.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

address_verify_relayhost (défaut : \$relayhost)

Surcharge la valeur du paramètre relayhost pour les sondages de vérification d'adresse.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

address_verify_sender (défaut : postmaster)

L'adresse d'expédition à utiliser pour les sondages de vérification d'adresse. Pour éviter les problèmes avec les sondages qui sont envoyés en réponse à un sondage d'adresse, le serveur SMTP de Postfix exclut de la vérification l'adresse d'expédition des sondages.

Indiquez une valeur vide (address_verify_sender =) ou <> si vous voulez utiliser une adresse nulle. Attention, certains sites rejettent le courrier de <>, même si les RFCs imposent que de telles adresses soient acceptées.

Exemples :

```
address_verify_sender = <>
address_verify_sender = postmaster@my.domain
```

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

address_verify_sender_dependent_relayhost_maps (défaut : vide)

Surcharge le paramètre sender_dependent_relayhost_maps pour les sondages de vérification d'adresse.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

address_verify_service_name (défaut : verify)

Documentation de Postfix en français

Le nom du service de vérification d'adresse verify(8). Ce service maintient le statut des sondages de vérification de l'adresse d'expéditeur et/ou de destination et génère les sondages sur requête des autres processus de Postfix.

address_verify_transport_maps (défaut : \$transport_maps)

Surcharge la valeur du paramètre transport_maps pour les sondages de vérification d'adresses.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

address_verify_virtual_transport (défaut : \$virtual_transport)

Surcharge le paramètre virtual_transport pour les sondages de vérification d'adresses.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

alias_database (défaut : voir la sortie de "postconf -d")

La base de données des alias pour la livraison locale (local(8)) qui est mise à jour avec "**newaliases**" ou "**sendmail -bi**".

Il s'agit d'un paramètre de configuration distinct car les tables indiquées par \$alias_maps ne sont pas nécessairement des fichiers locaux.

Exemples :

```
alias_database = hash:/etc/aliases
alias_database = hash:/etc/mail/aliases
```

alias_maps (défaut : utilisez "postconf -d")

La base de données des alias utilisée pour la livraison locale (local(8)). Reportez-vous à la page aliases(5) pour les détails sur la syntaxe.

La liste par défaut dépend du système. Sur les systèmes NIS, la recherche se fait d'abord sur la base locale, puis sur la base NIS.

Si vous changez cette base, lancez "**postalias /etc/aliases**" (changez le nom de fichier en fonction de votre système), ou plus simplement utilisez "**newaliases**" pour construire le fichier DBM ou DB.

L'agent de livraison local(8) interdit les substitutions dans les expressions régulières type \$1 etc. dans alias_maps, car ceci ouvre un trou de sécurité.

L'agent de livraison local(8) ignorera silencieusement les requêtes utilisant le serveur proxymap(8) dans alias_maps. À la place, il ouvrira directement la table. Dans les versions de Postfix antérieures à la 2.2, l'agent de livraison local(8) se terminait avec une erreur fatale.

Exemples :

```
alias_maps = hash:/etc/aliases, nis:mail.aliases
alias_maps = hash:/etc/aliases
```

allow_mail_to_commands (défaut : alias, forward)

Restreint la livraison des messages locaux aux commandes externes. Par défaut, la livraison aux "|commandes" dans les fichiers :include: est désactivée (voir aliases(5) pour la définition de ces terminologies).

Indiquez zéro ou plus de : **alias**, **forward** or **include**, pour autoriser les commandes dans les fichiers aliases(5), .forward ou dans les fichiers :include:, respectivement.

Exemple :

```
allow_mail_to_commands = alias,forward,include
```

allow_mail_to_files (défaut : alias, forward)

Restreint la livraison de messages locaux à des fichiers externes (local(8)). La valeur par défaut est de ne pas autoriser les desinations `"/nom/de/fichier"` dans les fichiers `:include:` (reportez-vous à la page aliases(5) pour voir la définition de cette terminologie).

Indiquez zero ou plus de: **alias**, **forward** ou **include**, afin d'autoriser les destinations `"/nom/de/fichier"` dans la base aliases(5), les fichiers `.forward` et `:include:` respectivement.

Exemple :

```
allow_mail_to_files = alias,forward,include
```

allow_min_user (défaut : no)

Autorise une adresse de destination à avoir un ``-'` en premier caractère. Par défaut, ceci n'est pas autorisé, pour éviter les incidents avec les logiciels qui passent les adresses email par une ligne de commande. De tels logiciels ne pourront pas distinguer une adresse malicieuse d'une option de la ligne de commande. Toutefois, ceci peut être évité en insérant une marque de fin des options `"--"` dans la ligne de commande, ceci reste difficile à mettre en œuvre.

allow_percent_hack (défaut : yes)

Active la réécriture des formes `"user%domain"` en `"user@domain"`. Cette fonctionnalité est activée par défaut.

Note : avec les versions 2.2 et supérieures de Postfix, la réécriture des adresses dans les en-têtes de message ne se produit que lorsque l'une des conditions suivantes est réalisée :

- ◇ le message est soumis par la commande sendmail(1) de Postfix,
- ◇ le message provient d'un client réseau qui correspond à \$local_header_rewrite_clients,
- ◇ le message provient du réseau et le paramètre remote_header_rewrite_domain contient une valeur non vide.

Pour retrouver le comportement des versions de Postfix antérieures à la 2.2, utilisez "local_header_rewrite_clients = static:all".

Exemple :

```
allow_percent_hack = no
```

allow_untrusted_routing (défaut : no)

Transfert les messages avec le routage spécifié par l'expéditeur (`user[@%!]remote[@%!]site`) des clients hors réseau de confiance vers les destinations correspondant à \$relay_domains.

Par défaut, cette fonctionnalité est désactivée. Ceci évite de créer un relai ouvert depuis un MX de secours qui pourrait alors être utilisé pour envoyer du spam via le MX primaire.

Ce paramètre contrôle si des adresses non-locales avec un routage spécifié par l'expéditeur peuvent correspondre aux tables d'accès de Postfix. Par défaut, de telles adresses ne peuvent correspondre à des tables d'accès de Postfix, car ces adresses sont ambiguës.

alternate_config_directories (défaut : vide)

Une liste de répertoires de configuration autres que ceux par défaut qui peuvent être indiqués avec `"-c répertoire_config"` en ligne de commande ou via le paramètre d'environnement `MAIL_CONFIG`.

Cette liste doit être renseignée dans le répertoire de configuration par défaut de Postfix et est utilisée par les commandes `setgid` de Postfix telles `postqueue(1)` et `postdrop(1)`.

always_bcc (défaut : vide)

Adresses optionnelles qui reçoivent une copie cachée (BCC) de chaque message reçus par le système de messagerie Postfix.

NOTE : si le message à destination de l'adresse cachée est rejetée, il sera retourné à l'expéditeur.

NOTE : les copies cachées automatiques ne sont produites que pour les nouveaux messages. Pour éviter les boucles de messages, elles ne sont pas générées pour les messages que Postfix transfère en interne ni pour les messages générés par Postfix lui-même.

anvil_rate_time_unit (défaut : 60s)

Délai à partir duquel les taux de connexion client et autres taux sont calculés.

cette fonctionnalité est implementée par le service `anvil(8)` qui ne fait pas partie de la version stable 2.1 de Postfix.

L'intervalle par défaut est relativement court. A cause des fréquents changements, le serveur `anvil(8)` utilise uniquement la mémoire volatile. Ainsi, l'information est perdue lorsque le processus s'arrête.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

anvil_status_update_time (défaut : 600s)

Fréquence de vérification du status du service `anvil(8)`. *Traduction très approximative...*

cette fonctionnalité est implementée par le service `anvil(8)` qui ne fait pas partie de la version stable 2.1 de Postfix.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

append_at_myorigin (défaut : yes)

Avec le courrier soumis localement, ajoute la chaîne `.$myorigin` aux adresses sans domaines ni nom d'hôte. Avec le courrier soumis depuis l'extérieur, ajoute la chaîne `"@$remote_header_rewrite_domain"` à la place

Note 1 : cette fonctionnalité est activée par défaut et ne doit pas être désactivée. Postfix ne supporte pas les adresses sans domaines.

Note 2 : avec les versions 2.2 et supérieures de Postfix, la réécriture des adresses dans les en-têtes de message ne se produit que lorsque l'une des conditions suivantes est réalisée :

- ◇ le message est soumis par la commande `sendmail(1)` de Postfix,
- ◇ le message provient d'un client réseau qui correspond à `$local_header_rewrite_clients`,
- ◇ le message provient du réseau et le paramètre `remote_header_rewrite_domain` contient une valeur non vide.

Pour retrouver le comportement des versions de Postfix antérieures à la 2.2, utilisez `"local_header_rewrite_clients = static:all"`.

append_dot_mydomain (défaut : yes)

Avec le courrier soumis localement, ajoute la chaîne `.$mydomain` aux adresses qui n'ont pas d'information `".domaine"` (seulement le nom d'hôte). Avec le courrier soumis depuis l'extérieur, ajoute la chaîne `"@$remote_header_rewrite_domain"` à la place

Documentation de Postfix en français

Note 1 : cette fonctionnalité est activée par défaut. Si elle est désactivée, les utilisateurs ne pourront pas envoyer de messages à "utilisateur@nom-de-domaine-partiel" mais devront spécifier le nom de domaine complet.

Note 2 : avec les versions 2.2 et supérieures de Postfix, la réécriture des adresses dans les en-têtes de message ne se produit que lorsque l'une des conditions suivantes est réalisée :

- ◇ le message est soumis par la commande sendmail(1) de Postfix,
- ◇ le message provient d'un client réseau qui correspond à \$local_header_rewrite_clients,
- ◇ le message provient du réseau et le paramètre remote_header_rewrite_domain contient une valeur non vide.

Pour retrouver le comportement des versions de Postfix antérieures à la 2.2, utilisez

"local_header_rewrite_clients = static:all".

application_event_drain_time (défaut : 100s)

Combien de temps la commande postkick(1) attend qu'une requête entre dans le buffer d'entrée du serveur avant de transmettre.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

authorized_flush_users (défaut : static:anyone)

Liste des utilisateurs autorisés à vider la file d'attente.

Par défaut, tous les utilisateurs sont autorisés à vider la file d'attente. L'accès est toujours autorisé si l'utilisateur l'invoquant est le super-utilisateur ou l'utilisateur propriétaire (\$mail_owner). Autrement, l'UID réel du processus est trouvé dans le fichier système passwd et l'accès n'est autorisé que si le nom de login correspondant est dans la liste d'accès. L'utilisateur "unknown" est utilisé pour les processus dont l'UID réel n'est pas trouvé dans le fichier des mots-de-passe.

Indiquez une liste de noms d'utilisateurs, "/nom/de/fichier" ou expressions "type:table", séparé par des virgules et/ou des espaces. La liste est examinée de gauche à droite, et la recherche s'arrête dès la première occurrence correspondante. Indiquez "!name" pour exclure un nom de la liste. Un "/nom/de/fichier" de correspondances est remplacé par son contenu ; une table de correspondances "type:table" correspond lorsqu'un nom correspond à la clef de recherche (le résultat de la consultation est ignoré). Continuez les lignes longues en commençant la ligne suivante par des espaces.

cette fonctionnalité est disponible dans les versions 2.2 et supérieures de Postfix.

authorized_mailq_users (défaut : static:anyone)

Liste des utilisateurs autorisés à voir la file d'attente.

Par défaut, tous les utilisateurs sont autorisés à voir la file d'attente. L'accès est toujours autorisé si l'utilisateur l'invoquant est le super-utilisateur ou l'utilisateur propriétaire (\$mail_owner). Autrement, l'UID réel du processus est trouvé dans le fichier système passwd et l'accès n'est autorisé que si le nom de login correspondant est dans la liste d'accès. L'utilisateur "unknown" est utilisé pour les processus dont l'UID réel n'est pas trouvé dans le fichier des mots-de-passe.

Indiquez une liste de noms d'utilisateurs, "/nom/de/fichier" ou expressions "type:table", séparé par des virgules et/ou des espaces. La liste est examinée de gauche à droite, et la recherche s'arrête dès la première occurrence correspondante. Indiquez "!name" pour exclure un nom de la liste. Un "/nom/de/fichier" de correspondances est remplacé par son contenu ; une table de correspondances

"type:table" correspond lorsqu'un nom correspond à la clef de recherche (le résultat de la consultation est ignoré). Continuez les lignes longues en commençant la ligne suivante par des espaces.

cette fonctionnalité est disponible dans les versions 2.2 et supérieures de Postfix.

authorized_submit_users (défaut : static:anyone)

Liste des utilisateurs autorisés à soumettre des messages avec la commande sendmail(1) (et avec la commande d'aide privilégiée postdrop(1)).

Par défaut, tous les utilisateurs sont autorisés à soumettre des messages. Autrement, l'UID réel du processus est trouvé dans le fichier système passwd et l'accès n'est autorisé que si le nom de login correspondant est dans la liste d'accès. L'utilisateur "unknown" est utilisé pour les processus dont l'UID réel n'est pas trouvé dans le fichier des mots-de-passe.

Indiquez une liste de noms d'utilisateurs, "/nom/de/fichier" ou expressions "type:table", séparé par des virgules et/ou des espaces. La liste est examinée de gauche à droite, et la recherche s'arrête dès la première occurrence correspondante. Indiquez "!name" pour exclure un nom de la liste. Un "/nom/de/fichier" de correspondances est remplacé par son contenu ; une table de correspondances "type:table" correspond lorsqu'un nom correspond à la clef de recherche (le résultat de la consultation est ignoré). Continuez les lignes longues en commençant la ligne suivante par des espaces.

Exemple :

```
authorized_submit_users = !www, static:all
```

cette fonctionnalité est disponible dans les versions 2.2 et supérieures de Postfix.

authorized_verp_clients (défaut : \$mynetworks)

Quels clients sont autorisés à utiliser la commande XVERP. Cette commande impose que le message ne soit fourni qu'avec un destinataire à la fois avec une adresse de retour par destinataire.

Par défaut, seuls les clients de confiance sont autorisés à l'utiliser.

Ce paramètre a été introduit dans la version 1.1 de Postfix. Postfix version 2.1 a renommé ce paramètre en smtpd_authorized_verp_clients et changé la valeur par défaut en aucun.

Indiquez une liste d'expressions réseau/masque séparées par des virgules et/ou des espaces. Le masque indique le nombre de bits de la partie réseau de l'adresse IP. Vous pouvez également indiquer des noms de machines ou de .domaines (remarquez le '.' initial), des "/nom/de/fichier" ou expressions "type:table". Une expression "/nom/de/fichier" est remplacée par son contenu; lorsqu'une entrée est trouvée dans une table de correspondance "type:table" elle est autorisée (le résultat de la consultation est ignoré).

Note : les adresses IP version 6 doivent être indiquées entre [] dans authorized_verp_clients et dans les fichier indiqués avec "/nom/de/fichier". Les adresses IP version 6 contiennent le caractère ":" et pourraient être confondues avec une expression "type:table".

backwards_bounce_logfile_compatibility (défaut : yes)

Produit des logs supplémentaires bounce(8) qui peuvent être lus par des versions anciennes de Postfix. L'actuel format "name = value", plus extensible, est requis pour implémenter des fonctionnalités plus sophistiquées.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

berkeley_db_create_buffer_size (défaut : 16777216)

Documentation de Postfix en français

La taille du buffer d'entrées/sorties par table pour les programmes qui créent des tables hash ou btree Berkeley DB. Indiquez un nombre d'octet.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

berkeley_db_read_buffer_size (défaut : 131072)

La taille du buffer d'entrées/sorties par table pour les programmes qui lisent des tables hash ou btree Berkeley DB. Indiquez un nombre d'octet.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

best_mx_transport (défaut : vide)

Où le client SMTP de Postfix doit livrer le courrier lorsqu'il détecte une boucle de message vers lui-même. Ceci arrive lorsque le MTA local est le meilleur échangeur SMTP pour une destination non listée dans \$mydestination, \$inet_interfaces, \$proxy_interfaces, \$virtual_alias_domains, ou \$virtual_mailbox_domains. Par défaut, le client SMTP de Postfix retourne ces courriers comme non-livrables.

Indiquez par exemple, "best_mx_transport = local" pour passer le message du client client SMTP à l'agent de livraison local(8). Vous pouvez indiquer tout les "transport" or "transport:nexthop" définis dans le fichier master.cf. Reportez-vous à la page de manuel transport(5) pour la syntaxe et la signification de "transport" ou "transport:nexthop".

Toutefois, cette fonctionnalité est couteuse car elle maintient le processus client SMTP de Postfix jusqu'à ce que l'agent de livraison local(8) ait fini son travail. Il est plus efficace (pour Postfix) de lister tous les domaines hébergés dans une table ou une base de données.

biff (défaut : yes)

Utiliser ou non le service local biff. Ce service envoie des notifications "nouveau message" aux utilisateurs qui ont demandé cette fonctionnalité avec la commande UNIX "biff y".

Pour des raisons de compatibilité, cette fonctionnalité est activée par défaut. Sur des systèmes avec un grand nombre d'utilisateurs interactifs, le service biff peut pénaliser les performances. Indiquez "biff = no" pour le désactiver.

body_checks (défaut : vide)

Tables de correspondances optionnelles pour l'inspection du contenu tel qu'indiqué à la page de manuel body_checks(5).

Note : avec les versions de Postfix antérieures à la version 2.0, ces règles inspectent tout le contenu après les en-têtes primaires de message.

body_checks_size_limit (défaut : 51200)

Taille d'un segment du corps du message soumis à l'inspection body_checks. La quantité de texte est limitée pour éviter de scanner les attachements.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

bounce_notice_recipient (défaut : postmaster)

Le destinataire des notifications au postmaster contenant les en-têtes de message des messages que Postfix n'a pas livré et la retranscription des conversations SMTP que Postfix a rejetées. Cette fonctionnalité est activée avec le paramètre notify_classes.

bounce_queue_lifetime (défaut : 5d)

Le temps maximum avant qu'un message en file d'attente soit considéré comme non-livrabable. Par défaut, c'est le même temps que la durée maximale de mise en file d'attente des messages réguliers.

Documentation de Postfix en français

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est d (jours).

Indiquez 0 lorsque la livraison ne doit être testée qu'une fois.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

bounce_service_name (défaut : bounce)

Le nom du service bounce(8). Ce service maintient un enregistrement des tentatives de livraison échouées et génère les notifications de non-livraison.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

bounce_size_limit (défaut : 50000)

Le volume maximum du message original envoyé dans une notification de non-livraison. Indiquez un nombre d'octets. Si vous augmentez cette limite, vous devez également augmenter la valeur mime_nesting_limit en proportion.

bounce_template_file (défaut : vide)

Chemin d'un fichier de configuration contenant des modèles de messages renvoyés. Ceci surcharge les modèles intégrés des messages de notification de statut de livraison (DSN) pour les messages non livrables, retardés ou bien livrés, ou des vérifications de livraison. La page de manuel bounce(5) décrit comment éditer et tester les fichiers de modèles.

Le texte du corps d'un message modèle devrait contenir des références \$nom aux paramètres de configuration de Postfix. Le résultat d'une substitution de \$nom peut être visualisé avec "**postconf -b nom_de_fichier**" avant que le fichier ne soit placé dans le répertoire de configuration de Postfix.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

broken_sasl_auth_clients (défaut : no)

Active l'interopérabilité avec les clients SMTP qui implémentent une version obsolète de la commande AUTH (RFC 2554). MicroSoft Outlook Express version 4 et MicroSoft Exchange version 5.0 sont des exemples de tels clients.

Indiquez "broken_sasl_auth_clients = yes" pour que Postfix indique le support de AUTH de manière non-standard.

canonical_classes (défaut : envelope_sender, envelope_recipient, header_sender, header_recipient)

Adresses sujettes aux substitutions d'adresses canonical_maps. Par défaut, les réécritures canonical_maps sont appliquées aux adresses d'expéditeur et de destination de l'enveloppe, et aux adresses d'expéditeur et de destination des en-têtes.

Indiquez une ou plusieurs des expressions : envelope_sender, envelope_recipient, header_sender, header_recipient

Cette fonctionnalité est disponible dans les versions 2.2 et supérieures de Postfix.

canonical_maps (défaut : vide)

Tables optionnelle de correspondance des adresses pour les en-têtes de message et les enveloppes. Les correspondances sont appliquées sur les adresses de destination et d'expédition, dans l'enveloppe et dans les en-têtes. C' est typiquement utilisé pour nettoyer les adresses sales des systèmes de messagerie, ou pour remplacer les noms de login par prenom.nom. Les correspondances et le format de la table sont étudiés à la page canonical(5). Pour découvrir les manipulations d'adresses de Postfix, lisez ADDRESS REWRITING README.

Documentation de Postfix en français

Si vous utilisez cette fonctionnalité, lancez "**postmap /etc/postfix/canonical**" pour construire le nécessaire fichier DBM ou DB après chaque modification. Les changements seront visible après une minute. Utilisez "**postfix reload**" pour éliminer ce délai.

Note : avec les versions 2.2 et supérieures de Postfix, la réécriture des adresses dans les en-têtes de message ne se produit que lorsque l'une des conditions suivantes est réalisée :

- ◇ le message est soumis par la commande sendmail(1) de Postfix,
- ◇ le message provient d'un client réseau qui correspond à \$local_header_rewrite_clients,
- ◇ le message provient du réseau et le paramètre remote_header_rewrite_domain contient une valeur non vide.

Pour retrouver le comportement des versions de Postfix antérieures à la 2.2, utilisez "local_header_rewrite_clients = static:all".

Exemples :

```
canonical_maps = dbm:/etc/postfix/canonical  
canonical_maps = hash:/etc/postfix/canonical
```

cleanup_milters (défaut : vide)

Une liste d'applications Milters (*mail filter*) pour les messages qui n'arrivent pas via le serveur smtpd(8). Voir MILTER_README pour plus de détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

cleanup_service_name (défaut : cleanup)

Le nom du service cleanup(8). Ce service réécrit les adresses dans la forme standard et procède aux correspondances d'adresse canoniques et aux alias virtuels.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

command_directory (défaut : voir "postconf -d" output)

L'emplacement des commandes administratives de Postfix.

command_execution_directory (défaut : vide)

Répertoire de travail de l'agent de livraison local(8) pour la livraison à une commande externe. Une erreur lors du changement de répertoire retarde la livraison.

Les substitutions suivantes \$name sont effectuées sur command_execution_directory avant le changement de répertoire. Les substitutions s'effectuent dans le contexte de requête de livraison. Le résultat de la substitution \$name est filtré avec les caractères indiqués au paramètre execution_directory_expansion_filter.

\$user

Le nom du destinataire.

\$shell

Le nom du shell du login du destinataire.

\$home

Le répertoire personnel du destinataire.

\$recipient

L'adresse complète de destination.

\$extension

L'extension optionnelle de l'adresse de destination.

\$domain

Le domaine du destinataire.

\$local

La partie locale entière du destinataire.

\$recipient_delimiter

Le délimiteur système de l'extension de adresse de destination.

\${name?value}

Substitue *value* à *\$name* si *\$name* est non vide.

\${name:value}

Substitue *value* à *\$name* si *\$name* est vide.

Au lieu de *\$name*, vous pouvez utiliser *\$(name)* ou *\${name}*.

cette fonctionnalité est disponible dans les versions 2.2 et supérieures de Postfix.

***command_expansion_filter* (défaut : voir la sortie de "postconf -d")**

Restreint les caractères que l'agent de livraison local(8) autorise dans les expansions de *\$name* de \$mailbox_command. Les caractères ne correspondant pas sont remplacés par des underscores (_).

***command_time_limit* (défaut : 1000s)**

La limite de toute commande externe. Elle est utilisée par l'agent de livraison local(8), et constitue le temps limite par défaut de l'agent de livraison pipe(8).

Note : si vous spécifiez une valeur élevée, vous devez également modifier le paramètre global ipc_timeout.

***config_directory* (défaut : voir la sortie de "postconf -d")**

L'emplacement par défaut des fichiers de configuration main.cf and master.cf. Cette valeur peut être surchargée par les mécanismes suivants :

◇ la variable d'environnement MAIL_CONFIG (processus démons et commandes).

◇ l'option "-c" de la ligne de commande (commandes uniquement).

Avec les commandes qui fonctionnent avec les privilèges set-gid, une surcharge de config_directory nécessite soit les privilèges root, soit requiert que le répertoire soit listé dans le paramètre alternate_config_directories du fichier main.cf par défaut.

***connection_cache_protocol_timeout* (défaut : 5s)**

Temps limite pour les opérations de connexion, envoi et réception du cache de connexions. Ce délai est imposé au client.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

***connection_cache_service* (défaut : scache)**

Nom du service de cache des connexions scache(8). Ce service maintient un nombre limité de sessions en cache.

***connection_cache_status_update_time* (défaut : 600s)**

Fréquence à laquelle le serveur scache(8) enregistre les statistiques d'emploi avec les taux de présence et d'absence des connexions dans le cache pour les destinations logiques et physiques.

***connection_cache_ttl_limit* (défaut : 2s)**

Délai maximum de maintien d'une connexion que le serveur de cache des connexions scache(8) accepte. Les requêtes qui demandent une durée plus longue seront stockées avec ce temps limite autorisé. Le but de ce contrôle additionnel est de protéger l'infrastructure contre les personnes insouciantes. Ce délai est lié à \$max_idle.

***content_filter* (défaut : vide)**

Le nom d'un transporteur du courrier qui filtre les messages après mise en file d'attente.

Ce paramètre utilise la même syntaxe coté droit qu'une table transport(5) de Postfix. Ce paramètre a une préférence moins élevée qu'un filtre de contenu renvoyé par une table access(5), header_checks(5) ou body_checks(5).

daemon_directory (défaut : voir la sortie de "postconf -d")

Le répertoire contenant les programmes support et les démons de Postfix. Il ne doit pas être invoqué directement par des utilisateurs et le répertoire doit appartenir à root.

daemon_timeout (défaut : 18000s)

Temps maximum pour qu'un démon achève une requête avant d'être interrompu par le watchdog de Postfix.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

debug_peer_level (défaut : 2)

La valeur d'incrément du niveau de verbiage des logs lorsqu'un client ou un serveur distant correspond à l'expression du paramètre debug_peer_list.

debug_peer_list (défaut : vide)

Liste optionnelle de clients ou serveurs distants qui déclenche l'élévation du niveau de débogage de la valeur indiquée par \$debug_peer_level.

Indiquez des expressions noms de domaine ou adresses-IP/masque, des "/noms/de/fichiers" ou des tables de correspondance "type:table". Le résultat de la consultation est ignoré.

Les expressions correspondant à des noms de domaine sont contrôlées par le paramètre parent_domain_matches_subdomains.

Exemples :

```
debug_peer_list = 127.0.0.1  
debug_peer_list = some.domain
```

debugger_command (défaut : vide)

La commande externe à exécuter lorsqu'un programme démon de Postfix est invoqué avec l'option -D.

Utilisez "commande .. & sleep 5" ainsi le debugger peut d'attacher avant que le processus ne démarre. Si vous utilisez un debugger basé sur X-window, assurez-vous d'avoir renseigné votre variable d'environnement XAUTHORITY avant de démarrer Postfix.

Exemple :

```
debugger_command =  
PATH=/usr/bin:/usr/X11R6/bin  
xxgdb $daemon_directory/$process_name $process_id & sleep 5
```

default_database_type (défaut : voir la sortie de "postconf -d")

La base de donnée par défaut à utiliser avec les commandes newaliases(1), postalias(1) et postmap(1). Sur beaucoup de systèmes UNIX le type par défaut est **dbm** or **hash**. Ce paramètre par défaut est gelé à la compilation du système Postfix.

Exemples :

```
default_database_type = hash  
default_database_type = dbm
```

default_delivery_slot_cost (défaut : 5)

Rythme où l'ordonnanceur du gestionnaire des files d'attente de Postfix est autorisé à donner la priorité à un message sur un autre.

Chaque transport maintient un "compteur de slot de livraison valide" pour chaque message. Un message peut prendre la priorité à un autre lorsqu'il peut être livré sans utiliser plus de slot (c'est à dire des invocations d'agents de livraison) que le compteur de message courant a accumulé (ou va accumuler – voir plus loin). Ce paramètre contrôle à quel rythme ce compteur est incrémenté – ceci arrive chaque fois que default_delivery_slot_cost destinataires ont été livrés.

Le coût 0 est utilisé pour désactiver le droit de préemption. La valeur minimale que l'algorithme de l'ordonnanceur peut utiliser est 2 – utilisez-la si vous voulez maximiser la rapidité de transfert des messages. Bien qu'il n'y ai pas de maximum, les valeurs élevées telles 50 n'ont aucun sens.

La seule raison pour laquelle 2 n'est pas la valeur par défaut est qu'il affecte la livraison des listes de diffusion. Dans le pire des cas, leur temps de livraison peut prendre entre (coût+1/coût) et (coût/coût-1) plus de temps que si le droit de préemption est désactivé. La valeur par défaut 5 est un compromis raisonnable évitant que les livraisons des listes de diffusions ne soient ralenties de 20 à 25% dans le pire des cas.

Exemples :

default_delivery_slot_cost = 0

default_delivery_slot_cost = 2

default_delivery_slot_discount (défaut : 50)

Valeur par défaut pour les paramètre spécifiques <transport>_delivery_slot_discount.

Ce paramètre détermine le moment où une préemption de message peut avoir lieu. Au lieu d'attendre que le compteur ait atteint la valeur désirée, la préemption peut arriver lorsque *transport_delivery_slot_discount%* de la valeur requise plus *transport_delivery_slot_loan* restent à accumuler. Notez que la valeur totale doit être atteinte avant qu'une autre préemption puisse avoir lieu ultérieurement.

default_delivery_slot_loan (défaut : 3)

Valeur par défaut pour les paramètres spécifiques <transport>_delivery_slot_loan.

Ce paramètre détermine le moment où une préemption de message peut avoir lieu. Au lieu d'attendre que le compteur ait atteint la valeur désirée, la préemption peut arriver lorsque *transport_delivery_slot_discount%* de la valeur requise plus *transport_delivery_slot_loan* restent à accumuler. Notez que la valeur totale doit être atteinte avant qu'une autre préemption puisse avoir lieu ultérieurement.

default_destination_concurrency_limit (défaut : 20)

Le nombre maximal par défaut de livraisons parallèles vers la même destination. C'est la limite par défaut pour la livraison via les agents de livraison lmtp(8), pipe(8), smtp(8) et virtual(8).

default_destination_recipient_limit (défaut : 50)

Nombre maximum par défaut de destinataires par livraison de message. C'est la limite par défaut pour la livraison via les agents de livraison lmtp(8), pipe(8), smtp(8) et virtual(8).

Mettre ce paramètre à 1 change implicitement la valeur du paramètre précédent.

default_extra_recipient_limit (défaut : 1000)

Valeur par défaut de la limite extra de chaque transport imposée au nombre de destinataires en mémoire. Cet espace destinataire extra est réservé pour le cas où l'ordonnanceur du gestionnaire des files d'attente de Postfix donne la priorité à un message sur un autre et soudainement requiert d'autres slot destinataires pour ce message pour éviter une dégradation des performances.

default_minimum_delivery_slots (défaut : 3)

Nombre de slots destinataires qu'un message doit avoir pour pouvoir invoquer l'algorithme de l'ordonnanceur du gestionnaire des files d'attente de Postfix. Les messages qui ne peuvent atteindre cette valeur ne peuvent être préemptés.

default_privs (défaut : nobody)

Les droits par défaut utilisés par l'agent de livraison local(8) pour la livraison à une commande ou un fichier externe. Ces droits sont utilisés lorsque la livraison est requise par un fichier aliases(5) qui appartient à **root** ou lorsque la livraison est effectuée à **root**. **N'INDIQUEZ PAS ICI UN UTILISATEUR PRIVILÉGIÉ OU L'UTILISATEUR POSTFIX.**

default_process_limit (défaut : 100)

Nombre maximum par défaut de processus fils de Postfix qui fournissent le même service. Cette limite peut être surchargée pour chacun des services dans le fichier master.cf.

default_rbl_reply (défaut : voir la sortie de "postconf -d")

La réponse par défaut du serveur SMTP pour une requête rejetée par une restriction basée sur une RBL. Elle peut être surchargée par les entrées spécifiques de la table de correspondances optionnelle rbl_reply_maps.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

Ce modèle est soumis à exactement un niveau de substitutions de \$name :

\$client

Le nom de machine et l'adresse IP du client, formatée ainsi: name[address].

\$client_address

L'adresse IP du client.

\$client_name

Le nom de machine du client ou "**unknown**".

\$reverse_client_name

Le nom de machine du client issu de la consultation DNS inverse (adresse->nom), ou "unknown". Voir reject_unknown_reverse_client_hostname pour plus de précisions.

\$helo_name

Le nom de machine transmit dans la commande HELO ou EHLO ou une chaîne vide.

\$rbl_class

Le type d'entrée rejetée: Client host, Helo command, Sender address, ou Recipient address.

\$rbl_code

Le code numérique de réponse SMTP, indiqué par le paramètre de configuration maps_rbl_reject_code. Note : Le code de réponse numérique est requis et doit apparaître au début de la réponse. Avec les versions 2.3 et supérieures de Postfix, cette information peut être suivie par un code de statut amélioré RFC 3463.

\$rbl_domain

Le domaine RBL domain où \$rbl_what est rejeté.

\$rbl_reason

La raison pour laquelle \$rbl_what est rejeté ou une chaîne vide.

\$rbl_what

L'entité rejeté (une adresse IP, un nom d'hôte, un nom de domaine ou une adresse de messagerie dont le domaine est inscrit en liste noire).

\$recipient

L'adresse de destination ou <> dans le cas d'une adresse nulle.

\$recipient_domain

Le domaine destinataire domain ou une chaîne vide.

\$recipient_name

La partie gauche de l'adresse de destination ou <> dans le cas d'une adresse nulle.

\$sender

L'adresse d'expédition ou <> dans le cas d'une adresse nulle.

\$sender_domain

Le domaine de l'expéditeur ou une chaîne vide.

\$sender_name

La partie gauche de l'adresse d'expédition ou <> dans le cas d'une adresse nulle.

\${name?text}

Inscrit `text' si \$name n'est pas vide.

\${name:text}

Inscrit `text' si \$name est vide.

Au lieu de \$name vous pouvez également indiquer \${name} ou \$(name).

Note : lorsqu'un code de statut amélioré est indiqué dans un modèle de réponse RBL, il est sujet à modification. Les transformations suivantes sont nécessaires lorsque le même modèle de réponse RBL est utilisé pour les restrictions d'accès sur le client, le *helo*, l'expéditeur ou le destinataire.

- ◇ Lorsqu'une adresse d'expédition est rejetée, le serveur SMTP de Postfix transforme un statut DSN de destinataire (e.g., 4.1.1–4.1.6) en statut DSN d'expédition correspondant, et vice versa.
- ◇ Lorsqu'une information autre qu'une adresse est rejetée (tel un argument de la commande HELO ou le nom de machine ou l'adresse IP du client), le serveur SMTP de Postfix transforme un statut DSN d'expéditeur et de destinataire en un statut DSN non–adresse générique (e.g., 4.0.0).

default_recipient_limit (défaut : 10000)

La limite par défaut pour chaque transport du nombre de destinataires en mémoire. Ces limites peuvent prendre le pas sur le paramètre global qmgr_message_recipient_limit après que le message a été assigné à ce transport. Voyez également default_extra_recipient_limit et qmgr_message_recipient_minimum.

default_transport (défaut : smtp)

Le transport par défaut pour les domaines qui ne correspondent pas à \$mydestination, \$inet_interfaces, \$proxy_interfaces, \$virtual_alias_domains, \$virtual_mailbox_domains, ou \$relay_domains. Dans l'ordre de précedence inverse, la destination suivante est déterminée par \$default_transport, \$sender_dependent_relayhost_maps, \$relayhost, ou par le domaine de destination. Cette information peut être surchargée par la table transport(5).

Indiquez une chaîne sous la forme *transport:nexthop*, où *transport* est le nom du transporteur du message défini dans le fichier master.cf. La partie *:nexthop* est optionnelle. Pour plus de détails, reportez-vous à la page de manuel transport(5).

Exemple :

```
default_transport = uucp:relayhostname
```

default_verp_delimiters (défaut : +=)

Les deux caractères de délimitation par défaut. Ils sont utilisés lorsqu'aucun délimiteur n'est spécifié dans la commande SMTP XVERP ou avec l'option **-V** sur la ligne de commande **"sendmail -V"**. Indiquez les caractères autorisés par le paramètre verp_delimiter_filter.

Cette fonctionnalité est disponible dans les versions 1.1 et supérieures de Postfix.

defer_code (défaut : 450)

Le code de réponse numérique renvoyée par le serveur SMTP de Postfix lorsqu'un client SMTP distant est rejeté par la restriction "defer".

Ne changez pas ceci avant d'avoir bien compris la [RFC 821](#).

defer_service_name (défaut : defer)

Le nom du service [defer\(8\)](#). Ce service maintient un enregistrement des tentatives de livraison échouées et génère les avis de non-livraison.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

defer_transports (défaut : vide)

Les noms des transports qui ne doivent pas livrer avant que quelqu'un ne lance "**sendmail -q**" ou une commande équivalente. Indiquez zero ou plus de noms de transports de livraison de messages (qui apparaissent dans le premier champ du fichier).

Exemple :

```
defer_transports = smtp
```

delay_logging_resolution_limit (défaut : 2)

Nombre maximal de chiffres après la virgule lors de l'enregistrement de valeurs de délai inférieures à la seconde. Indiquez un nombre compris entre 0 et 6.

Les délais plus long sont arrondis en nombres entiers ; ceux inférieurs à [delay_logging_resolution_limit](#) sont arrondis à "0", et les valeurs de petits délais sont enregistrés avec au plus 2 chiffres significatifs.

Le format de l'enregistrement "delays=a/b/c/d" est le suivant :

- ◇ a = temps avant le gestionnaire des files d'attentes, incluant la transmission du message
- ◇ b = temps passé en file d'attente
- ◇ c = temps de connexion, incluant les échanges DNS, EHLO et TLS
- ◇ d = temps de transmission du message

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

delay_notice_recipient (défaut : postmaster)

Le destinataire des notifications au postmaster avec les en-têtes de message des messages qui ne peuvent être livrés dans le temps [\\$delay_warning_time](#).

Cette fonctionnalité est activée avec le paramètre [delay_warning_time](#).

delay_warning_time (défaut : 0h)

Temps au delà duquel l'expéditeur reçoit les en-têtes d'un message toujours en file d'attente.

Pour activer cette fonctionnalité, indiquez une valeur non nulle.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est h (heures).

deliver_lock_attempts (défaut : 20)

Nombre maximal de tentatives d'acquisition de l'accès exclusif à un fichier boîte-aux-lettres ou un fichier de logs [bounce\(8\)](#).

deliver_lock_delay (défaut : 1s)

Temps entre deux tentatives d'acquisition de l'accès exclusif à un fichier boîte-aux-lettres ou un fichier de logs [bounce\(8\)](#).

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

disable_dns_lookups (défaut : no)

Désactive la consultation DNS dans les clients SMTP et LMTP de Postfix. Lorsqu'elle est désactivée, les hôtes sont recherchés par la fonction système `gethostbyname()` qui normalement consulte aussi `/etc/hosts`.

Les consultations DNS sont activées par défaut.

disable_mime_input_processing (défaut : no)

Désactive le traitement MIME à la réception des messages. Cela signifie qu'aucun traitement n'est effectué sur les en-têtes de message Content-Type: et que tout le texte après les en-têtes initiaux du message est considéré comme faisant partie du corps du message.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

Le traitement Mime en entrée est activé par défaut et est nécessaire pour reconnaître les en-têtes MIME dans le contenu du message.

disable_mime_output_conversion (défaut : no)

Désactive la conversion du format 8BITMIME en format 7BIT. Cette conversion est nécessaire lorsque le serveur de destination n'annonce pas le support 8BITMIME.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

disable_verp_bounces (défaut : no)

Désactive l'envoi d'un rapport de rejet par destinataire.

La valeur par défaut (actif) est nécessaire à `ezmlm`.

Cette fonctionnalité est disponible dans les versions 1.1 et supérieures de Postfix.

disable_vrfy_command (défaut : no)

Désactive la commande SMTP VRFY. Ceci arrête certaines techniques utilisées pour obtenir des adresses mail.

Exemple :

```
disable_vrfy_command = no
```

dont_remove (défaut : 0)

Ne pas supprimer les fichiers de la file d'attente et les sauver dans la file "saved". C'est une aide au déboguage. Pour examiner les informations de l'enveloppe et le contenu d'un fichier en file d'attente Postfix, utilisez la commande `postcat(1)`.

double_bounce_sender (défaut : double-bounce)

L'adresse d'expédition des notifications au postmaster qui sont générées par le système de messagerie. Tout le courrier à destination de cette adresse est silencieusement supprimé pour éviter les boucles de message.

duplicate_filter_limit (défaut : 1000)

Nombre maximal d'adresses en mémoire du filtre des adresses dupliquées pour les remplacements d'`aliases(5)` ou de `virtual(5)` ou pour les affichages de `showq(8)`.

empty_address_recipient (défaut : MAILER-DAEMON)

Le destinataire des messages adressés à l'adresse nulle. Postfix n'accepte pas de telles adresses dans les commandes SMTP, mais elles peuvent être créées localement comme résultat d'une erreur de configuration ou de logiciel.

enable_errors_to (défaut : no)

Rapporte les erreurs de livraison à l'adresse spécifiée dans l'en-tête non-standard Errors-To: au lieu de l'adresse d'expédition de l'enveloppe (cette fonctionnalité supprimée à partir de la version 2.2 de Postfix et désactivé par défaut sur Postfix 2.1 et supérieurs et est toujours activé sur les versions

antérieures).

enable_original_recipient (défaut : yes)

Active le support des en-tête de message X-Original-To. Cet en-tête est nécessaire pour les boîtes-aux-lettres multi-destinataires.

Lorsque ce paramètre est mis à "yes", le démon cleanup(8) procède à l'élimination des doublons sur la base des paires (destinataire original, destinataire réécrit) et génère un "destinataire original" dans les fichiers en file d'attente.

Lorsque ce paramètre est mis à "no", le démon cleanup(8) procède à l'élimination des doublons sur la base de l'adresse de destination réécrite seulement et génère un "destinataire original" vide dans les fichiers en file d'attente.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix. Avec Postfix 2.0, le support des en-tête de message X-Original-To est toujours activé. Les versions antérieures de Postfix n'ont pas de support de ces en-têtes.

error_notice_recipient (défaut : postmaster)

Le destinataire des notifications au postmaster à propos des problèmes de livraison de message causés par la politique, les ressources, le logiciel ou les erreurs de protocole. Ces notifications sont activées avec le paramètre notify_classes.

error_service_name (défaut : error)

le nom du pseudo agent de livraison error(8). Ce service rejette systématiquement le courrier comme non livrable.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

execution_directory_expansion_filter (défaut : voir la sortie de "postconf -d")

Restreint les caractères que l'agent de livraison local(8) autorisés dans les substitutions \$name de \$command_execution_directory. Les caractères non autorisés sont remplacés par des underscores ("_").

cette fonctionnalité est disponible dans les versions 2.2 et supérieures de Postfix.

expand_owner_alias (défaut : no)

Lors de la livraison à un alias "aliasname" qui a un alias compagnon "owner-aliasname", mettre l'adresse d'expédition de l'enveloppe à la valeur retournée par la conversion de l'alias "owner-aliasname". Normalement, Postfix met l'adresse d'expédition de l'enveloppe au nom de l'alias "owner-aliasname".

export_environment (défaut : voir la sortie de "postconf -d")

La liste des variables d'environnement qu'un processus Postfix va exporter à un processus non-Postfix. La variable TZ est nécessaire pour conserver une cohérence du temps sur les système System-V-ish.

Indiquez une liste de noms et/ou de paires name=value, séparées par des espaces ou des virgules.

Exemple :

```
export_environment = TZ PATH=/bin:/usr/bin
```

extract_recipient_limit (défaut : 10240)

Le nombre maximal d'adresses de destination que Postfix extraira des en-têtes de message lorsqu'un message est soumis par "sendmail -t".

Cette fonctionnalité a été supprimée dans Postfix 2.1.

fallback_relay (défaut : vide)

Liste optionnelle de machines relais pour les destinations qui n'ont pu être trouvées où qui sont innaccessibles. À partir de la version 2.3 de Postfix 2.3, ce paramètre est renommé smtp fallback_relay.

Par défaut, le message est retourné à l'expéditeur lorsqu'une destination n'est pas trouvée et le livraison est retardée si la destination n'est pas accessible.

Ces relais doivent être des destinations SMTP. Indiquez un domaine, machine, machine:port, [machine]:port, [adresse] ou [adresse]:port; la forme [machine] désactive la consultations DNS des champs MX. Si vous indiquez plusieurs destinations SMTP, Postfix les essaiera dans l'ordre indiqué.

Note : sur les versions de Postfix antérieures à la 2.2, n'utilisez pas la fonctionnalité fallback_relay lorsque vous relayer le courrier pour un MX primaire ou de secours : les messages boucleront entre la machine MX Postfix et le relais fallback_relay lorsque la destination finale n'est pas joignable.

- ◇ Dans main.cf indiquez "relay_transport = relay",
- ◇ Dans master.cf indiquez "-o fallback_relay =" (i.e., vide) à la fin de l'entrée relay.
- ◇ Dans les tables de transport, utilisez "relay:saut-suivant..." en partie droite pour les entrées MX primaire ou backup.

Postfix version 2.2 ou supérieure n'utilise pas le relais de secours fallback_relay pour les destinations pour lesquelles ce dernier est MX.

fallback_transport (défaut : vide)

Transport de livraison de messages optionnel que l'agent de livraison local(8) utilisera pour les noms qui ne sont trouvés ni dans la base des alias ni dans la base de données UNIX (passwd).

Le choix de la livraison locale se fait dans l'ordre suivant : les alias, les fichiers .forward, mailbox_transport_maps, mailbox_transport, mailbox_command_maps, mailbox_command, home_mailbox, mail_spool_directory, fallback_transport_maps, fallback_transport et luser_relay.

fallback_transport_maps (défaut : vide)

Tables de correspondance optionnelles de transports pour la livraison des messages par destinataire pour les destinataires que l'agent de livraison local(8) ne trouve pas dans les alias ou la base de données des mots-de-passe UNIX.

Le choix de la livraison locale se fait dans l'ordre suivant : les alias, les fichiers .forward, mailbox_transport_maps, mailbox_transport, mailbox_command_maps, mailbox_command, home_mailbox, mail_spool_directory, fallback_transport_maps, fallback_transport et luser_relay.

Pour des raisons de sécurité, cette fonctionnalité n'autorise pas les substitutions \$nombre dans les tables d'expressions régulières.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

fast_flush_domains (défaut : \$relay_domains)

Liste optionnelle des destinations qui sont éligible pour l'enregistrement dans logs par destination des messages mis en file d'attente pour ces destinations.

Par défaut, Postfix maintient des fichiers de logs "fast flush" seulement pour les destinations que le serveur SMTP de Postfix est censé relayer (i.e. la valeur par défaut est : "fast_flush_domains = \$relay_domains"; consultez le paramètre relay_domains de cette page.

Documentation de Postfix en français

Indiquez une liste de machines ou de domaines, des `"/noms/de/fichiers"` de correspondances ou des tables de correspondances `"type:table"`, séparés par des virgules et/ou des espaces. Continuez les lignes longues en commençant la ligne suivante par des espaces. un `"/nom/de/fichier"` de correspondances est remplacé par son contenu; une table de correspondances `"type:table"` est utilisée lorsque le domaine ou ses domaines parents apparaissent comme clef de consultation.

Indiquez `"fast_flush_domains ="` pour désactiver cette fonctionnalité.

fast_flush_purge_time (défaut : 7d)

Temps au delà duquel un fichier journal par destination vide "fast flush" est effacé.

Vous pouvez indiquer le temps avec un nombre ou un nombre suivi d'une lettre qui indique l'unité de temps : s=secondes, m=minutes, h=heures, d=jours, w=semaines. L'unité de temps par défaut est le jour.

fast_flush_refresh_time (défaut : 12h)

Le temps au delà duquel un fichier journal par destination non-vide "fast flush" mais non lu doit être rafraîchi. Les contenus de ces fichiers sont rafraîchis en essayant la livraison de tous les messages listés dans le fichier de logs.

Vous pouvez indiquer le temps par un nombre ou un nombre suivi d'une lettre qui indique l'unité de temps : s=secondes, m=minutes, h=heures, d=jours, w=semaines. L'unité de temps par défaut est heures.

fault_injection_code (défaut : 0)

Force le test interne spécifié à échouer pour tester le résultat d'erreurs difficiles à reproduire autrement.

flush_service_name (défaut : flush)

Le nom du service `flush(8)`. Ce service maintient des fichiers de logs par destination avec le nom des fichiers en file d'attente correspondant à des messages mis en file d'attente pour ces destinations.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

fork_attempts (défaut : 5)

Nombre maximum de tentatives de `fork()` d'un processus fils.

fork_delay (défaut : 1s)

Délai entre deux tentatives de `fork()` d'un processus fils.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

forward_expansion_filter (défaut : voir la sortie de "postconf -d")

Restreint les caractères que l'agent de livraison `local(8)` autorise dans les substitutions `$name` de `$forward_path`. Les caractères autres sont remplacés par des underscores ("`_`").

forward_path (défaut : voir la sortie de "postconf -d")

Liste de recherche d'un fichier `.forward` par l'agent de livraison `local(8)` pour la méthode de livraison spécifiée par l'utilisateur. Le premier fichier trouvé est utilisé.

Les substitutions suivantes sont effectuées sur `forward_path` avant qu'une recherche n'arrive. Le résultat des substitutions `$name` est filtré avec les caractères indiqués au paramètre `forward_expansion_filter`.

\$user

Le nom du destinataire.

\$shell

Le nom du shell du login du destinataire.

\$home

Le répertoire personnel du destinataire.

\$recipient

L'adresse complète de destination.

\$extension

L'extension optionnelle de l'adresse de destination.

\$domain

Le domaine du destinataire.

\$local

La partie locale entière du destinataire.

\$recipient_delimiter

Le délimiteur système de l'extension de adresse de destination.

\${name?value}

Substitue *value* à *\$name* si *\$name* est non vide.

\${name:value}

Substitue *value* à *\$name* si *\$name* est vide.

Au lieu de *\$name*, vous pouvez utiliser *\$(name)* ou *\${name}*.

Exemples :

```
forward_path = /var/forward/$user
forward_path =
    /var/forward/$user/.forward$recipient_delimiter$extension,
    /var/forward/$user/.forward
```

***frozen_delivered_to* (défaut : yes)**

Met à jour l'idée l'adresse Delivered-To: calculée par de l'agent de livraison local(8) (voir prepend_delivered_header) une seule fois au démarrage de la tentative de livraison ; ne met pas à jour l'adresse Delivered-To: lors du remplacement par les alias ou les fichiers .forward.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix. Les versions antérieures de Postfix se comportent comme si ce paramètre était à "no". L'ancien paramétrage peut être couteux avec des fichiers alias ou .forward complexes. Lorsqu'un fichier alias ou .forward change l'adresse Delivered-To:, il utilise un fichier en file d'attente et un processus cleanup pendant que le message est transféré.

***hash_queue_depth* (défaut : 1)**

Nombre de niveaux de sous-répertoires d'un répertoire de file d'attente listée par le paramètre hash_queue_names.

Après avoir changé le paramètre hash_queue_names ou hash_queue_depth, exécutez la commande "postfix reload".

***hash_queue_names* (défaut : voir la sortie de "postconf -d")**

Noms des répertoires des files d'attente eux-mêmes coupés en multiples sous-répertoires.

Avant la version 2.2 de Postfix, la liste par défaut des fichiers de file d'attente était sensiblement plus importante. Les résultats des tests sur les technologies des systèmes de fichier suggèrent que le hachage des files d'attentes entrante et active n'est plus nécessaire. Moins de répertoires hachés diminue le temps nécessaire au redémarrage de Postfix.

Après avoir changé le paramètre hash_queue_names ou hash_queue_depth, exécutez la commande "postfix reload".

***header_address_token_limit* (défaut : 10240)**

Nombre maximum d'adresses autorisées dans un en-tête de message. Les informations au-delà de cette limite sont effacées. La limite est traitée par le serveur cleanup(8).

header_checks (défaut : vide)

Tables de correspondances optionnelles pour l'inspection du contenu des en-têtes de message primaires non-MIME, tel que décrit à la page de manuel header_checks(5).

header_size_limit (défaut : 102400)

Nombre maximal d'octets en mémoire pour un en-tête de message. L'excédent est supprimé. Cette limite est traitée par le serveur cleanup(8).

helpful_warnings (défaut : yes)

Logs d'avertissement à propos de paramètres de configuration problématiques fournissant des suggestions.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

home_mailbox (défaut : vide)

Chemin optionnel d'un fichier de boîte-aux-lettres relatif au répertoire personnel d'un utilisateur local(8).

Indiquez un chemin se terminant par "/" pour des livraisons style qmail.

L'ordre de traitement des fonctionnalités de livraison locale est : aliases, fichiers .forward, mailbox transport, mailbox command maps, mailbox command, home mailbox, mail spool directory, fallback transport et user relay.

Exemples :

```
home_mailbox = Mailbox
home_mailbox = Maildir/
```

hopcount_limit (défaut : 50)

Nombre maximum d'en-têtes de message Received: autorisés dans les en-têtes de message primaires. Un message outrepassant cette limite est rejeté pour éviter des boucles de message.

html_directory (défaut : voir la sortie de "postconf -d")

L'emplacement des fichiers HTML de Postfix décrivant comment compiler, configurer ou opérer un sous-système ou une fonctionnalité particulière de Postfix.

ignore_mx_lookup_error (défaut : no)

Ignore les consultations DNS MX qui ne produisent aucune réponse. Par défaut, le client SMTP de Postfix retarde la livraison puis essaie de nouveau après un certain délai. Cette fonctionnalité est requise par le standard SMTP.

Indiquez "ignore_mx_lookup_error = yes" pour forcer une consultation d'enregistrement A du DNS à la place. Ceci viole le standard SMTP et peut engendrer des échecs de livraison.

import_environment (défaut : voir la sortie de "postconf -d")

La liste des paramètres d'environnement qu'un processus Postfix importera d'un processus parent non-Postfix. Des exemples de tels paramètres :

TZ

Nécessaire pour une saine gestion du temps sur les systèmes System-V-ish.

DISPLAY

Nécessaire pour les démons de débogage de Postfix utilisant un débogueur X-windows.

XAUTHORITY

Nécessaire pour les démons de débogage de Postfix utilisant un débogueur X-windows.

MAIL_CONFIG

Nécessaire au travail de "**postfix -c**".

Indiquez une liste de noms et/ou de paires noms=valeur, séparées par des espaces ou des virgules. La forme nom=valeur est supportée par les versions 2.1 et supérieures de Postfix.

***in_flow_delay* (défaut : 1s)**

Temps d'attente avant d'accepter un nouveau message lorsque le taux d'arrivée de messages excède le taux de livraison. Cette fonctionnalité est activée par défaut (sauf sur SCO UNIX à cause d'un bug SCO).

Avec la limite par défaut de 100 processus serveurs SMTP, "in_flow_delay = 1s" limite l'arrivée à 100 messages par seconde au-delà du nombre de messages livrés par seconde.

Indiquez 0 pour désactiver cette fonctionnalité. 1 à 10 secondes sont des valeurs raisonnables.

***inet_interfaces* (défaut : all)**

Les adresses réseau par lesquelles le système de messagerie reçoit les messages. Par défaut, le logiciel accepte toutes les interfaces de la machine. Ce paramètre contrôle également la livraison des messages à utilisateur@[address.ip].

Note 1 : vous devez arrêter et redémarrer Postfix lorsque ce paramètre change.

Note 2 : les adresses peuvent être encadrées par [], mais cette forme n'est pas recommandée ici.

Lorsque inet_interfaces ne contient qu'une seule adresse IP qui n'est pas l'adresse de la boucle locale (réseau 127), le client SMTP de Postfix utilisera cette adresse comme adresse source pour le courrier sortant.

Sur un firewall hébergement plusieurs domaines avec des instances séparées de Postfix écoutant sur les interfaces "inside" et "outside", ceci peut éviter à chaque instance d'être capable d'atteindre des serveurs de "l'autre côté" du firewall. Mettre smtp_bind_address à 0.0.0.0 pour IPv4 ou smtp_bind_address6 à :: évite ce problème potentiel.

Une meilleure solution est de laisser inet_interfaces à la valeur par défaut et au lieu d'utiliser des adresses IP explicites dans le fichier master.cf. Ceci préserve la détection des boucles SMTP en s'assurant que chaque côté du firewall connaît l'autre adresse IP de la même machine. Renseigner \$inet_interfaces avec un adresse IP unique est utile pour l'hébergement virtuel de domaines sur une adresse IP secondaire, lorsque chaque adresse IP sert un domaine différent (et a une valeur \$myhostname différente).

Voyez aussi le paramètre proxy_interfaces, pour les adresses transférées par le biais d'un proxy ou d'un traducteur d'adresse.

Exemples :

```
inet_interfaces = all (DÉFAUT)
inet_interfaces = loopback-only (Postfix version 2.2 ou supérieure)

inet_interfaces = 127.0.0.1
inet_interfaces = 127.0.0.1, [::1] (Postfix version 2.2 ou supérieure)
inet_interfaces = 192.168.1.2, 127.0.0.1
```

***inet_protocols* (défaut : ipv4)**

Le protocole Internet que Postfix tentera d'utiliser lorsqu'il crée ou accepte des connexions. Indiquez "ipv4" et/ou "ipv6", séparés par des espaces ou des virgules. La forme "all" est équivalente à "ipv4, ipv6" ou "ipv4", suivant que le système supporte IPv6 ou non.

Documentation de Postfix en français

Cette fonctionnalité est disponibles dans les versions 2.3 et supérieures de Postfix.

Note : vous DEVEZ arrêter et redémarrer Postfix après avoir changer ce paramètre.

Sur les système qui antérieurs au support IPV6_V6ONLY ([RFC 3493](#)), un serveur IPv6 acceptera également les connexions IPv4 même si IPv4 est désactivé dans le paramètre `inet_protocols`. Sur les systèmes supportant IPV6_V6ONLY, Postfix utilisera des sockets distinctes pour IPv6 et IPv4, et chacune n'acceptera que les connexions correspondant à leur protocole.

Lorsque le support IPv4 est activé via le paramètre `inet_protocols`, Postfix essaie de rechercher les enregistrement de type A du DNS et convertit les adresses des clients IPv4—dans—IPv6 (::ffff:1.2.3.4) en leur forme IPv4 originale (1.2.3.4). Ceci est nécessaire pour les machines antérieures au support IPV6_V6ONLY ([RFC 3493](#)).

Lorsque le support IPv6 est activé via le paramètre `inet_protocols`, Postfix effectue les recherches d'enregistrement DNS de type AAAA.

Lorsque les supports IPv4 et IPv6 sont tous deux activés, le client SMTP de Postfix tente de se connecter via IPv6 avant d'essayer IPv4.

Exemples :

```
inet_protocols = ipv4 (DEFAULT)
inet_protocols = all
inet_protocols = ipv6
```

```
inet_protocols = ipv4, ipv6
```

initial_destination_concurrency (défaut : 5)

Le nombre initial de livraisons parallèles vers la même destination. Cette limite s'applique aux livraisons via les agents de livraison `smtp(8)`, `pipe(8)` et `virtual(8)`.

Attention : avec une valeur fixée à 1, un seul message incorrect peut suffire à bloquer le courrier de tout un site.

invalid_hostname_reject_code (défaut : 501)

Le code numérique de réponse du serveur SMTP de Postfix lorsqu'un paramètre de la commande HELO ou EHLO du client est rejeté par la restriction `reject_invalid_hostname`.

Ne changez pas ceci avant d'avoir bien compris la [RFC 821](#).

ipc_idle (défaut : 100s)

Temps au delà duquel un client ferme un canal de communication interne inactif. Le but est d'autoriser les serveurs à les fermer volontairement. C'est utilisé, par exemple, par les clients de résolution et de réécriture d'adresses.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

ipc_timeout (défaut : 3600s)

Temps limite pour envoyer ou recevoir des informations via un canal de communication interne. Le but est de fermer les situations bloquées. Si le temps limite est dépassé, le logiciel se termine avec une erreur fatale.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

ipc_ttl (défaut : 1000s)

Temps au delà duquel un client ferme un canal de communication interne. Le but est de permettre aux serveurs de les clore volontairement après avoir atteint leur limites client. C'est utilisé, par exemple, par les clients de résolution et de réécriture d'adresses.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

line_length_limit (défaut : 2048)

Limite de longueur des lignes. Les lignes trop longues sont coupées en plusieurs de cette longueur au plus; elles sont reconstruites au cours de la livraison.

lmtp_bind_address (défaut : vide)

La version spécifique LMTP du paramètre smtp_bind_address. Reportez-vous au paragraphe ad-hoc pour les détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_bind_address6 (défaut : vide)

La version spécifique LMTP du paramètre smtp_bind_address6. Reportez-vous au paragraphe ad-hoc pour les détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_cache_connection (défaut : yes)

Garde les connexions du client LMTP de Postfix client connections ouvertes \$max_idle secondes. Lorsque le client LMTP reçoit une requête pour la même destination, la connexion est réutilisée.

La réalité de ces connexions cachées sera déterminée par le nombre de serveurs LMTP un fonctionnement et la limite de parallélisme définie pour le client LMTP. Les connexions cachées sont fermées après n'importe laquelle des conditions suivantes :

- ◇ La limite d'inactivité du client LMTP est atteinte. Cette limite est indiquée par le paramètre de configuration max_idle
- ◇ Une requête de livraison indique une destination différente que la destination cachée.
- ◇ La limite par processus du nombre de requêtes de livraison est atteinte. Cette limite est indiquée par le paramètre de configuration max_use.
- ◇ Lors d'une autre requête de livraison, the serveur LMTP associé à la session en cours ne répond pas à la la commande RSET.

La plupart de ces limitations seront supprimées après que Postfix implemente un cache de connexion partagé par de multiples programmes client LMTP.

lmtp_cname_overrides_servername (défaut : yes)

La version spécifique LMTP du paramètre smtp_cname_overrides_servername. Reportez-vous au paragraphe ad-hoc pour les détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_connect_timeout (défaut : 0s)

Temps limite pour qu'un client LMTP termine une connexion TCP connection ou zéro (utilise la limite intrinsèque de système d'exploitation). Lorsqu'aucune connexion n'a pu être établie avant la limite, the client LMTP essaie l'adresse suivante de la liste des échangeurs de messagerie.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

Exemple :

lmtp_connect_timeout = 30s

lmtp_connection_cache_destinations (défaut : vide)

La version spécifique LMTP du paramètre smtp_connection_cache_destinations. Reportez-vous au paragraphe ad-hoc pour les détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_connection_cache_on_demand (défaut : yes)

La version spécifique LMTP du paramètre smtp_connection_cache_on_demand. Reportez-vous au paragraphe ad-hoc pour les détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_connection_cache_time_limit (défaut : 2s)

La version spécifique LMTP du paramètre smtp_connection_cache_time_limit. Reportez-vous au paragraphe ad-hoc pour les détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_connection_reuse_time_limit (défaut : 300s)

La version spécifique LMTP du paramètre smtp_connection_reuse_time_limit. Reportez-vous au paragraphe ad-hoc pour les détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_data_done_timeout (défaut : 600s)

Temps limite pour que le client LMTP envoie le "." LMTP et que le serveur réponde. A défaut de réponse dans le temps imparti, un avertissement est enregistré dans les journaux indiquant que le message a pu être livré plusieurs fois.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

lmtp_data_init_timeout (défaut : 120s)

Temps limite pour que le client LMTP transmette la commande LMTP DATA, et pour que le serveur envoie la réponse.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

lmtp_data_xfer_timeout (défaut : 180s)

Temps limite pour que le client LMTP transmette le contenu du message LMTP. Lorsque la connexion dure au delà de \$lmtp_data_xfer_timeout, le client LMTP termine le transfert.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

lmtp_destination_concurrency_limit (défaut : \$default_destination_concurrency_limit)

Nombre maximum de livraisons parallèles vers la même destination via le transporteur de message LMTP. Cette limite est utilisée par le gestionnaire des files d'attente. Le nom du transporteur de message correspond au premier champ du fichier master.cf.

lmtp_defer_if_no_mx_address_found (défaut : no)

La version spécifique LMTP du paramètre smtp_defer_if_no_mx_address_found. Reportez-vous au paragraphe ad-hoc pour les détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_destination_recipient_limit (défaut : \$default_destination_recipient_limit)

Nombre maximum de destinataires par livraison via le transporteur de message lmtp. Cette limite est utilisée par le gestionnaire des files d'attente. Le nom du transporteur de message correspond au premier champ du fichier master.cf.

Mettre ce paramètre à 1 change le sens de lmtp_destination_concurrency_limit de parallélisme par domaine en parallélisme par destinataire.

lmtp_discard_lhlo_keyword_address_maps (défaut : vide)

Tables de correspondances indexées par les adresses des serveurs LMTP distants, avec la liste (insensible à la casse) des mots-clefs LHLO (pipelining, starttls, auth, etc.) que le client LMTP ignorera dans la réponse LHLO du serveur LMTP distant. Voir lmtp_discard_lhlo_keywords pour les détails. La table n'est pas indexée par nom de machine comme smtpd_discard_ehlo_keyword_address_maps.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_discard_lhlo_keywords (défaut : \$myhostname)

Une liste insensible à la casse des mots-clefs LHLO (pipelining, starttls, auth, etc.) que le client LMTP ignorera dans la réponse LHLO du serveur LMTP distant.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

Notes :

- ◇ Indiquez le pseudo mot-clef **silent-discard** pour éviter d'enregistrer cette action dans les journaux.
- ◇ Utilisez la fonctionnalité lmtp_discard_lhlo_keyword_address_maps pour ignorer les mots-clefs LHLO sélectivement.

lmtp_enforce_tls (défaut : no)

La version spécifique LMTP du paramètre smtp_enforce_tls. Reportez-vous au paragraphe ad-hoc pour les détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_generic_maps (défaut : vide)

La version spécifique LMTP du paramètre smtp_generic_maps. Reportez-vous au paragraphe ad-hoc pour les détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_host_lookup (défaut : dns)

La version spécifique LMTP du paramètre smtp_host_lookup. Reportez-vous au paragraphe ad-hoc pour les détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_lhlo_name (défaut : \$myhostname)

Le nom de machine à envoyer dans la commande LHLO.

La valeur par défaut est le nom de la machine hôte. Indiquez un nom de machine ou [une.adr.esse.IP].

Cette information peut être indiquée dans le fichier main.cf pour tous les clients LMTP, ou être renseigné dans le fichier master.cf pour un client spécifique, par exemple :

/etc/postfix/master.cf:

lmtplhlo_timeout (défaut : 300s)

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

La version spécifique LMTP du paramètre smtp line length limit. Reportez-vous au paragraphe ad-hoc pour les détails.

lmtpl_mail_timeout (défaut : 300s)

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

La version spécifique LMTP du paramètre smtp_mx_address_limit. Reportez-vous au paragraphe ad-hoc pour les détails.

lmtp_mx_session_limit (défaut : 2)

La version spécifique LMTP du paramètre `smtp_mx_session_limit`. Reportez-vous au paragraphe ad-hoc pour les détails.

lmtplib_pix_workaround_delay_time (défaut : 10s)

La version spécifique LMTP du paramètre `smtp_pix_workaround_delay_time`. Reportez-vous au paragraphe ad-hoc pour les détails.

lmtplib_pix_workaround_threshold_time (défaut : 500s)

La version spécifique LMTP du paramètre `smtp_pix_workaround_threshold_time`. Reportez-vous au paragraphe ad-hoc pour les détails.

lmtp_quit_timeout (défaut : 300s)

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

La version spécifique LMTP du paramètre smtp_quote_rfc821_envelope. Reportez-vous au paragraphe ad-hoc pour les détails.

267

lmtp_randomize_addresses (défaut : yes)

La version spécifique LMTP du paramètre smtp_randomize_addresses. Reportez-vous au paragraphe ad-hoc pour les détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_rcpt_timeout (défaut : 300s)

Temps limite pour que le client LMTP envoie la commande RCPT TO, et pour que le serveur réponde.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

lmtp_rset_timeout (défaut : 120s)

Temps limite pour que le client LMTP envoie la commande RSET, et pour que le serveur réponde. Le client LMTP envoie RSET pour voie si une connexion cachée est toujours valide.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

lmtp_sasl_auth_enable (défaut : no)

Active l'authentification SASL dans le client LMTP de Postfix.

lmtp_sasl_mechanism_filter (défaut : vide)

La version spécifique LMTP du paramètre smtp_sasl_mechanism_filter. Reportez-vous au paragraphe ad-hoc pour les détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_sasl_password_maps (défaut : vide)

Tables optionnelles de consultation du client LMTP avec une entrée username:password par hôte ou domaine. Si aucune entrée ne correspond, le client LMTP de Postfix ne tentera pas de s'authentifier sur la machine en question.

lmtp_sasl_path (défaut : vide)

Information liée à l'implémentation passée au plug-in SASL sélectionné par lmtp_sasl_type. Généralement elle indique le nom d'un fichier de configuration ou un point de rendezvous.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_sasl_security_options (défaut : noplaintext, noanonymous)

Mécanismes d'authentification que le client LMTP de Postfix est autorisé à utiliser. La liste des mécanismes d'authentification disponible dépend du client SASL sélectionné dans lmtp_sasl_type.

Les fonctionnalité de sécurité suivantes sont définies pour l'implémentation SASL du client **cyrus**.

noplaintext

Interdit les méthodes utilisant les mots de passe en clair.

noactive

Interdit les méthodes sujettes à une attaque active (sans dictionnaire).

nodictionary

Interdit les méthodes d'authentification sujettes à une attaque passive (par dictionnaire).

noanonymous

Interdit les authentifications anonymes.

Exemple :

```
lmtp_sasl_security_options = noplaintext
```

lmtp_sasl_tls_security_options (défaut : \$lmtp_sasl_security_options)

La version spécifique LMTP du paramètre smtp_sasl_tls_security_options. Reportez-vous au paragraphe ad-hoc pour les détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_sasl_tls_verified_security_options (défaut : \$lmtp_sasl_tls_security_options)

La version spécifique LMTP du paramètre smtp_sasl_tls_verified_security_options. Reportez-vous au paragraphe ad-hoc pour les détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_sasl_type (défaut : cyrus)

Type de plug-in SASL que le client LMTP de Postfix doit utiliser pour l'authentification. La liste des types disponibles est affichée par la commande "**postconf -A**".

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_send_xforward_command (défaut : no)

Envoie une commande XFORWARD au serveur LMTP lorsque la réponse du serveur LMTP LHLO annonce le support XFORWARD. Ceci autorise un agent de livraison lmtp(8), utilisé pour filtrer le contenu du message injecté, à transférer le nom, l'adresse, le protocole et le nom passé par HELO du client original au filtre de contenu. Avant de mettre cette valeur à "yes", il faut vous assurer que le filtre supporte cette commande.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

lmtp_sender_dependent_authentication (défaut : no)

La version spécifique LMTP du paramètre smtp_sender_dependent_authentication. Reportez-vous au paragraphe ad-hoc pour les détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_skip_5xx_greeting (défaut : yes)

La version spécifique LMTP du paramètre smtp_skip_5xx_greeting. Reportez-vous au paragraphe ad-hoc pour les détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_skip_quit_response (défaut : no)

Attent la réponse à la commande LMTP QUIT.

lmtp_tcp_port (défaut : 24)

Le port TCP par défaut auquel se connecte le client LMTP de Postfix.

lmtp_starttls_timeout (défaut : 300s)

La version spécifique LMTP du paramètre smtp_starttls_timeout. Reportez-vous au paragraphe ad-hoc pour les détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_tcp_port (défaut : 24)

Le port TCP par défaut auquel le client LMTP doit se connecter.

lmtp_tls_enforce_peername (défaut : yes)

La version spécifique LMTP du paramètre smtp_tls_enforce_peername. Reportez-vous au paragraphe ad-hoc pour les détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_tls_note_starttls_offer (défaut : no)

La version spécifique LMTP du paramètre smtp_tls_note_starttls_offer. Reportez-vous au paragraphe ad-hoc pour les détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_tls_per_site (défaut : vide)

La version spécifique LMTP du paramètre smtp_tls_per_site. Reportez-vous au paragraphe ad-hoc pour les détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_tls_scert_verifydepth (défaut : 5)

La version spécifique LMTP du paramètre smtp_tls_scert_verifydepth. Reportez-vous au paragraphe ad-hoc pour les détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_use_tls (défaut : no)

La version spécifique LMTP du paramètre smtp_use_tls. Reportez-vous au paragraphe ad-hoc pour les détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

lmtp_xforward_timeout (défaut : 300s)

Temps limite pour que le client LMTP envoie la commande XFORWARD et que le serveur réponde.

En cas de problème, le client n'essaie PAS l'adresse suivante de la liste des échangeurs de messagerie.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

local_command_shell (défaut : vide)

Program shell optionel pour les livraisons locales aux commandes non-Postfix. Par défaut, les commandes non-Postfix sont exécutées directement; les commandes sont passées à /bin/sh seulement lorsqu'elles contiennent les méta caractères shell (Shebang: "#!/bin/sh").

"sendmail's restricted shell" (smrsh) est ce que la plupart utiliseront afin de restreindre les programmes qui peuvent être lancés par les fichiers types .forward (smrsh fait partie de la distribution Sendmail).

Note : lorsqu'un programme shell est indiqué, il est invoqué même si la commande ne contient pas les méta caractères shell.

Exemple :

```
local_command_shell = /some/where/smrsh -c
```

local_destination_concurrency_limit (défaut : 2)

Nombre maximum de livraisons parallèles via le transport local au même destinataire (lorsque "local_destination_recipient_limit = 1") ou nombre maximum de livraisons parallèles au même domaine local (lorsque "local_destination_recipient_limit > 1"). Cette limite est utilisée par le gestionnaire des files d'attente. Le nom du transporteur de message correspond au premier champ du fichier master.cf.

Une limite faible de l'ordre de 2 est recommandée, juste au cas où quelqu'un aurait une commande shell lourde dans son fichier .forward ou dans un alias (typiquement les gestionnaires de listes de diffusion). Vous ne souhaitez probablement pas en lancer beaucoup à la fois.

local_destination_recipient_limit (défaut : 1)

Documentation de Postfix en français

Nombre maximum de destinataires par livraison de message via le transporteur de messages local. Cette limite est utilisée par le gestionnaire des files d'attente. Le nom du transporteur de message correspond au premier champ du fichier master.cf.

Mettre ce paramètre à une valeur > 1 change le sens de local_destination_concurrency_limit de concurrence par destinataire en concurrence par domaine.

local_header_rewrite_clients (défaut : permit_inet_interfaces)

Ajoute le nom de domaine de \$myorigin ou \$mydomain aux adresses des en-têtes de message de ces clients seulement ; et ne réécrit pas les en-têtes de message des autres clients, ou ajoute le nom de domaine indiqué au paramètre remote_header_rewrite_domain.

See the append_at_myorigin and append_dot_mydomain parameters for details of how noms de domaine are appended to incomplete addresses.

Indiquez une liste de zéro ou plus des propositions suivantes :

permit_inet_interfaces

Append the nom de domaine in \$myorigin or \$mydomain when the client adresse IP matches \$inet_interfaces. This is enabled par défaut.

permit_mynetworks

Ajoute le nom de domaine de \$myorigin ou \$mydomain lorsque l'adresse IP du client correspond à un réseau ou une adresse de réseau listé dans \$mynetworks. Ce paramètre n'évite pas la réécriture des adresses dans les en-têtes des messages extérieurs lorsque le courrier provenant de l'extérieur est transféré par un système voisin.

permit_sasl_authenticated

Ajoute le nom de domaine de \$myorigin ou \$mydomain lorsque le client est authentifié avec succès via le protocole AUTH (RFC 2554). Ceci est activé par défaut.

permit_tls_clientcerts

Ajoute le nom de domaine de \$myorigin ou \$mydomain lorsque le certificat TLS du client est vérifié avec succès, et l'empreinte du certificat est listé sur le serveur. Ceci est activé par défaut.

permit_tls_all_clientcerts

Ajoute le nom de domaine de \$myorigin ou \$mydomain lorsque le certificat client est vérifié avec succès sans vérifier s'il est listé sur le serveur ou signé par une autorité reconnue.

check_address_map type:table

type:table

Ajoute le nom de domaine de \$myorigin ou \$mydomain lorsque l'adresse IP du client correspond à la table de correspondances indiquée. Le résultat de la consultation est ignoré, et aucune recherche de sous-réseau n'est effectuée. Ceci peut suivre les tables de correspondances pop-before-smtp .

Exemples :

Valeur rétro-compatible : réécrit toujours les en-têtes de message et ajoute toujours le domaine propriété de Postfix aux adresses incomplètes des en-têtes.

```
local_header_rewrite_clients = static:all
```

Valeur des puristes : réécrit seulement les en-têtes des messages issus de la commande sendmail de Postfix et des messages SMTP de cette machine.

```
local_header_rewrite_clients = permit_mynetworks
```

Documentation de Postfix en français

Valeur intermédiaire : réécrit les adresses des en-têtes et ajoute l'information `$myorigin` ou `$mydomain` seulement aux messages venant de la commande `sendmail` de Postfix, des clients locaux et des clients SMTP autorisés.

Note : ce paramètre n'évitera pas la réécriture des adresses dans les en-têtes lorsque le message d'un client distant est transféré par un système voisin.

```
local_header_rewrite_clients = permit_mynetworks,  
                                permit_sasl_authenticated permit_tls_clientcerts  
                                check_address_map hash:/etc/postfix/pop-before-smtp
```

***local_recipient_maps* (défaut : `proxy:unix:passwd.byname $alias_maps`)**

Tables de correspondances contenant tous les noms ou adresses des destinataires locaux : une adresse de destination est locale lorsque son domaine correspond à `$mydestination`, `$inet_interfaces` ou `$proxy_interfaces`. Indiquez `@domaine` pour les domaines qui n'ont pas de liste de destinataires valide. Techniquement, les tables listées dans `$local_recipient_maps` sont utilisées comme des listes : Postfix n'a besoin que de savoir si la chaîne a été trouvée et ignore le résultat de la consultation.

Si ce paramètre n'est pas vide (défaut), alors le serveur SMTP de Postfix rejettera le courrier des destinataires inconnus localement.

Pour désactiver le contrôle des destinataires locaux dans le serveur SMTP de Postfix, indiquez `"local_recipient_maps ="` (c'est à dire vide).

La valeur par défaut suppose que vous utilisez l'agent local de livraison pour les livraisons locales. Vous devez modifier le paramètre `local_recipient_maps` si :

- ◇ Vous redéfinissez l'agent local de livraison dans `master.cf`.
- ◇ Vous redéfinissez le paramètre `"local_transport"` dans `main.cf`.
- ◇ Vous utilisez les fonctionnalités `"luser_relay"`, `"mailbox_transport"`, ou `"fallback_transport"` de l'agent `local(8)` de livraison.

Plus de détails dans le fichier `LOCAL_RECIPIENT_README`.

Attention : si le serveur SMTP de Postfix fonctionne en cage chroot, vous devez accéder au fichier `passwd` via le service `proxymap(8)`, pour contourner les restrictions. L'alternative consiste à maintenir une copie du fichier système `passwd` dans la cage, ce qui n'est pas pratique.

Exemples :

```
local_recipient_maps =  
local_transport (défaut : local:$myhostname)
```

Le transporteur de messages par défaut pour les domaines qui correspondent à `$mydestination`, `$inet_interfaces` ou `$proxy_interfaces`. Cette information peut être surchargée par la table `transport(5)`.

Par défaut, le courrier local est livré au transporteur nommé "local", qui est le nom d'un service défini dans le fichier `master.cf`.

Indiquez une chaîne sous la forme `transport:nexthop`, où `transport` est le nom du transporteur de messages défini dans `master.cf`. La partie `:nexthop` est optionnelle. Pour plus de détails reportez-vous à la page de manuel `transport(5)`.

Attention : si vous surchargez l'agent local de livraison par défaut, lisez attentivement la page LOCAL RECIPIENT README, sinon, le serveur SMTP risque de rejeter le courrier des destinataires locaux.

***luser_relay* (défaut : vide)**

Destination de collecte optionnelle pour les destinataires locaux inconnus. Par défaut, le courrier des destinataires inconnus dans les domaines correspondants à \$mydestination, \$inet_interfaces ou \$proxy_interfaces est retourné comme non livrable.

Les substitutions suivantes sur \$name sont effectuées par luser_relay:

\$domain

Le domaine du destinataire.

\$extension

The adresse de destination extension.

\$home

Le répertoire personnel du destinataire.

\$local

La partie locale entière de l'adresse de destination.

\$recipient

L'adresse complète de destination.

\$recipient_delimiter

Le délimiteur système de l'extension de adresse de destination.

\$shell

Le nom du shell du login du destinataire.

\$user

Le nom du destinataire.

\${name?value}

Substitue *value* à *\$name* si *\$name* a une valeur non vide.

\${name:value}

Substitue *value* à *\$name* si *\$name* a une valeur vide.

Au lieu de \$name, vous pouvez utiliser \$(name) ou \${name}.

Note : luser_relay ne fonctionne qu'avec l'agent local(8) de livraison de Postfix.

NOTE: si vous utilisez cette fonctionnalité pour des comptes non présent dans le fichier /etc/passwd, vous devez indiquer "local_recipient_maps =" (c'est à dire vide) dans le fichier main.cf, autrement le serveur SMTP de Postfix rejettera tout le courrier des comptes non-UNIX avec la mention "User unknown in local destinataire table".

Exemples :

```
luser_relay = $user@other.host  
luser_relay = $local@other.host  
luser_relay = admin+$local
```

***mail_name* (défaut : Postfix)**

Le nom du système de messagerie qui est affiché dans les en-têtes Received:, dans la bannière d'accueil, et dans les avis de rejet.

***mail_owner* (défaut : postfix)**

Le compte du système UNIX qui possède la file d'attente et la plupart des processus démons de Postfix. Indiquez le nom d'un compte Unix qui ne partage pas de groupe avec d'autres comptes et qui ne possède aucun autre fichier ou processus du système. En particulier, n'utilisez pas nobody ou

daemon. UTILISEZ UN COMPTE ET UN GROUPE DÉDIÉ.

Lorsque la valeur de ce paramètre est modifiée, vous devez relancer "**postfix set-permissions**" (avec Postfix 2.0 et les versions antérieures : "**/etc/postfix/post-install set-permissions**").

mail_release_date (défaut : voir la sortie de "**postconf -d**")

La date de sortie de la version de Postfix, dans le format "YYYYMMDD".

mail_spool_directory (défaut : voir la sortie de "**postconf -d**")

Le répertoire où les boîtes-aux-lettres locales type UNIX sont stockées. La valeur par défaut dépend du système. Indiquez un nom se terminant par / pour des livraisons type maildir.

Note : le livraison maildir est effectuée avec les privilèges du destinataire. Si vous utilisez le paramètre mail_spool_directory pour des livraisons type maildir, vous devez créer le répertoire de plus haut niveau par avance car Postfix ne le créera pas.

Exemples :

```
mail_spool_directory = /var/mail
```

```
mail_spool_directory = /var/spool/mail
```

mail_version (défaut : voir la sortie de "**postconf -d**")

La version du système de messagerie. Les versions stables sont nommées *major.minor.patchlevel*. Les expérimentales incluent également la date d sortie. Cette chaîne peut être utilisée par exemple dans la bannière d'accueil.

mailbox_command (défaut : vide)

Commande externe optionnelle que l'agent de livraison local(8) doit utiliser pour la livraison des messages. La commande est lancée avec les privilèges ID utilisateur et ID du groupe primaire du destinataire. Exception: les livraisons pour root sont exécutées avec les privilèges \$default_privs. Ce n'est pas un problème car :

1. les messages de root devraient toujours être transférés à un utilisateur réel (alias)
2. ne vous loguez pas comme root, utilisez plutôt "su".

Les variables d'environnement suivantes sont exportées à la commande :

CLIENT_ADDRESS

Adresse réseau du client distant. Disponible dans les versions 2.2 et supérieures de Postfix.

CLIENT_HELO

Paramètre de la commande EHLO de client. Disponible dans les versions 2.2 et supérieures de Postfix.

CLIENT_HOSTNAME

Nom de machine du client distant. Disponible dans les versions 2.2 et supérieures de Postfix.

CLIENT_PROTOCOL

Protocole du client distant. Disponible dans les versions 2.2 et supérieures de Postfix.

DOMAIN

La partie "domaine" de l'adresse de destination.

EXTENSION

L'extension optionnelle de l'adresse.

HOME

Le répertoire personnel du destinataire.

LOCAL

La partie locale de l'adresse de destination.

LOGNAME

Le nom du destinataire.

RECIPIENT

L'adresse complète de destination.

SASL_METHOD

Méthode d'authentification SASL indiquée dans la commande AUTH du client distant.
Disponible dans les versions 2.2 et supérieures de Postfix.

SASL_SENDER

Adresse d'expéditeur SASL indiquée dans la commande MAIL FROM du client distant.
Disponible dans les versions 2.2 et supérieures de Postfix.

SASL_USER

Nom d'utilisateur SASL indiqué dans la commande AUTH du client distant. Disponible dans les versions 2.2 et supérieures de Postfix.

SENDER

L'adresse complète d'expédition.

SHELL

Le nom du shell du login du destinataire.

USER

Le nom du destinataire.

Contrairement aux autres paramètres de configuration de Postfix, le paramètre mailbox_command n'est pas sujet aux substitutions \$name afin de rendre plus simple l'écriture de shell (voir l'exemple ci-dessous).

Si vous pouvez, évitez les méta caractères du shell car ils forceront Postfix à lancer un processus shell coûteux. Si vous livrez via Procmail, lancer un shell n'aura pas de différence de coût notable dans le coût total.

Note : si vous utilisez la fonctionnalité mailbox_command pour livrer tout le système de messagerie, vous devez renseigner un alias pour transférer le courrier de root à un utilisateur réel.

L'ordre de priorité des fonctionnalités de livraison locale est : alias, fichiers .forward, mailbox transport maps, mailbox transport, mailbox command maps, mailbox command, home mailbox, mail spool directory, fallback transport maps, fallback transport et user relay.

Exemples :

```
mailbox_command = /some/where/procmail
mailbox_command = /some/where/procmail -a "$EXTENSION"
mailbox_command = /some/where/maildrop -d "$USER"
                  -f "$SENDER" "$EXTENSION"
```

mailbox_command_maps (défaut : vide)

Tables de correspondances optionnelles des commandes externes à utiliser par destinataire pour la livraison dans les boîtes-aux-lettres locales. Surcharge mailbox_command.

L'ordre de priorité des fonctionnalités de livraison locale est : aliases, fichiers .forward, mailbox transport, mailbox transport, mailbox command maps, mailbox command, home mailbox, mail spool directory, fallback transport maps, fallback transport mailbox command maps, fallback transport et user relay.

mailbox_delivery_lock (défaut : voir la sortie de "postconf -d")

Comment verrouiller les boîtes-aux-lettres locales type Unix avant de tenter la livraison. Pour voir la liste des méthodes de verrouillage, utilisez la commande "**postconf -l**".

Ce paramètre est ignoré avec les livraisons style **maildir**, car de telles livraisons sont sûres sans

verrouillage explicite.

Note : la méthode **dotlock** nécessite que l'UID ou le GID du destinataire ait le droit d'écrire dans le répertoire parent du fichier boîte-aux-lettres.

Note : la valeur par défaut de ce paramètre dépend du système.

mailbox_size_limit (défaut : 51200000)

La taille maximale des fichiers boîtes-aux-lettres locaux ou zéro (pas de limite). En pratique, ceci limite la taille de tout fichier écrit pour livrer localement, y compris les fichiers écrits par des commandes externes exécutées par l'agent de livraison local(8).

Cette limite ne doit pas être inférieure à la taille limite des messages.

mailbox_transport (défaut : vide)

Transporteur optionnel que l'agent de livraison local(8) utilise pour livrer le courrier des destinataires locaux, qu'ils soient trouvés ou non dans la base de données passwd d'UNIX.

L'ordre de priorité des fonctionnalités de livraison locale est : alias, fichiers .forward, mailbox transport, mailbox transport, mailbox command maps, mailbox command, home mailbox, mail spool directory, fallback transport maps, fallback transport mailbox command maps, fallback transport et luser relay.

mailbox_transport_maps (défaut : vide)

Tables optionnelles de correspondances entre destinataire et transport à utiliser pour la livraison dans la boîte-aux-lettres locale, que le destinataire soit trouvé ou non dans la base de données des mots-de-passe UNIX.

Le choix de la livraison locale se fait dans l'ordre suivant : les alias, les fichiers .forward, mailbox transport maps, mailbox transport, mailbox command maps, mailbox command, home mailbox, mail spool directory, fallback transport maps, fallback transport et luser relay.

Pour des raisons de sécurité, cette fonctionnalité n'autorise pas les substitutions \$nombre dans les tables d'expressions régulières.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

mailq_path (défaut : voir la sortie de "postconf -d")

Fonctionnalité compatible Sendmail qui indique où est installée la commande Postfix mailq(1). Cette commande peut être utilisée pour afficher la file d'attente.

manpage_directory (défaut : voir la sortie de "postconf -d")

Emplacement des pages de manuel de Postfix.

maps_rbl_domains (défaut : vide)

Paramètre obsolète : utilisez plutôt reject_rbl_client.

maps_rbl_reject_code (défaut : 554)

Le code numérique de réponse du serveur SMTP de Postfix lorsqu'une requête d'un client SMTP distant est bloquée par une restriction reject_rbl_client, reject_rhsbl_client, reject_rhsbl_sender ou reject_rhsbl_recipient.

Ne changez pas ceci avant d'avoir bien compris la RFC 821.

masquerade_classes (défaut : envelope_sender, header_sender, header_recipient)

Adresses soumises au masquage.

Par défaut, le masquage d'adresse est limité aux adresses de destination dans l'enveloppe, et aux adresses de destination et de l'expéditeur dans les en-têtes. Ceci vous permet d'utiliser le masquage

Documentation de Postfix en français

d'adresse sur une passerelle en conservant la possibilité de transférer le courrier des utilisateurs sur des machines individuelles.

Indiquez zéro ou plusieurs éléments parmi: `envelope_sender`, `envelope_recipient`, `header_sender`, `header_recipient`

masquerade_domains (défaut : vide)

Liste optionnelle de domaines dont la structure des sous-domaines sera masquée dans les adresses.

Cette liste est lue de gauche à droite et le processus s'arrête dès qu'une entrée correspond. Ainsi,

```
masquerade_domains = foo.exemple.com exemple.com
```

remplace "user@any.thing.foo.exemple.com" en "user@foo.exemple.com", mais remplace "user@any.thing.else.exemple.com" par "user@exemple.com".

Un nom de domaine prefixé par ! signifie qu'il ne faut pas masquer ses sous-domaines. Ainsi,

```
masquerade_domains = !foo.exemple.com exemple.com
```

ne remplace pas "user@any.thing.foo.exemple.com" ou "user@foo.exemple.com", mais remplace "user@any.thing.else.exemple.com" to "user@exemple.com".

Note 2 : avec les versions 2.2 et supérieures de Postfix, le masquage des adresses dans les en-têtes de message ne se produit que lorsque l'une des conditions suivantes est réalisée :

- ◇ le message est soumis par la commande `sendmail(1)` de Postfix,
- ◇ le message provient d'un client réseau qui correspond à `$local_header_rewrite_clients`,
- ◇ le message provient du réseau et le paramètre `remote_header_rewrite_domain` contient une valeur non vide.

Pour retrouver le comportement des versions de Postfix antérieures à la 2.2, utilisez "`local_header_rewrite_clients = static:all`".

Exemple :

```
masquerade_domains = $mydomain
```

masquerade_exceptions (défaut : vide)

Liste optionnelle de noms d'utilisateurs non soumis au masquage d'adresse, même lorsque leurs adresses correspondent à `$masquerade_domains`.

par défaut, address masquage d'adresse ne fait pas d'exceptions.

Indiquez une liste de noms d'utilisateurs, "/nom/de/fichier" ou d'expressions "`type:table`" séparé par des virgules et/ou des espaces. La liste est examinée de gauche à droite, et la recherche s'arrête dès la première occurrence correspondante. Indiquez "!"name" pour explorer un nom de la liste. Un "/nom/de/fichier" de correspondances est remplacé par son contenu; une table de correspondances "`type:table`" correspond lorsqu'un nom correspond à une clef (le résultat de la consultation est ignoré). Continuez les lignes longues en commençant la ligne suivante par des espaces.

Exemples :

```
masquerade_exceptions = root, mailer-daemon  
masquerade_exceptions = root
```


max_idle (défaut : 100s)

Temps maximum d'inactivité d'un processus démon de Postfix entre deux requêtes. Passé ce délai, le démon s'arrête. Ce paramètre est ignoré par le gestionnaire des files d'attente.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

max_use (défaut : 100)

Nombre maximum de requêtes de connexion pour un processus démon de Postfix. Au delà, le démon s'arrête. Ce paramètre est ignoré par le gestionnaire des files d'attente et les autres processus démons longue durée de Postfix.

maximal_backoff_time (défaut : 4000s)

Temps maximal entre deux tentatives de livraison d'un message retardé.

Ce paramètre devrait être mis à une valeur supérieure ou égale à \$minimal_backoff_time. Voir aussi \$queue_run_delay.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

maximal_queue_lifetime (défaut : 5d)

Temps maximal de présence dans la file d'attente avant rejet.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est d (days).

Indiquez 0 si la livraison ne doit être tentée qu'une fois.

message_reject_characters (défaut : vide)

L'ensemble des caractères que Postfix doit rejeter dans le contenu des messages. Les séquences d'échappement type C habituelles sont reconnues \a \b \f \n \r \t \v \nnn (jusqu'à trois chiffre octaux) et \\.

Exemple :

```
message_reject_characters = \0
```

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

message_size_limit (défaut : 10240000)

Taille maximale d'un message en octets, y compris les informations de l'enveloppe.

message_strip_characters (défaut : vide)

L'ensemble des caractères que Postfix doit effacer dans le contenu des messages. Les séquences d'échappement type C habituelles sont reconnues \a \b \f \n \r \t \v \nnn (jusqu'à trois chiffre octaux) et \\.

Exemple :

```
message_strip_characters = \0
```

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

mlter_command_timeout (défaut : 10s)

Temps limite pour l'envoi d'une commande SMTP à une application Milter (mail filter), et pour recevoir la réponse milter.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

militer_connect_macros (défaut : see postconf -n output)

Macros envoyées aux applications Militer (mail filter) à la fin d'une connexion SMTP. Voir [MILTER_README](#) pour la liste des noms de macros et leur significations.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

militer_connect_timeout (défaut : 10s)

Temps limite pour se connecter à une application Militer (mail filter) et pour négocier les options de protocole.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

militer_content_timeout (défaut : 100s)

Temps limite pour envoyer le sending contenu du message à une application Militer (mail filter), et pour recevoir la réponse Militer.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

militer_data_macros (défaut : see postconf -n output)

Macros envoyées envoyées aux applications Militer (mail filter) version 4 et au-dessus après la commande SMTP DATA. Voir [MILTER_README](#) pour la liste des macros disponibles et leur significations.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

militer_default_action (défaut : tempfail)

Action par défaut lorsqu'une application Militer (mail filter) n'est pas disponible ou mal configurée. Indiquez l'une des propositions suivantes :

accept

Procède comme si le filtre de message n'existait pas.

reject

Rejette toutes les autres commandes dans cette session avec un code de statut permanent.

tempfail

Rejette toutes les autres commandes dans cette session avec un code de statut temporaire.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

militer_end_of_data_macros (défaut : see postconf -n output)

Macros envoyées aux applications Militer (mail filter) après la fin des données du message. Voir [MILTER_README](#) pour la liste des noms de macros disponibles et leur significations..

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

militer_helo_macros (défaut : see postconf -n output)

Macros envoyées aux applications Militer (mail filter) après les commandes SMTP HELO ou EHLO. Voir [MILTER_README](#) pour la liste des macros disponibles et leur significations.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

militer_macro_daemon_name (défaut : \$var_myhostname)

La valeur de la macro {daemon_name} envoyée aux applications Militer (mail filter). See [MILTER_README](#) pour la liste des noms de macros et leurs significations.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

militer_macro_v (défaut : \$mail_name \$mail_version)

La valeur de la macro {v} envoyée aux applications Militer (mail filter). Voir [MILTER_README](#) pour la liste des noms de macros et leurs significations.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

mlter_mail_macros (défaut : see postconf -n output)

Macros envoyées aux applications Milter (mail filter) après la commande SMTP MAIL FROM. Voir MILTER_README pour la liste des macros disponibles et leur significations.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

mlter_protocol (défaut : 2)

La version du protocole Milter (mail filter) et les optionnelles extensions de protocole utilisées pour communiquer avec les applications Milters (mail filter). Cette information doit correspondre au protocole attendu par l'application de filtrage utilisée.

Versions de protocole :

2

utilise la version 2 du protocole Milter de Sendmail 8.

3

utilise la version 3 du protocole Milter de Sendmail 8.

4

utilise la version 4 du protocole Milter de Sendmail 8.

Extensions du protocole :

no_header_reply

Utilisez ceci lorsque l'application Milter ne doit pas répondre à chaque en-tête de message.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

mlter_rcpt_macros (défaut : see postconf -n output)

Macros envoyées aux applications Milter (mail filter) after the SMTP RCPT TO command. See MILTER_README pour la liste des macros disponibles et leur significations.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

mlter_unknown_command_macros (défaut : see postconf -n output)

Macros envoyées aux version 3 or higher applications Milter (mail filter) after an unknown SMTP command. See MILTER_README pour la liste des macros disponibles et leur significations.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

mime_boundary_length_limit (défaut : 2048)

Longueur maximale des chaînes de caractère MIME multipart. Le processeur MIME ne peut distinguer les chaînes de caractère qui ne diffèrent pas sur les \$mime_boundary_length_limit premiers caractères.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

mime_header_checks (défaut : \$header_checks)

Tables de correspondances optionnelles pour l'inspection du contenu des en-têtes de message relatives à MIME (voir la page de manuel header_checks(5)).

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

mime_nesting_limit (défaut : 100)

Le niveau maximal d'emboîtement de messages multiparties que le processuer MIME manipulera. Postfix refuse le courrier plus imbriqué.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

minimal_backoff_time (défaut : 1000s)

Documentation de Postfix en français

Temps minimum entre deux tentatives de livraison d'un message retardé. Ce paramètre limite également le temps où le status "non-joignable" d'une destination est conservé dans le cache mémoire.

Ce paramètre devrait être mis à une valeur supérieure ou égale à \$queue_run_delay. Voir également \$maximal_backoff_time.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

multi_recipient_bounce_reject_code (défaut : 550)

Le code numérique de réponse du serveur SMTP de Postfix lorsqu'une requête d'un client SMTP est bloquée par la restriction reject_multi_recipient_bounce.

Ne changez pas ceci avant d'avoir bien compris la RFC 821.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

mydestination (défaut : \$myhostname, localhost.\$mydomain, localhost)

Liste des domaines livrés par le transporteur de messages \$local_transport. Par défaut c'est l'agent local(8) de livraison qui recherche les destinataires dans /etc/passwd et /etc/aliases. Le serveur SMTP valide les adresses de destination avec \$local_recipient_maps et rejette les destinataires inconnus. Voyez également la classe local domain dans le fichier ADDRESS CLASS README.

La valeur de mydestination par défaut ne correspond qu'à la machine hôte. Sur une passerelle de messagerie d'un domaine, vous devrez également inclure \$mydomain.

La méthode de livraison \$local_transport est également sélectionnée pour le courrier adressé à utilisateur@[l.adresse.IP] du système de messagerie (toutes les adresses IP indiquées dans inet_interfaces et proxy_interfaces).

Attention :

- ◇ N'indiquez pas les noms des domaines virtuels – ils sont indiqués ailleurs. Voyez la page VIRTUAL README pour plus d'information.
- ◇ N'indiquez pas les noms des domaines dont votre machine est un MX de secours. Reportez-vous à la page STANDARD CONFIGURATION README pour la mise en œuvre de secours MX.
- ◇ Par défaut, le serveur SMTP de Postfix rejette le courrier des destinataires non listés avec le paramètre local_recipient_maps. Pour plus d'information, consultez les paragraphes local_recipient_maps et unknown_local_recipient_reject_code.

Indiquez une liste de host or noms de domaine, "/nom/de/fichier" or expressions "type:table", séparé par des virgules et/ou des espaces. Un "/nom/de/fichier" de correspondances est remplacé par son contenu; une table de correspondances "type:table" correspond lorsqu'une de ses clefs correspond au nom (le résultat de la consultation est ignoré). Continuez les lignes longues en commençant la ligne suivante par des espaces.

Exemples :

```
mydestination = $myhostname, localhost.$mydomain $mydomain
mydestination = $myhostname, localhost.$mydomain www.$mydomain, ftp.$mydomain
```

mydomain (défaut : voir la sortie de "postconf -d")

Documentation de Postfix en français

Le nom de domaine Internet de ce système de messagerie. Par défaut, ce paramètre vaut use \$myhostname oté de son premier composant. \$mydomain est utilisé comme valeur par défaut pour beaucoup de paramètres de configuration.

Exemple :

```
mydomain = domain.tld
```

myhostname (défaut : voir la sortie de "postconf -d")

Le nom de machine Internet de ce système de messagerie. Par défaut, il vaut le résultat de gethostname() au format pleinement. \$myhostname est utilisé comme valeur par défaut pour beaucoup d'autres paramètres de configuration.

Exemple :

```
myhostname = host.domain.tld
```

mynetworks (défaut : voir la sortie de "postconf -d")

La liste des clients SMTP "internes" qui ont plus de privilèges que les "étrangers".

En particulier, ces clients SMTP internes sont autorisés à relayer du courrier via Postfix. Voyez le paragraphe du paramètre smtpd_recipient_restrictions

Vous pouvez indiquer la liste des adresses réseaux autorisées à la main ou laisser Postfix le faire (valeur par défaut. Voyez la description du paramètre mynetworks_style pour plus d'information.

Autrement, vous pouvez renseigner mynetworks à la main auquel cas Postfix ignore le paramètre mynetworks_style.

Indiquez une liste d'expressions réseau/masque, séparé par des virgules et/ou des espaces. Continuez les lignes longues en commençant la ligne suivante par des espaces.

Le masque indique le nombre de bits de la partie réseau de l'adresse. Vous pouvez également indiquer des "/nom/de/fichiers" ou des expressions "type:table". Un "/nom/de/fichier" de correspondances est remplacé par son contenu; une table de correspondances "type:table" correspond lorsqu'une de ses clefs correspond (le résultat de la consultation est ignoré).

La recherche est effectuée de gauche à droite et s'arrête à la première correspondance. Indiquez "!expression" pour exclure une adresse ou un sous-réseau de la liste.

Note : les adresses IP version 6 doivent être indiquées entre [] dans mynetworks et dans les fichier indiqués avec "/nom/de/fichier". Les adresses IP version 6 contiennent le caractère ":" et pourraient être confondues avec une expression "type:table".

Exemples :

```
mynetworks = 168.100.189.0/28, 127.0.0.0/8
```

```
mynetworks = !192.168.0.1, 192.168.0.0/28
```

```
mynetworks = $config_directory/mynetworks
```

```
mynetworks = hash:/etc/postfix/network_table
```

mynetworks_style (défaut : subnet)

La méthode pour générer la valeur par défaut pour le paramètre mynetworks. C'est la liste des réseaux internes autorisés à relayer le courrier.

- ◇ Indiquez "mynetworks_style = host" lorsque Postfix ne doit accepter que la machine locale.
- ◇ Indiquez "mynetworks_style = subnet" lorsque Postfix doit accepter les clients SMTP raccordés aux mêmes sous-réseaux que la machine hôte. Sur Linux, cela ne fonctionne correctement qu'avec les interfaces retournées par la commande "ifconfig".
- ◇ Indiquez "mynetworks_style = class" lorsque Postfix doit accepter les clients SMTP raccordés sur la même classe d'adresses IP A/B/C que la machine hôte. A ne pas utiliser sur un site raccordé par modem – cela revient à valider toutes les machines de votre fournisseur d'accès. Préférez renseigner mynetworks à la main, comme décrit au paragraphe concernat le paramètre de configuration mynetworks.

myorigin (défaut : \$myhostname)

Le domaine par défaut utilisé pour les messages postés localement. La valeur par défaut de \$myhostname, est adéquate pour les petits sites. Si vous contrôlez un domaine avec de multiples machines, vous devrez changer cela en \$mydomain et (2) mettre en place une base d'alias globale qui référencera tous les utilisateur@cette.machine.

Exemple :

myorigin = \$mydomain

nested_header_checks (défaut : \$header_checks)

Tables de correspondances optionnelles pour l'inspection du contenu des en-têtes de message non MIME dans les messages attachés, tel que décrit dans la page de manuel header_checks(5).

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

newaliases_path (défaut : voir la sortie de "postconf -d")

Fonctionnalité compatible Sendmail qui indique l'emplacement de la commande newaliases(1). Cette commande peut être utilisé pour reconstruire la base de données locale alias.

non_fqdn_reject_code (défaut : 504)

Le code de réponse numérique que renvoie le serveur SMTP de Postfix lorsqu'une requête d'un client est rejeté par une restriction reject_non_fqdn_hostname, reject_non_fqdn_sender ou reject_non_fqdn_recipient.

notify_classes (défaut : resource, software)

La liste des classes d'erreur qui doivent être rapportées au postmaster. Par défaut, seuls les problèmes graves sont rapportés. Les paranoïaques pourront recevoir les rapports d'erreur de protocole (logiciels bogués) ou de rejet.

Les classes d'erreur sont :

bounce (implique également 2bounce)

Envoie au postmaster les copies des en-têtes des messages rejetés et et les retranscriptions des sessions SMTP des messages rejetés à la livraison. La notification est envoyée à l'adresse renseignée dans le paramètre de configuration bounce_notice_recipient (défaut : postmaster)

2bounce

Envoie au postmaster les avis de rejet non-livrables. La notification est envoyée à l'adresse renseignée dans le paramètre de configuration 2bounce_notice_recipient (défaut : postmaster).

delay

Envoie au postmaster les copies des en-têtes des messages retardés. La notification est envoyée) l'adresse renseignée dans le paramètre de configuration delay_notice_recipient (défaut : postmaster).

policy

Documentation de Postfix en français

Envoie au postmaster une retranscription des sessions SMTP lorsqu'une requête client est rejetée à cause d'une restriction. La notification est envoyée à l'adresse renseignée dans le paramètre de configuration error_notice_recipient (défaut : postmaster).

protocol

Envoie au postmaster une retranscription des sessions SMTP en cas d'erreur de protocole du client ou du serveur. La notification est envoyée à l'adresse renseignée dans le paramètre de configuration error_notice_recipient (défaut : postmaster).

resource

Informe le postmaster lorsqu'un courrier est retardé à cause d'un problème de ressources. La notification est envoyée à l'adresse renseignée dans le paramètre de configuration error_notice_recipient (défaut : postmaster).

software

Informe le postmaster lorsqu'un message est retardé à cause d'un problème logiciel. La notification est envoyée à l'adresse renseignée dans le paramètre de configuration error_notice_recipient (défaut : postmaster).

Exemples :

```
notify_classes = bounce, delay, policy, protocol, resource, software  
notify_classes = 2bounce, resource, software
```

owner_request_special (défaut : yes)

Applique un traitement particulier aux parties locales des adresses de listes de propriétaires ou de requêtes : ne découpe pas de telles adresses lorsque recipient_delimiter est mis à "-". Cette fonctionnalité est pratiquée pour les listes de diffusion.

parent_domain_matches_subdomains (défaut : voir la sortie de "postconf -d")

Quelles fonctionnalités de Postfix traitent les sous-domaines comme leur domaine parent au lieu de requérir explicitement un ".domaine.com" (premier caractère "."). C'est utilisé pour des raisons de compatibilité: éventuellement, toutes les fonctionnalités de Postfix peuvent requérir explicitement des expressions ".domaine.com" lorsque vous voulez détailler les sous-domaines.

permit_mx_backup_networks (défaut : vide)

Restreint l'utilisation de la fonctionnalité d'accès SMTP permit_mx_backup aux seuls domaines dont l'hôte MX primaire est listé ici.

pickup_service_name (défaut : pickup)

Le nom du service pickup(8). Ce service prélève les soumissions locales de la file d'attente maildrop de Postfix.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

plaintext_reject_code (défaut : 450)

Le code numérique de réponse du serveur SMTP de Postfix lorsqu'une requête est rejetée par la restriction reject_plaintext_session.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

prepend_delivered_header (défaut : command, file, forward)

Le contexte de livraison de message dans lequel l'agent local(8) de livraison de Postfix pour ajouter un en-tête de message Delivered-to:.

Par défaut, l'agent local de livraison de Postfix ajoute un en-tête Delivered-To: lorsque le message est transféré ou lorsqu'il est livré dans un fichier (boîte-aux-lettres) ou à une commande. Désactivez l'inscription de l'en-tête Delivered-To: lorsque le transfert n'est pas recommandé.

Indiquez zéro ou plusieurs mention parmi **forward**, **file**, or **command**.

Exemple :

```
prepend_delivered_header = forward
```

process_id (lecture seule)

L'ID du processus d'une commande ou d'un démon de Postfix.

process_id_directory (défaut : pid)

L'emplacement des fichiers PID de Postfix relativement au répertoire \$queue_directory. C'est un paramètre en lecture seule.

process_name (lecture seule)

Le nom du processus d'une commande ou d'un démon Postfix.

propagate_unmatched_extensions (défaut : canonical, virtual)

Quelles tables de correspondances d'adresses copient une extension d'adresse de la clef de recherche vers le résultat de la consultation.

Par exemple, avec une correspondance virtuelle(5) "joe@domain -> joe.user", l'adresse "joe+foo@domain" sera réécrite en "joe.user+foo".

Indiquez zéro ou plus de mentions parmi **canonical**, **virtual**, **alias**, **forward** et **include**. Cela provoque la propagation des extensions d'adresse respectivement avec les tables canonical(5), virtual(5) et aliases(5), et avec les fichiers locaux.forward and locaux.include.

Note : activer cette fonctionnalité pour d'autres types que **canonical** et **virtual** risque de provoquer des problèmes lorsque les messages sont transférés à d'autres sites, en particulier avec le courrier transmis à une liste de diffusion.

Exemples :

```
propagate_unmatched_extensions = canonical, virtual, alias,  
                                forward, include
```

```
propagate_unmatched_extensions = canonical, virtual
```

proxy_interfaces (défaut : vide)

Les adresses réseau pour lesquelles ce système de messagerie reçoit du courrier par le biais d'un proxy ou d'un traducteur d'adresse.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

Vous devez indiquer l'adresse "outside" de votre proxy/NAT lorsque votre système est une machine MX de secours d'un autre domaine, sinon la livraison de message bouclera lorsque l'hôte primaire ne fonctionnera pas.

Exemple :

```
proxy_interfaces = 1.2.3.4
```

proxy_read_maps (défaut : voir la sortie de "postconf -d")

Les tables de correspondances que le serveur proxymap(8) est autorisé à lire. Les références qui ne commencent pas par proxy : sont ignorées. Les accès à la table proxymap(8) sont en lecture seule.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

qmgr_clog_warn_time (défaut : 300s)

Le délai minimal entre les avertissements indiquant qu'une destination spécifique encombre la file d'attente active de Postfix. Indiquez 0 pour le désactiver.

Documentation de Postfix en français

Cette fonctionnalité est activée avec le paramètre helpful_warnings.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

qmgr_fudge_factor (défaut : 100)

Fonctionnalité obsolète : le pourcentage de ressources de livraison qu'un système de messagerie chargé utilisera pour la livraison d'un message à une grande liste de diffusion.

Cette fonctionnalité existe seulement dans l'ancien gestionnaire des files d'attentes oqmgr(8). Le gestionnaire actuel résout ce problème de meilleure manière.

qmgr_message_active_limit (défaut : 20000)

Nombre maximum de messages dans la file d'attente active.

qmgr_message_recipient_limit (défaut : 20000)

Nombre maximum de destinataires en mémoire dans le gestionnaire des files d'attente de Postfix, et taille maximale du cache de statut des destinations "mortes" dans la mémoire rapide.

qmgr_message_recipient_minimum (défaut : 10)

Nombre minimal de destinataires en mémoire pour chaque message. Ceci prend la priorité sur toute limite de destinataire en mémoire (c'est à dire le qmgr_message_recipient_limit global et le <transport>_recipient_limit) si nécessaire. La valeur minimum pour ce paramètre est 1.

qmqpd_authorized_clients (défaut : vide)

Clients autorisés à se connecter au port du serveur QMQP.

Par défaut, aucun client n'est autorisé à utiliser ce service car le serveur QMQP accepte le relai vers toutes les destination.

Indiquez une liste d'expression indiquant des noms de machines, de domaines, des expressions réseau/masque des adresses Internet. Lorsqu'une expression indique un nom de fichier, son contenu le remplace; lorsque l'expression indique "type:table", la table de correspondance est utilisée à la place.

Les expressions sont séparées par des espaces et/ou des virgules. Pour inverser le résultat, faites précéder une expression ne désignant pas un fichier par un point d'exclamation (!).

Exemple :

```
qmqpd_authorized_clients = !192.168.0.1, 192.168.0.0/24
```

qmqpd_error_delay (défaut : 1s)

Temps pendant lequel le serveur QMQP attendra avant de renvoyer une réponse négative au client. Le but est de ralentir les clients mal intentionnés.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

qmqpd_timeout (défaut : 300s)

Temps limite pour envoyer ou recevoir des informations sur le réseau. Si une opération de lecture ou d'écriture est bloquée pendant plus de \$qmqpd_timeout secondes le serveur QMQP se déconnecte.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

queue_directory (défaut : voir la sortie de "postconf -d")

L'emplacement du répertoire racine de la file d'attente de Postfix. C'est le répertoire racine des processus démons de Postfix qui fonctionnent en cage chroot.

queue_file_attribute_count_limit (défaut : 100)

Nombre maximum d'attribut (name=value) qui peuvent être stockés dans un fichier de la file d'attente de Postfix. Cette limite est utilisée par le serveur cleanup(8).

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

queue_minfree (défaut : 0)

Espace minimum d'espace libre en octet dans le système de fichiers qui est utilisé pour recevoir le courrier. C'est utilisé par le serveur SMTP pour décider s'il doit accepter du courrier.

Par défaut, le serveur SMTP de Postfix 2.1 rejette les commandes MAIL FROM lorsque la quantité d'espace libre est inférieure à $1.5 * \$\text{message_size_limit}$. Pour indiquer un minimum d'espace plus élevé, renseignez ce paramètre avec une valeur au moins égale à $1.5 * \$\text{message_size_limit}$.

Avec les versions 2.0 et antérieures de Postfix, une valeur queue_minfree nulle signifie qu'il n'y a pas de minimum d'espace libre.

queue_run_delay (défaut : 1000s)

Délai entre deux examens de la file d'attente retardée par le gestionnaire des files d'attente.

Ce paramètre devrait être fixé à une valeur inférieure ou égale à \$minimal_backoff_time. Voir aussi \$maximal_backoff_time.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

queue_service_name (défaut : qmgr)

Le nom du service qmgr(8). Ce service pilote la file d'attente et ordonne les requêtes de livraison.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

rbl_reply_maps (défaut : vide)

Tables de correspondances optionnelles contenant les modèles de réponse RBL. Les tables sont indexées par le nom de domaine RBL. Par défaut, Postfix utilise le modèle par défaut indiqué avec le paramètre de configuration default_rbl_reply. Reportez-vous à ce paragraphe pour la syntaxe des modèles de réponse RBL.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

readme_directory (défaut : voir la sortie de "postconf -d")

Emplacement des fichiers README de Postfix qui décrivent comment compiler configurer ou opérer un sous-système ou une fonctionnalité spécifique de Postfix.

receive_override_options (défaut : vide)

Active ou désactive la validation des destinataires, le filtrage de contenu intégré ou les correspondances d'adresse. Typiquement, elles sont indiquées dans le fichier master.cf comme argument de ligne de commande des démons smtpd(8), qmqpd(8) ou pickup(8).

Indiquez zéro ou plus options parmi les suivantes. Ces options surchargent les paramètres du fichier main.cf et sont implémentés soit par smtpd(8), qmqpd(8), ou pickup(8) eux-mêmes ou transmises au serveur cleanup(8).

no_unknown_recipient_checks

Ne pas rejeter les destinataires inconnus (serveur SMTP seulement). C'est typiquement utilisé APRÈS un filtre externe de contenu.

no_address_mappings

Désactive les traductions d'adresses canoniques, les substitutions d'alias virtuels et les copies cachées automatiques (BCC: blind carbon-copy). C'est typiquement utilisé AVANT un filtre

externe de contenu.

no_header_body_checks

Désactive l'examen des en-têtes/du corps (header/body_checks). C'est typiquement utilisé APRÈS un filtre externe de contenu.

Note : lorsque le paramètre "AVANT filtrage de contenu" receive_override_options est indiqué dans le fichier main.cf, indiquez les paramètres "APRÈS le filtrage" receive_override_options dans le fichier mastre.cf (et vice versa).

Exemples :

```
receive_override_options =  
    no_unknown_recipient_checks, no_header_body_checks  
receive_override_options = no_address_mappings
```

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

recipient_bcc_maps (défaut : vide)

Table de correspondances optionnelle de copies cachées (BCC: blind carbon-copy), indexée par adresse de destination. L'adresse BCC (les résultats multiples ne sont pas supportés) est ajoutée lorsque le message arrive de l'extérieur de Postfix.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

L'ordre de recherche dans les tables est le suivant :

- ◇ Recherche l'adresse "utilisateur+extension@domaine.tld" en incluant l'extension optionnelle de destination.
- ◇ Recherche l'adresse "utilisateur@domain.tld" otée de son éventuelle extension.
- ◇ Recherche la partie locale de l'adresse "utilisateur+extension" lorsque le domaine est listé dans \$myorigin, \$mydestination, \$inet_interfaces ou \$proxy_interfaces.
- ◇ Recherche la partie locale de l'adresse "utilisateur" lorsque le domaine est listé dans \$myorigin, \$mydestination, \$inet_interfaces ou \$proxy_interfaces.
- ◇ Recherche la partie "@domaine.tld".

Indiquez les types et noms des bases de données à utiliser. Après chaque changement lancez "postmap /etc/postfix/recipient_bcc".

NOTE : si le courrier envoyé à l'adresse BCC est rejeté, il sera retourné à l'expéditeur.

NOTE : les copies cachées automatiques ne sont produites que pour les nouveaux messages. Pour éviter les boucles de messages, elles ne sont pas générées pour les messages que Postfix transfère en interne ni pour les messages générés par Postfix lui-même.

Exemple :

```
recipient_bcc_maps = hash:/etc/postfix/recipient_bcc
```

recipient_canonical_classes (défaut : envelope_recipient, header_recipient)

Adresses sujettes aux réécritures d'adresses recipient_canonical_maps. Par défaut, les réécritures d'adresses recipient_canonical_maps sont appliquées aux adresses de destination de l'enveloppe et des en-têtes.

Indiquez une ou plusieurs propositions parmi : envelope_recipient, header_recipient

Cette fonctionnalité est disponible dans les versions 2.2 et supérieures de Postfix.

recipient_canonical_maps (défaut : vide)

Tables de correspondances d'adresse optionnelles pour les adresses de destination de l'enveloppe et des en-têtes. Les formats de ces tables sont documentés à la page [canonical\(5\)](#).

Note : `$recipient_canonical_maps` est examinée avant `$canonical_maps`.

Exemple :

```
recipient_canonical_maps = hash:/etc/postfix/recipient_canonical
```

recipient_delimiter (défaut : vide)

Le séparateur entre noms d'utilisateurs et extensions d'adresse (user+foo). Reportez-vous aux pages [canonical\(5\)](#), [local\(8\)](#), [relocated\(5\)](#) et [virtual\(5\)](#) pour les effets de ce paramètre sur les fichiers alias, canonical, virtual, relocated et .forward. Par défaut, le logiciel essaie user+foo et .forward+foo avant d'essayer user et .forward.

Exemple :

```
recipient_delimiter = +
```

reject_code (défaut : 554)

Le code numérique de réponse du serveur SMTP de Postfix lorsqu'une requête d'un client SMTP distant est rejetée par la restriction "reject".

Ne changez pas ceci avant d'avoir bien compris la [RFC 821](#).

relay_clientcerts (défaut : vide)

Liste des certificats de clients SMTP extérieurs pour lesquels le serveur SMTP de Postfix autorise l'accès avec la fonctionnalité `permit_tls_clientcerts`. Cette fonctionnalité n'utilise pas les noms des certificats car Les routines de manipulation de liste de Postfix ont un traitement particulier pour les espace et certains autres caractères. À la place, nous utilisons les empreintes des certificats car elles sont difficile à falsifier mais faciles à utiliser dans les consultations.

Les tables de correspondances de Postfix ont la forme de paires (clef, valeur). Comme nous n'avons besoin que de la clef, la valeur peut être choisie librement, par exemple le nom de l'utilisateur ou de la machine :

```
D7:04:2F:A7:0B:8C:A5:21:FA:31:77:E1:41:8A:EE:80 lutzpc.at.home
```

Exemple :

```
relay_clientcerts = hash:/etc/postfix/relay_clientcerts
```

Pour un contrôle plus fin, utilisez `check_ccert_access` pour sélectionner une politique d'accès appropriée pour chaque client. Voir [RESTRICTION CLASS README](#).

Cette fonctionnalité est disponible à partir de la version 2.2 de Postfix.

relay_destination_concurrency_limit (défaut : \$default_destination_concurrency_limit)

Nombre maximum de livraisons parallèles vers la même destination via le transporteur de messages "relay". Cette limite est utilisée par le gestionnaire des files d'attente. Le nom du transporteur de message correspond au premier champ du fichier `master.cf`.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

relay_destination_recipient_limit (défaut : \$default_destination_recipient_limit)

Nombre maximum de destinataires par livraison via le transporteur de messages "relay". Cette limite est utilisée par le gestionnaire des files d'attente. Le nom du transporteur de message correspond au premier champ du fichier master.cf.

Mettre ce paramètre à 1 change le sens de relay_destination_concurrency_limit de concurrence par domaine en concurrence par destinataire.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

relay_domains (défaut : \$mydestination)

Domaines de destination (et leurs sous-domaines) que ce système acceptera de relayer. Les correspondances des sous-domaines sont contrôlées par le paramètre parent_domain_matches_subdomains. Pour plus de détails sur l'emploi de relay_domains, voyez la description des restrictions sur les destinataires SMTP permit_auth_destination et reject_unauth_destination.

Les domaines qui correspondent à \$relay_domains sont livrés par le transporteur de messages \$relay_transport. Le serveur SMTP valide les adresses de destination avec \$relay_recipient_maps et rejette les destinataires inconnus. Voyez également la classe d'adresse relay_domains de la page ADDRESS CLASS README.

NOTE: Postfix ne transférera pas automatiquement les messages pour les domaines qui indiquent ce système comme machine MX primaire ou de secours. Reportez-vous au paragraphe permit_mx_backup.

Indiquez une liste de machines ou de noms de domaine, des expressions `"/nom/de/fichier"` ou des tables de correspondances `"type:table"`, séparé par des virgules et/ou des espaces. Continuez les lignes longues en commençant la ligne suivante par des espaces. un `"/nom/de/fichier"` de correspondances est remplacé par son contenu; une table de correspondance `"type:table"` correspond lorsqu'un domaine (parent) apparaît comme clef.

relay_domains_reject_code (défaut : 554)

Le code numérique de réponse du serveur SMTP de Postfix lorsqu'une requête d'un client est rejetée par la restriction reject_unauth_destination.

Ne changez pas ceci avant d'avoir bien compris la RFC 821.

relay_recipient_maps (défaut : vide)

Tables de correspondances optionnelles contenant toutes les adresse valides des domaines correspondant à \$relay_domains. Indiquez `@domaine` pour valider tout un domaine qui n'a pas de liste de destinataires valides. Techniquement, les tables listées dans \$relay_recipient_maps sont utilisées comme listes: Postfix n'a besoin de savoir seulement si une chaîne est trouvée et ignore le résultat de la consultation.

Si ce paramètre n'est pas vide, le serveur SMTP de Postfix rejettera le courrier des utilisateurs relayés inconnus. Cette fonctionnalité est désactivée par défaut.

Voyez également la classe d'adresses relay_domains de la page ADDRESS CLASS README.

Exemple :

```
relay_recipient_maps = hash:/etc/postfix/relay_recipients
```

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

relay_transport (défaut : relay)

Le transporteur de messages par défaut et l'information sur le saut suivant pour les domaines correspondant à `$relay_domains`. Dans l'ordre décroissant de préférence, la destination suivante est choisie dans `$relay_transport`, `$sender_dependent_relayhost_maps`, `$relayhost`, ou par le domaine de destination. Cette information peut être surchargée par la table [transport\(5\)](#).

Indiquez une chaîne sous la forme `transport:nexthop`, où `transport` est le nom d'un transporteur de messages défini dans [master.cf](#). La partie `:nexthop` est optionnelle. Pour plus de détails voyez la page de manuel [transport\(5\)](#).

Étudiez également la classe d'adresse `relay_domains` à la page [ADDRESS CLASS README](#).

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

relayhost (défaut : vide)

La machine par défaut où livrer le courrier extérieur lorsqu'aucune entrée de la table optionnelle [transport\(5\)](#). Si aucun `relayhost` n'est renseigné, le courrier est routé directement vers sa destination finale (champs MX).

Dans un intranet, indiquez le nom de domaine de l'organisation. Si votre DNS interne n'utilise pas de champs MX, indiquez le nom de la passerelle à la place.

Dans le cas SMTP, indiquez un nom de domaine, de machine, `machine:port`, `[machine]:port`, `[adresse IP]` ou `[adresse IP]:port`. La forme `[machine]` désactive la consultation des champs MX.

Si vous êtes connectés via UUCP, reportez-vous à la page [UUCP README](#).

Exemples :

```
relayhost = $mydomain
relayhost = [gateway.my.domain]
relayhost = uucphost
relayhost = [1.2.3.4]
```

relocated_maps (défaut : vide)

Tables de correspondances optionnelles contenant les nouvelles coordonnées des utilisateurs ou des domaines plus supportés par le système. Le format de la table et des correspondances est exposé dans la page de manuel [relocated\(5\)](#).

Si vous utilisez cette fonctionnalité, lancez "**postmap /etc/postfix/relocated**" pour (re)construire le nécessaire fichier DBM ou DB après tout changement, lancez ensuite "**postfix reload**" pour activer ces changements.

Exemples :

```
relocated_maps = dbm:/etc/postfix/relocated
relocated_maps = hash:/etc/postfix/relocated
```

remote_header_rewrite_domain (défaut : vide)

Ne réécrit pas les en-têtes de message des clients distants lorsque ce paramètre est vide ; autrement, réécrit les en-têtes de message des clients distants et ajoute le nom de domaine indiqué aux adresses incomplètes. Le paramètre `local_header_rewrite_clients` contrôle quels clients Postfix doit considérer comme locaux.

Exemples :

Documentation de Postfix en français

Valeur sûre : ajoute "domain.invalid" aux adresses incomplètes dans les en-têtes des clients SMTP distants, ainsi ces adresses ne peuvent être confondues avec des adresses locales.

`remote_header_rewrite_domain = domain.invalid`

Valeur par défaut, puriste : ne réécrit pas les en-têtes des clients extérieurs.

`remote_header_rewrite_domain =`

require_home_directory (défaut : no)

Détermine si oui ou non le répertoire d'un utilisateur local(8) doit exister avant de tenter la livraison. Par défaut, ce test est désactivé. Ce peut être utile dans des environnements qui importent les répertoires utilisateurs au serveur de messagerie (DÉCONSEILLÉ).

resolve_dequoted_address (défaut : yes)

Résout une adresse de destination en sécurité au lieu de la résoudre correctement.

Par défaut, la résolution des adresses de Postfix n'examine pas la partie locale des adresses comme indiqué dans la RFC 822, ainsi les opérateurs additionnels @ or % or ! restent visibles. Ce comportement est sain mais également techniquement incorrect.

Si vous indiquez "resolve_dequoted_address = no", alors le résolveur de Postfix ne verra pas les opérateurs additionnels @, etc. de la partie locale de l'adresse. Ceci ouvre des opportunités pour d'obscurités attaques de relais de messagerie avec des adresses sous la forme utilisateur@domaine@domaine lorsque Postfix est un service MX de secours pour un système Sendmail.

resolve_null_domain (défaut : no)

Résout une adresse se terminant par le domaine nul "@" comme si le nom de la machine locale était indiqué au lieu de rejeter l'adresse comme invalide.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix. Les précédentes interprètent toujours le domaine nul comme nom de l'hôte local.

le serveur SMTP de Postfix utilise cette fonctionnalité pour rejeter le courrier pour ou provenant d'une adresse se terminant par "@", ou réécrite en une telle forme.

resolve_numeric_domain (défaut : no)

Résout "utilisateur@adresseIP" en "utilisateur@[adresseIP]", au lieu de rejeter l'adresse comme invalide.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

rewrite_service_name (défaut : rewrite)

Le nom du service de réécriture d'adresse. Ce service réécrit les adresses sous la forme standard et les résout en un tuple (méthode de livraison, machine suivante, destinataire).

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

sample_directory (défaut : /etc/postfix)

Le nom du répertoire contenant les exemples de fichiers de configuration de Postfix.

sender_based_routing (défaut : no)

Ce paramètre ne devrait pas être utilisé. Il a été remplacé par sender_dependent_relayhost_maps dans Postfix version 2.3.

sender_bcc_maps (défaut : vide)

Table de correspondances optionnelles des copies cachées (BCC: blind carbon-copy), indexée par adresse d'expédition. L'adresse BCC (les résultats multiples ne sont pas supportés) est ajoutée lorsque

le message provient de l'extérieur de Postfix.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

L'ordre de recherche dans les tables est le suivant :

- ◇ Examine l'adresse "user+extension@domain.tld" en incluant l'éventuelle extension d'adresse.
- ◇ Examine l'adresse "user@domain.tld" sans l'éventuelle extension d'adresse.
- ◇ Examine la partie locale de l'adresse "user+extension" lorsque le domaine du destinataire correspond à \$myorigin, \$mydestination, \$inet_interfaces ou \$proxy_interfaces.
- ◇ Examine la partie locale de l'adresse "user" lorsque le domaine du destinataire correspond à \$myorigin, \$mydestination, \$inet_interfaces ou \$proxy_interfaces.
- ◇ Examine la partie "@domain.tld".

Indiquez les types et noms des tables à utiliser. Après tout changement, lancez **"postmap /etc/postfix/sender_bcc"**.

NOTE: si le message à destination de l'adresse cachée BCC est rejeté, il est retourné à l'expéditeur.

NOTE : les copies cachées automatiques ne sont produites que pour les nouveaux messages. Pour éviter les boucles de messages, elles ne sont pas générées pour les messages que Postfix transfère en interne ni pour les messages générés par Postfix lui-même.

Exemple :

```
sender_bcc_maps = hash:/etc/postfix/sender_bcc
```

sender_canonical_classes (défaut : envelope_sender, header_sender)

Adresses sujettes aux réécritures d'adresses sender_canonical_maps. Par défaut, les réécritures d'adresses sender_canonical_maps sont appliquées à l'adresse d'expédition de l'enveloppe et des en-têtes.

Indiquez une proposition ou plus parmi : envelope_sender, header_sender

Cette fonctionnalité est disponible dans les versions 2.2 et supérieures de Postfix.

sender_canonical_maps (défaut : vide)

Tables optionnelles de correspondances d'adresses d'expéditions pour l'enveloppe et l'en-tête. Le format de la table et les correspondances sont documentés à la page canonical(5).

Exemple : vous voulez réécrire l'adresse d'EXPÉDITION "user@ugly.domain" en "user@pretty.domain", en gardant la possibilité de recevoir du courrier à l'adresse de DESTINATION "user@ugly.domain".

Note : \$sender_canonical_maps est examinée avant \$canonical_maps.

Exemple :

```
sender_canonical_maps = hash:/etc/postfix/sender_canonical
```

sender_dependent_relayhost_maps (défaut : vide)

Tables par expéditeur pour surcharger le paramètre de configuration global relayhost. Les tables sont consultées avec l'adresse d'expédition puis avec le @domaine. Cette information est surchargée par relay_transport, default_transport et les tables de transports.

Documentation de Postfix en français

Pour des raisons de sécurité, cette fonctionnalité n'autorise pas les substitutions \$nombre dans les tables d'expressions régulières.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

***sendmail_path* (défaut : voir la sortie de "postconf -d")**

Compatibilité Sendmail qui indique l'emplacement de la commande sendmail(1) de Postfix. Cette commande peut être utilisée pour soumettre un message à la file d'attente de Postfix.

***service_throttle_time* (défaut : 60s)**

Temps d'attente du démon master(8) de Postfix avant de doubler un serveur qui paraît dysfonctionner.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

***setgid_group* (défaut : postdrop)**

Le groupe propriétaire des commandes set-gid de Postfix et des répertoires en écriture pour le groupe. Lorsque ce paramètre est changé, vous devez relancer "**post-install set-permissions**" (avec Postfix 2.0 et versions supérieures : **"/etc/postfix/post-install set-permissions"**).

***show_user_unknown_table_name* (défaut : yes)**

Affiche le nom de la table des destinataires dans les réponses "User unknown". Ces détails supplémentaires aident au diagnostic mais révèlent l'information à l'extérieur.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

***showq_service_name* (défaut : showq)**

Le nom du service showq(8). Ce service produit les rapports sur les statuts des messages en file d'attente.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

***smtp_always_send_ehlo* (défaut : yes)**

Envoie systématiquement un EHLO en début de session SMTP.

Dans le cas contraire ("smtp_always_send_ehlo = no"), Postfix n'envoie un EHLO que lorsque le mot "ESMTP" apparaît dans la bannière d'accueil du serveur (exemple: 220 spike.porcupine.org ESMTP Postfix).

***smtp_bind_address* (défaut : vide)**

Une adresse réseau numérique optionnelle à partir de laquelle le client SMTP établit ses connexion.

Ceci peut être spécifié dans le fichier main.cf pour tous les clients SMTP ou dans le fichier master.cf pour un client spécifique. Par exemple :

```
/etc/postfix/master.cf :  
smtp ... smtp -o smtp_bind_address=11.22.33.44
```

Note 1 : lorsque inet_interfaces indique exactement une adresse qui n'est pas la boucle locale, elle est automatiquement utilisée pour smtp_bind_address. Ceci supporte l'hébergement IP virtuel mais peut être un problème sur un firewall hébergeant plusieurs sites. Reportez-vous à la documentation du paramètre inet_interfaces pour plus de détail.

Note 2 : les adresses peuvent être encadrées dans [], mais cette forme n'est pas recommandée ici.

***smtp_bind_address6* (défaut : vide)**

Une adresse réseau numérique IPv6 optionnelle à partir de laquelle le client SMTP établit ses connexion.

Documentation de Postfix en français

Cette fonctionnalité est disponibles dans les versions 2.3 et supérieures de Postfix.

Ceci peut être spécifié dans le fichier `main.cf` pour tous les clients SMTP ou dans le fichier `master.cf` pour un client spécifique. Par exemple :

```
/etc/postfix/master.cf:
smtp ... smtp -o smtp_bind_address6=1:2:3:4:5:6:7:8
```

Note 1 : lorsque `inet_interfaces` n'indique pas plus d'une adresse IPv6, et que cette adresse ne correspond pas à la boucle locale, elle est automatiquement utilisée par `smtp_bind_address6`. Ceci est compatible avec l'hébergement IP virtuel, mais peut poser problème sur un firewall multi-hôte. Voir le paragraphe `inet_interfaces` pour plus de détails.

Note 2 : les adresses peuvent être encadrées par [], mais cette forme n'est pas recommandée ici.

smtp_cname_overrides_servername (défaut : version dependent)

Autorise les enregistrement DNS de type CNAME à surcharger le nom de serveur que le client SMTP de Postfix utilise pour les journaux, les recherches de mots-de-passe SASL, les décisions concernant les politiques TLS ou la vérification des certificats. La valeur "no" renforce la politique TLS par machine `smtp_tls_per_site` de Postfix contre les faux noms de machine dans les enregistrement CNAME du DNS et rend plus sûr la consultation des fichiers de mots-de-passe SASL. C'est la valeur par défaut depuis Postfix 2.3.

Cette fonctionnalité est disponible sur les versions 2.2.9 et supérieures de Postfix.

smtp_connect_timeout (défaut : 30s)

Le temps limite pour que le client SMTP établisse une connexion TCP ou zéro (utilise la limite intrinsèque du système d'exploitation).

Lorsqu'aucune connexion ne peut être établie avant cette limite, le client SMTP tente de se connecter à l'adresse suivante dans la liste des échangeurs de messagerie. Indiquez 0 pour désactiver cette limite (ce qui signifie utiliser la limite implémentée dans le système d'exploitation).

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

smtp_connection_cache_destinations (défaut : vide)

Active en permanence le cache des connexions SMTP pour les destinations spécifiées. Avec le cache des connexions SMTP, une connexion n'est pas fermée immédiatement après la fin de la transaction de livraison. À la place, la connexion est transmise ouverte pendant `$smtp_connection_cache_time_limit` secondes. Ceci autorise la réutilisation des connexions pour d'autres livraisons, et peut améliorer les performances de livraison de messages.

Indiquez une liste de destinations ou pseudo-destinations séparées par des virgules ou des espaces :

- ◇ si le message est envoyé sans machine `relais` : un nom de domaine (la partie droite d'une adresse de messagerie sans les [] optionnels),
- ◇ si le message est envoyé via une machine `relais` : une machine `relais` (sans les [] optionnels ou à un port TCP autre que par défaut), comme indiqué dans `main.cf` ou la table de transport,
- ◇ si le message est envoyé via une socket du domaine UNIX : un chemin (sans le préfixe "unix:"),
- ◇ un /nom/de/fichier avec des domainss et/ou des machines relais,
- ◇ une "`type:table`" avec des domaines et/ou des machines relais sur la partie gauche. Le résultat (partie droite) de la consultation "`type:table`" est ignoré.

cette fonctionnalité est disponible dans les versions 2.2 et supérieures de Postfix.

smtp_connection_cache_on_demand (défaut : yes)

Active temporairement le cache des connexions SMTP tant qu'une destination a un volume important de messages en file d'attente active. Avec le cache de connexions SMTP, une connexion n'est pas fermée immédiatement après la fin d'une transaction de message. À la place, la connexion est maintenue ouverte pour un temps de `$smtp_connection_cache_time_limit` secondes. Ceci autorise les connexions à être réutilisées pour d'autres livraisons et peut améliorer les performances de la livraison.

Cette fonctionnalité est disponible dans les versions 2.2 et supérieures de Postfix.

smtp_connection_cache_reuse_limit (défaut : 10)

Lorsque le cache de connexions SMTP est activé, le temps pendant lequel une session SMTP est réutilisé avant d'être fermée.

Ce paramètre est disponible dans Postfix 2.2. Dans Postfix 2.3, il est remplacé par

`$smtp_connection_reuse_time_limit`.

smtp_connection_cache_time_limit (défaut : 2s)

Lorsque le cache de connexions SMTP est activé, temps pendant lequel une socket inutilisée du client SMTP est conservée ouverte. N'indiquez pas de valeur trop élevée sans la permission des sites distants.

cette fonctionnalité est disponible dans les versions 2.2 et supérieures de Postfix.

smtp_connection_reuse_time_limit (défaut : 300s)

La durée pendant laquelle Postfix peut utiliser une connexion SMTP. Le délai court à partir de l'établissement de la connexion (i.e. y compris la connexion, les échanges HELO, en plus de la latence de la transaction de livraison du message).

Cette fonctionnalité peut engendrer un problème de stabilité des performances avec les serveurs SMTP extérieurs. Ce problème n'est pas spécifique à Postfix : il peut se produire lorsqu'un MTA envoie un grand nombre de messages SMTP email à un site qui dispose de plusieurs machines MX.

Le problème apparaît lorsque l'une des machines MX devient plus lente que les autres. Bien que les clients SMTP se connectent avec une probabilité équivalente aux machines MX lentes ou rapides, les plus lentes se retrouvent connectées à plus de machines simultanément que les plus rapides, car les premières ont besoin de plus de temps pour servir chaque requête.

Les machines MX lentes deviennent attractives pour les connexions. Si une machine MX devient N fois plus lente que les autres, elle domine la latence de livraison sauf s'il y a plus de N machines MX plus rapide pour contrer l'effet. Si le nombre de machines MX est plus petit que N, la latence de livraison des messages devient effectivement celle de la plus lente divisée par le nombre total de machines MX.

La solution utilise le cache des connexions d'une manière différente que dans Postfix version 2.2. En limitant le délai pendant lequel une connexion peut être utilisée (au lieu de limiter le nombre de livraisons par connexions), Postfix ne restaure pas seulement l'équité de la distribution des connexions simultanées sur un ensemble de machines MX, il favorise également la livraison via les connexions les plus performantes, ce qui correspond exactement à ce que nous recherchons.

La durée maximum par défaut, 300s, est comparable aux différents délais limites des transactions SMTP et correspond à la latence maximum tolérée pour les livraisons lentes. Notez que les machines peuvent accepter des milliers de messages sur une seule connexion dans cette durée limite. Ce nombre

est plus grand que la limite de 10 messages par connexion cachée de Postfix version 2.2. Il peut être nécessaire de réduire cette limite pour conserver l'interopérabilité avec les MTA qui boguent lorsque de nombreux messages sont livrés dans la même connexion. Toutefois, un délai limite trop faible peut faire perdre le bénéfice du cache des connexions, et la latence de livraison peut excéder le temps limite.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

smtp_data_done_timeout (défaut : 600s)

Temps limite pour que le client SMTP envoie le "." SMTP et reçoive la réponse du serveur.

Lorsque la réponse n'arrive pas dans les délais, un avertissement est enregistré dans les journaux indiquant que le message a pu être envoyé plusieurs fois.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

smtp_data_init_timeout (défaut : 120s)

Temps limite pour que le client SMTP envoie la commande SMTP DATA et pour que le serveur réponde.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

smtp_data_xfer_timeout (défaut : 180s)

Temps limite pour que le client SMTP envoie le contenu du message SMTP. Lorsque la connexion est inactive plus de \$smtp_data_xfer_timeout secondes le client SMTP termine le transfert.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

smtp_defer_if_no_mx_address_found (défaut : no)

Retarde la livraison si aucun enregistrement MX n'est résolu en adresse IP.

Par défaut, les messages sont retournés comme non livrables. Avec les versions précédentes de Postfix, le comportement par défaut consistait à conserver le message et réessayer jusqu'à résolution du problème ou dépassement du temps limite.

Note : Postfix ignore toujours les enregistrements MX avec une préférence égale ou plus faible que le MTA local lui-même.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

smtp_destination_concurrency_limit (défaut : \$default_destination_concurrency_limit)

Nombre maximum de livraisons parallèles vers la même destination via le transporteur de messages smtp. Cette limite est utilisée par le gestionnaire des files d'attente. Le nom du transporteur de message correspond au premier champ du fichier master.cf.

smtp_destination_recipient_limit (défaut : \$default_destination_recipient_limit)

Nombre maximum of destinataires par livraison via le transporteur de messages smtp. Cette limite est utilisée par le gestionnaire des files d'attente. Le nom du transporteur de message correspond au premier champ du fichier master.cf.

Mettre ce paramètre à 1 change le sens de smtp_destination_concurrency_limit de concurrence par domaine en concurrence par destinataire.

smtp_discard_ehlo_keyword_address_maps (défaut : vide)

Tables de correspondances, indexées par les adresses des serveurs SMTP distants, avec une liste non sensible à la casse des mots-clés EHLO (pipelining, starttls, auth, etc.) que le client SMTP doit ignorer dans les réponses EHLO du serveur SMTP. Voir [smtp_discard_ehlo_keywords](#) pour les détails. La table n'est pas indexée par nom de machine for uniformité avec [smtpd_discard_ehlo_keyword_address_maps](#).

Cette fonctionnalité est disponibles dans les versions 2.3 et supérieures de Postfix.

smtp_discard_ehlo_keywords (défaut : vide)

Une liste insensible à la casse des mots-clés EHLO (pipelining, starttls, auth, etc.) que le client SMTP doit ignorer dans les réponses EHLO d'un serveur SMTP extérieur.

Cette fonctionnalité est disponibles dans les versions 2.3 et supérieures de Postfix.

Notes :

- ◇ Utilisez le pseudo mot-clé **silent-discard** pour éviter que ces actions soient enregistrées dans les journaux.
- ◇ Utilisez la fonctionnalité [smtp_discard_ehlo_keyword_address_maps](#) pour interdire les mots-clés EHLO selectivement.

smtp_enforce_tls (défaut : no)

Impose que les serveurs SMTP extérieurs utilisent le chiffrement TLS. Ceci impose également que le nom de machine du serveur SMTP corresponde à l'information de son certificat et que ce certificat soit issu d'une autorité reconnue par le client SMTP de Postfix. Dans le cas contraire, la livraison est retardée et le message reste en file d'attente.

Le nom de machine est comparé à tous les noms fournis comme dNSNames du SubjectAlternativeName. Si aucun dNSNames n'est indiqué, la comparaison est faite avec le CommonName. Ce comportement peut être changé avec l'option [smtp_tls_enforce_peername](#).

Cette option n'est intéressante que si vous êtes absolument sûr que vous ne vous connecterez qu'à des serveurs supportant la [RFC 2487](#) et fournissant des certificats serveurs valides. Il est relativement sûr de l'utiliser pour les clients locaux ne se connectant qu'à un commutateur supportant le nécessaire STARTTLS.

smtp_fallback_relay (défaut : \$fallback_relay)

Liste optionnelle des machines relais pour les destinations SMTP qui ne peuvent être trouvées ou jointes. Avec Postfix 2.2 et supérieur, ce paramètre est appelé [fallback_relay](#).

Par défaut, le message est retourné à l'expéditeur lorsqu'une destination n'est pas trouvée, et la livraison est retardée lorsque la destination ne peut être jointe.

Les relais de secours doivent être des destinations SMTP. Indiquez un domaine, une machine, machine:port, [machine]:port, [adresse.IP] ou [adresse.IP]:port ; la forme [machine] désactive la consultations DNS des champs MX. Si vous indiquez de multiples destinations SMTP, Postfix les essaiera dans l'ordre indiqué.

Pour éviter les boucles de messages entre machines MX et de secours, les versions 2.3 et supérieures de Postfix n'utilisent pas le [smtp_fallback_relay](#) pour une destination pour laquelle ce dernier est MX.

smtp_generic_maps (défaut : vide)

Tables optionnelles de correspondances qui exécutent des réécritures d'adresses dans le client SMTP, typiquement pour transformer une adresse locale en adresse valide lors de l'envoi d'un message via Internet. C'est utile lorsqu'une machine locale ne dispose pas de son propre nom de domaine Internet,

mais autre chose tel *localdomain.local*.

Le format et la consultation de ces tables sont documentés dans la page [generic\(5\)](#) ; des exemples sont présentés dans les pages [ADDRESS REWRITING README](#) et [STANDARD CONFIGURATION README](#).

Cette fonctionnalité est disponibles dans les versions 2.3 et supérieures de Postfix.

smtp_helo_name (défaut : \$myhostname)

Le nom de machine envoyé dans les commandes SMTP EHLO ou HELO.

la valeur par défaut est le nom de la machine. Indiquez un nom de machine ou [une.adresse.IP].

Cette information peut être indiquée dans le fichier main.cf pour tous les clients SMTP, ou il peut être spécifié dans le fichier master.cf de Postfix pour un client spécifique. Par exemple :

```
/etc/postfix/master.cf :  
mysmtp ... smtp -o smtp_helo_name=foo.bar.com
```

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

smtp_helo_timeout (défaut : 300s)

Temps limite pour que le client SMTP envoie la commande HELO ou EHLO et reçoive la réponse.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

smtp_host_lookup (défaut : dns)

Quel paramètre le client SMTP doit utiliser pour résoudre l'adresse IP d'une machine. Ce paramètre est ignoré lorsque la consultation DNS est désactivée.

Indiquez l'une des valeurs suivantes :

dns

Résolution par DNS (par défaut).

native

Résolution native seulement (nsswitch.conf, ou mécanisme équivalent).

dns, native

Résolution native pour les machines non trouvées par DNS.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

smtp_line_length_limit (défaut : 990)

Longueur maximale des lignes du corps ou de l'en-tête du message que Postfix transmettra par SMTP. Les lignes trop longues sont coupées en insérant "<CR><LF><SPACE>". Ceci minimise les dommages des messages formatés MIME.

Par défaut, la longueur de ligne est limitée à 990 caractères, car certaines implémentations de serveurs ne peuvent recevoir de lignes plus longues.

smtp_mail_timeout (défaut : 300s)

Temps limite pour que le client SMTP envoie la commande MAIL FROM et reçoive la réponse du serveur.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

smtp_mx_address_limit (défaut : 0)

Nombre maximum d'adresses IP MX (échangeurs de messagerie) pouvant résulter d'une recherche ou zéro (pas de limites).

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

smtp_mx_session_limit (défaut : 2)

Nombre maximum de sessions SMTP par requête de livraison avant d'arrêter ou de transmettre à la machine de secours (fall-back), ou zéro (pas de limite). Cette restriction ignore les adresses IP pour lesquelles l'ouverture de session échoue.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

smtp_never_send_ehlo (défaut : no)

N'envoie jamais de EHLO au démarrage d'une session SMTP. Voir également

smtp_always_send_ehlo.

smtp_pix_workaround_delay_time (défaut : 10s)

Temps d'attente avant que le client SMTP de Postfix envoie "<CR><LF>" pour contourner le bug "<CR><LF>.<CR><LF>" des firewalls PIX.

Choisir un temps trop court empêche ce contournement d'être effectif lors de l'envoi de messages longs sur des réseaux lents.

smtp_pix_workaround_threshold_time (défaut : 500s)

Temps de mise en file d'attente avant activation du contournement du bug "<CR><LF>.<CR><LF>" des firewalls PIX.

Par défaut, le contournement est désactivé pour le courrier en file d'attente depuis moins de 500 secondes. En d'autres mots, il est normalement désactivé à la première tentative de livraison.

Indiquez 0 pour activer le contournement du bug "<CR><LF>.<CR><LF>" des firewalls PIX dès la première tentative de livraison.

smtp_quit_timeout (défaut : 300s)

Temps limite pour que le client SMTP envoie la commande QUIT et reçoive la réponse.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

smtp_quote_rfc821_envelope (défaut : yes)

Encadre entre apostrophes les adresses dans les commandes SMTP MAIL FROM et RCPT TO comme requis par la RFC 821. Ceci inclut la mise entre apostrophes d'une adresse dont la partie locale se termine par ".".

La valeur par défaut est conforme à la RFC 821. Si vous devez envoyer du courrier à un serveur non compatible, configurez un client SMTP particulier dans master.cf :

```
/etc/postfix/master.cf :  
    broken-smtp . . . smtp -o smtp_quote_rfc821_envelope=no
```

et routez le courrier pour la destination en question vers ce "broken-smtp" dans la table transport(5).

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

smtp_randomize_addresses (défaut : yes)

Tire au hasard l'ordre des adresses des machines MX de même préférence. C'est une fonctionnalité de performance du client SMTP de Postfix.

smtp_rcpt_timeout (défaut : 300s)

Temps limite pour que le client SMTP envoie la commande SMTP RCPT TO et reçoive la réponse.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

smtp_rset_timeout (défaut : 120s)

Temps limite pour que le client SMTP envoie la commande RSET et reçoive la réponse. Le client SMTP envoie un RSET pour finir un sondage de vérification d'adresse ou pour vérifier que la session cachée est utilisable.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

smtp_sasl_auth_enable (défaut : no)

Active l'authentification SASL dans le client SMTP de Postfix. Par défaut, le client SMTP de Postfix n'utilise pas d'authentification.

Exemple :

```
smtp_sasl_auth_enable = yes
```

smtp_sasl_mechanism_filter (défaut : vide)

Si non vide, filtre la liste des mécanismes SASL offerts par le serveur SMTP. Les différentes implémentations clientes et serveurs peuvent supporter différentes listes de mécanismes. Par défaut, le client utilisera l'intersection des deux. smtp_sasl_mechanism_filter restreignent les mécanismes serveur que le client prendra en considération.

Indiquez des noms de mécanismes, des expressions "/nom/de/fichier" ou des tables de correspondances "type:table". Le résultat (partie droite) de la consultation "type:table" est ignorée.

Cette fonctionnalité est disponible dans les versions 2.2 et supérieures de Postfix.

Exemples :

```
smtp_sasl_mechanism_filter = plain, login
```

```
smtp_sasl_mechanism_filter = /etc/postfix/smtp_mechs
```

```
smtp_sasl_mechanism_filter = !gssapi, !login, static:rest
```

smtp_sasl_password_maps (défaut : vide)

Tables optionnelles de consultation du client SMTP contenant une entrée username:password par nom de machine ou domaine distant. Si un hôte n'a pas d'entrée de ce type, le client SMTP de Postfix n'essaiera pas de s'authentifier auprès de lui.

Le client SMTP de Postfix ouvre la table de correspondances avant d'entrer en cage chroot, vous pouvez ainsi la laisser dans le répertoire /etc/postfix.

smtp_sasl_path (défaut : vide)

Information spécifique à l'implémentation passée au plug-in SASL sélectionné dans smtp_sasl_type. Généralement, ceci indique le nom d'un fichier de configuration ou point de rendez-vous.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

smtp_sasl_security_options (défaut : noplain, noanonymous)

Options de sécurité SASL ; à partir de Postfix 2.3, la liste des fonctionnalités disponibles dépend de l'implémentation du client SASL sélectionné avec smtp_sasl_type.

Les options suivantes sont définies pour le client SASL **cyrus** :

noplaintext

Interdit les méthodes utilisant les mots de passe en clair.

noactive

Interdit les méthodes sujettes à une attaque active (sans dictionnaire).

nodictionary

Interdit les méthodes sujettes à une attaque passive (par dictionnaire).

noanonymous

Interdit les méthodes qui autorisent l'authentification anonyme.

mutual_auth

N'autorise que les méthodes fournissant une authentification mutuelle (non disponible avec SASL version 1).

Exemple :

```
smtp_sasl_security_options = noplaintext
```

smtp_sasl_tls_security_options (défaut : \$smtp_sasl_security_options)

Options de sécurité d'authentification SASL que le client SMTP de Postfix utilise dans les sessions SMTP chiffrées par TLS.

smtp_sasl_tls_verified_security_options (défaut : \$smtp_sasl_tls_security_options)

Options de sécurité de l'authentification SASL security que le client SMTP de Postfix utilise pour les sessions chiffrées TLS avec un certificat serveur vérifié.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

smtp_sasl_type (défaut : cyrus)

Type de plug-in SASL que le client SMTP de Postfix doit utiliser pour l'authentification. Les types disponibles sont affichés par la commande "**postconf -A**".

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

smtp_send_xforward_command (défaut : no)

Envoie la commande non-standard XFORWARD lorsque le serveur SMTP de Postfix reçoit l'annonce du support de XFORWARD en réponse à la commande EHLO.

Ceci permet à un agent de livraison "smtp", utilisé pour injecter un message dans un filtre de contenu, de transférer le nom, l'adresse, le protocole et le nom HELO du client original au filtre. Ceci peut produire des journaux plus utiles que localhost[127.0.0.1] etc.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

smtp_sender_dependent_authentication (défaut : no)

Active l'authentification en fonction de l'expéditeur dans le client SMTP ; ceci n'est disponible qu'avec l'authentification SASL, et désactive le cache des connexions SMTP pour s'assurer que le courrier de différents expéditeurs sera utilisé avec les éléments d'authentification appropriés.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

smtp_skip_4xx_greeting (défaut : yes)

Ignore les serveurs SMTP qui répondent avec un code de statut 4XX (go away, try again later).

Par défaut, Postfix utilise l'échangeur de messagerie suivant. Indiquez "smtp_skip_4xx_greeting = no" si Postfix doit retarder la livraison immédiatement.

Cette fonctionnalité est disponible dans les versions 2.0 et antérieures de Postfix. Les versions supérieures ignorent toujours les serveurs SMTP accueillant les connexions avec un code de statut 4XX.

smtp_skip_5xx_greeting (défaut : yes)

Ignore les serveurs SMTP qui répondent avec un code de statut 5XX (go away, try again later).

Par défaut, le client SMTP de Postfix redirige le courrier vers l'échangeur de messagerie suivant. Indiquez "smtp_skip_5xx_greeting = no" si Postfix doit renvoyer le courrier immédiatement. La valeur par défaut est incorrecte, mais est le comportement que beaucoup attendent.

smtp_skip_quit_response (défaut : yes)

Ne pas attendre la réponse à la commande SMTP QUIT.

smtp_starttls_timeout (défaut : 300s)

Temps limite pour que le client SMTP de Postfix écrive et reçoive les éléments des procédures de démarrage et d'arrêt de TLS.

smtp_tls_CAfile (défaut : vide)

Fichier contenant les certificats des autorités (CA) que le client SMTP de Postfix utilisera pour valider les certificats des serveurs distants. Ceci n'est nécessaire que si le certificat de l'autorité n'est pas présent dans le fichier du certificat du client.

Exemple :

```
smtp_tls_CAfile = /etc/postfix/CAcert.pem
```

smtp_tls_CApith (défaut : vide)

Répertoire contenant les certificats des autorités (CA) que le client SMTP de Postfix utilisera pour valider les certificats des serveurs distants. N'oubliez pas de créer les nécessaires liens symboliques "hash" avec par exemple "\$OPENSSL_HOME/bin/c_rehash /etc/postfix/certs".

Pour utiliser cette option en mode chroot, ce répertoire (ou une copie) doit être dans la cage chroot.

Exemple :

```
smtp_tls_CApith = /etc/postfix/certs
```

smtp_tls_cert_file (défaut : vide)

Fichier contenant le certificat RSA du client SMTP de Postfix au format PEM. Ce fichier peut également contenir la clef privée du client et peut être le même que celui du serveur SMTP.

Pour vérifier les certificats, le certificat de l'autorité (tous les certificats d'autorité dans le cas d'une chaîne de certification) doit être disponible. Vous devrez ajouter ces certificats au certificat du serveur, le certificat du serveur en premier, ceux des autorités ensuite.

Exemple : le certificat de "client.dom.ain" est issu de "intermediate CA" elle-même certifiée par "root CA". Créez le fichier client.pem file avec

```
"cat client_cert.pem intermediate_CA.pem root_CA.pem > client.pem".
```

Si vous voulez accepter les certificats des serveur SMTP distants issus de ces autorités, vous pouvez également ajouter ces certificats d'autorité au fichier smtp_tls_CAfile, auquel cas il n'est pas nécessaire de les insérer dans le fichier smtp_tls_cert_file ou smtp_tls_dcrt_file.

Un certificat utilisé ici doit être utilisable comme certificat client SSL et donc passer le test "openssl verify -purpose sslclient ...".

Exemple :

```
smtp_tls_cert_file = /etc/postfix/client.pem
```

smtp_tls_cipherlist (défaut : vide)

Contrôle le schéma de sélection du chiffrement TLS du client SMTP de Postfix. Pour plus de détails, lisez la documentation d'OpenSSL. Note : n'utilisez pas de guillemets "" autour de la valeur de ce paramètre.

smtp_tls_dcert_file (défaut : vide)

Fichier contenant le certificat DSA du client SMTP de Postfix au format PEM. Ce fichier peut également contenir la clef privée du serveur.

Reportez-vous au paragraphe smtp_tls_cert_file pour plus de détails.

Exemple :

```
smtp_tls_dcert_file = /etc/postfix/client-dsa.pem
```

smtp_tls_dkey_file (défaut : \$smtp_tls_dcert_file)

Fichier contenant la clef privée DSA du client SMTP de Postfix au format PEM. La clef privée ne doit pas être chiffrée. En d'autres mots, la clef doit être accessible sans mot-de-passe.

Ce fichier peut être le même que le fichier contenant le certificat du serveur : \$smtp_tls_cert_file.

smtp_tls_enforce_peername (défaut : yes)

Lorsque le chiffrement TLS est obligatoire, requiert que le nom de machine du serveur SMTP corresponde à l'information contenue dans son certificat. Comme indiqué dans la RFC 2487 cette correspondance avec le nom de machine n'est pas activée pour les clients MTA.

Cette option peut être mise à "no" pour désactiver le contrôle strict du nom. Ce paramètre n'a pas d'effet pour les sessions contrôlées via la table smtp_tls_per_site.

Désactiver la vérification du nom de machine peut avoir un sens dans un environnement clos où des autorités particulières sont créées. Utilisée sans attention particulière, cette option ouvre une brèche pour une attaque "man-in-the-middle" (le CommonName de cet attaquant sera enregistré).

smtp_tls_key_file (défaut : \$smtp_tls_cert_file)

Fichier contenant la clef privée RSA du client SMTP de Postfix au format PEM. Ce fichier peut être le même que le fichier contenant le certificat du serveur : \$smtp_tls_cert_file.

La clef privée ne doit pas être chiffrée. En d'autres mots, la clef doit être accessible sans mot-de-passe.

Exemple :

```
smtp_tls_key_file = $smtp_tls_cert_file
```

smtp_tls_loglevel (défaut : 0)

Active l'enregistrement additionnel de l'activité TLS du client SMTP de Postfix. Chaque niveau de log inclus également les informations des niveaux inférieurs.

- 0 Désactive l'enregistrement de l'activité TLS.
- 1 Enregistre les informations concernant la négociation et les certificat.
- 2 Enregistre les niveaux durant la négociation TLS.
- 3 Enregistre la copie hexadécimale et ASCII du processus de négociation TLS.
- 4 Enregistre également la retranscription complète hexadécimale et ASCII de la session après STARTTLS.

Utilisez "smtp_tls_loglevel = 3" seulement en cas de problèmes. L'utilisation du niveau 4 est vivement déconseillée.

smtp_tls_note_starttls_offer (défaut : no)

Documentation de Postfix en français

Enregistre les noms de machine serveur SMTP extérieurs qui offrent STARTTLS lorsqu'un TLS n'est pas déjà activé pour ce serveur.

Les enregistrements dans les journaux ressemblent à :

```
postfix/smtp[pid]: Host offered STARTTLS: [name.of.host]
```

smtp_tls_per_site (défaut : vide)

Tables de correspondances optionnelles contenant la politique d'emploi de TLS pour le client SMTP de Postfix par destination suivante et par nom de serveur SMTP extérieur. Lorsque les deux consultations réussissent, la politique la plus spécifique par site (NONE, MUST, etc.) prend la pas sur la moins spécifique (MAY), et la politique par site la plus sécurisée (MUST, etc.) prend le pas sur la moins sécurisée (NONE).

Indiquez une destination suivante ou le nom de machine d'un serveur sur la partie gauche ; aucune carte blanche n'est autorisée. La destination suivante est soit le domaine du destinataire, soit une destination indiquée dans la table transport(5), dans le paramètre relayhost ou dans le paramètre relay_transport. Sur la partie droite, indiquez l'un des mots-clefs suivants :

NONE

Ne pas utiliser TLS. Ceci surcharge un résultat **MAY** issu de la consultation avec l'autre clef de recherche (destination ou serveur), et surcharge également les paramètres globaux smtp_use_tls, smtp_enforce_tls, et smtp_tls_enforce_peername.

MAY

Tente d'utiliser STARTTLS s'il est annoncé par le serveur, autrement utilise une connexion en clair. Ce résultat a une préférence moindre qu'un résultat plus spécifique (y compris **NONE**) issu de la consultation avec l'autre clef de recherche (destination ou serveur), ainsi qu'un paramétrage global plus spécifique "smtp_enforce_tls = yes" ou "smtp_tls_enforce_peername = yes".

MUST_NOPEERMATCH

Requiert l'emploi de STARTTLS, mais ne requiert pas que le nom de machine du serveur SMTP corresponde aux informations du certificat ni que ce certificat soit issu d'une autorité reconnue. Ce résultat surcharge un résultat moins sécurisé **NONE** ou moins spécifique **MAY** issu de la consultation avec l'autre clef de recherche (destination ou serveur) et surcharge les paramètres globaux smtp_use_tls, smtp_enforce_tls, et smtp_tls_enforce_peername.

MUST

Requiert l'emploi de STARTTLS, que le nom de machine du serveur SMTP corresponde aux informations du certificat et que ce certificat soit issu d'une autorité reconnue. Ce résultat surcharge un résultat moins sécurisé **NONE** ou **MUST_NOPEERMATCH** ou moins spécifique **MAY** issu de la consultation avec l'autre clef de recherche (destination ou serveur) et surcharge les paramètres globaux smtp_use_tls, smtp_enforce_tls, et smtp_tls_enforce_peername.

Tant qu'aucun mécanisme de consultation DNS sécurisé n'est disponible, de fausses réponses sur les noms de machines dans les réponses MX or CNAME peuvent changer le nom de machine du serveur que Postfix utilise pour le choix de la politique TLS et la vérification du certificat du serveur. Même avec une correspondance parfaite entre le nom de machine et le certificat de serveur, il n'y a pas de garantie que Postfix est connecté au bon serveur. Pour éviter ce trou de sécurité, suivez les étapes suivantes :

- ◇ Désactivez la surcharge des nom de machine par les CNAME. Dans main.cf indiquez "smtp_cname_overrides_servername = no". Ceci évite aux faux noms de machines dans les enregistrement DNS CNAME de changer le nom de machine du serveur. Cette fonctionnalité

requires Postfix 2.2.9 or later.

- ◇ Eliminez les consultations MX. Renseignez une table locale transport(5) pour les domaines sensibles avec des destinations explicites smtp:[échangeur] ou smtp:[échangeur]:port. Ceci évite que de faux noms de machines dans les enregistrements DNS MX ne changent le nom de machine que Postfix utilise pour le choix de la politique TLS et la vérification du certificat du serveur.

- ◇ Utilisez "MUST" pour ces échangeurs (y compris [] et port) dans la table smtp_tls_per_site.

smtp_tls_scert_verifydepth (défaut : 5)

Profondeur de vérification des certificats de serveurs SMTP distants. Une profondeur de 1 est suffisante si le certificat est directement issu d'une autorité listée dans les fichiers CA. La valeur par défaut (5) devrait suffir pour les chaînes plus longues (l'autorité racine certifie une autorité qui certifie le certificat actuel...).

smtp_tls_session_cache_database (défaut : vide)

Nom du fichier contenant le cache optionnel des sessions TLS du client SMTP de Postfix. Utilisez un type de table qui supporte l'énumération tel **btree** ou **sdbm** ; il n'est pas nécessaire de supporter les accès concurrents. Le fichier est créé s'il n'existe pas déjà.

Note : les bases de données **dbm** ne sont pas utilisables ici car les objets TLS sont trop gros.

Exemple :

```
smtp_tls_session_cache_database = btree:/var/postfix/smtp_scache
```

smtp_tls_session_cache_timeout (défaut : 3600s)

Temps d'expiration des informations du cache des sessions TLS du client SMTP de Postfix. Un nettoyage du cache est effectué périodiquement toutes les \$smtp_tls_session_cache_timeout secondes.

smtp_use_tls (défaut : no)

Utiliser toujours TLS lorsque le serveur SMTP annonce le support STARTTLS. Attention : certains serveurs SMTP offrent STARTTLS même s'il n'est pas configuré. Si la négociation TLS échoue et qu'aucun autre serveur n'est disponible, la livraison est retardée et le message reste en file d'attente. Si vous êtes concernés, utilisez la fonctionnalité smtp_tls_per_site à la place.

smtp_xforward_timeout (défaut : 300s)

Temps limite pour que le client SMTP envoie la commande XFORWARD et reçoive la réponse.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

smtpd_authorized_verp_clients (défaut : \$authorized_verp_clients)

Clients SMTP autorisés à envoyer la commande XVERP. Cette commande requiert que le message doit être livré à un destinataire à la fois avec une adresse de retour spécifique.

Par défaut, aucun client n'est autorisé à envoyer XVERP.

Ce paramètre a été renommé avec Postfix 2.1. La valeur par défaut est compatible avec Postfix 2.0.

Indiquez une liste d'expression adresse/réseau, séparé par des virgules et/ou des espaces. Vous pouvez également indiquer des noms de machines ou des noms de domaines (s'ils sont précédés d'un "." tous les sous-domaines sont pris en compte), des "/nom/de/fichier" ou des expressions "type:table". Un "/nom/de/fichier" de correspondances est remplacé par son contenu; une table de correspondances "type:table" correspond lorsqu'une entrée de la table correspond (le résultat de la consultation est ignoré). Continuez les lignes longues en commençant la ligne suivante par des espaces.

Note : les adresses IP version 6 doivent être indiquées entre [] dans smtpd_authorized_verp_clients et dans les fichiers indiqués avec "/nom/de/fichier". Les adresses IP version 6 contiennent le caractère ":" et pourraient être confondues avec une expression "type:table".

smtpd_authorized_xclient_hosts (défaut : vide)

Clients SMTP autorisés à utiliser la fonctionnalité XCLIENT. Cette commande surcharge l'information du client SMTP utilisée pour le contrôle d'accès. L'utilisation type est le test des filtres de contenus, des programmes type fetchmail ou des règles d'accès du serveur SMTP. Reportez-vous à la page XCLIENT README pour plus de détails.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

Par défaut, aucun client n'est autorisé à utiliser XCLIENT.

Indiquez une liste d'expression adresse/réseau, séparé par des virgules et/ou des espaces. Vous pouvez également indiquer des noms de machines ou des noms de domaines (s'ils sont précédés d'un "." tous les sous-domaines sont pris en compte), des "/nom/de/fichier" ou des expressions "type:table". Un "/nom/de/fichier" de correspondances est remplacé par son contenu; une table de correspondances "type:table" correspond lorsqu'une entrée de la table correspond (le résultat de la consultation est ignoré). Continuez les lignes longues en commençant la ligne suivante par des espaces.

Note : les adresses IP version 6 doivent être indiquées entre [] dans smtpd_authorized_xclient_hosts et dans les fichiers indiqués avec "/nom/de/fichier". Les adresses IP version 6 contiennent le caractère ":" et pourraient être confondues avec une expression "type:table".

smtpd_authorized_xforward_hosts (défaut : vide)

Clients SMTP autorisés à utiliser la fonctionnalité XFORWARD. Cette commande transfère les informations utilisées pour générer les journaux après les filtres de contenu. Reportez-vous à la page XFORWARD README pour plus de détails.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

Par défaut, aucun client n'est autorisé à utiliser XFORWARD.

Indiquez une liste d'expression adresse/réseau, séparé par des virgules et/ou des espaces. Vous pouvez également indiquer des noms de machines ou des noms de domaines (s'ils sont précédés d'un "." tous les sous-domaines sont pris en compte), des "/nom/de/fichier" ou des expressions "type:table". Un "/nom/de/fichier" de correspondances est remplacé par son contenu; une table de correspondances "type:table" correspond lorsqu'une entrée de la table correspond (le résultat de la consultation est ignoré). Continuez les lignes longues en commençant la ligne suivante par des espaces.

Note : les adresses IP version 6 doivent être indiquées entre [] dans smtpd_authorized_xforward_hosts et dans les fichiers indiqués avec "/nom/de/fichier". Les adresses IP version 6 contiennent le caractère ":" et pourraient être confondues avec une expression "type:table".

smtpd_banner (défaut : \$myhostname ESMTP \$mail_name)

Texte qui suit le code de statut 220 dans la bannière d'accueil. Certaines personnes aiment y voir la version, par défaut Postfix ne la montre pas.

Vous DEVEZ indiquer \$myhostname au début du texte. C'est requis par le protocole SMTP.

Exemple :

```
smtpd_banner = $myhostname ESMTP $mail_name ($mail_version)
```

smtpd_client_connection_count_limit (défaut : 50)

Nombre de connexions simultanées qu'un client SMTP est autorisé à établir avec le service SMTP. Par défaut, cette limite vaut la moitié du nombre de processus autorisés.

Pour désactiver cette limite, indiquez 0.

ATTENTION : Le but de cette fonctionnalité est de limiter les abus. Elle ne doit pas être utilisée pour réguler le trafic légitime.

Cette fonctionnalité ne fait pas partie de la version stable 2.1 de Postfix.

smtpd_client_connection_rate_limit (défaut : 0)

Nombre maximum de tentatives de connexion qu'un client est autorisé à faire à ce service par unité de temps. L'unité de temps est indiqué par le paramètre de configuration anvil_rate_time_unit (1 minute par défaut).

Par défaut, un client peut faire autant de connexions par unité de temps que Postfix peut accepter.

Pour désactiver cette fonctionnalité, indiquez 0.

ATTENTION: Le but de cette fonctionnalité est de limiter les abus. Elle ne doit pas être utilisée pour réguler le trafic légitime.

Cette fonctionnalité ne fait pas partie de la version stable 2.1 de Postfix.

Exemple :

smtpd_client_connection_rate_limit = 1000

smtpd_client_event_limit_exceptions (défaut : \$mynetworks)

Clients exclus des restrictions sur le compte, le taux de connexions, le taux de messages ou de destinataires.

Par défaut, les clients des réseaux internes sont exclus. Indiquez une liste de réseaux, de noms de machines ou .noms-de-domaine (le point initial permet de faire correspondre tous les sous-domaines).

Note : les adresses IP version 6 doivent être indiquées entre [] dans smtpd_client_event_limit_exceptions et dans les fichier indiqués avec "/nom/de/fichier". Les adresses IP version 6 contiennent le caractère ":" et pourraient être confondues avec une expression "type:table".

Cette fonctionnalité est disponible dans les versions 2.2 et supérieures de Postfix.

smtpd_client_message_rate_limit (défaut : 0)

Nombre maximal de requêtes de livraison de messages que tout client est autorisé à faire à ce service par unité de temps, suivant que Postfix accepte ou non ces messages. L'unité de temps est indiquée au paramètre de configuration anvil_rate_time_unit.

Par défaut, un client peut envoyer autant de requêtes de livraison de messages que Postfix est en mesure d'accepter.

Pour désactiver cette fonctionnalité, indiquez 0.

ATTENTION : le but de cette fonctionnalité est de limiter les abus. Il ne doit pas être utilisé pour réguler le trafic de message légitime.

Cette fonctionnalité est disponible dans les versions 2.2 et supérieures de Postfix.

Exemple :

```
smtpd_client_message_rate_limit = 1000
```

smtpd_client_new_tls_session_rate_limit (défaut : 0)

Nombre maximal de nouvelles (c'est à dire non cachées) sessions TLS qu'un client SMTP extérieur est autorisé à négocier avec ce service par unité de temps. L'unité de temps est indiquée par le paramètre de configuration anvil_rate_time_unit.

Par défaut, un client SMTP extérieur peut négocier autant de nouvelles sessions TLS par unité de temps que Postfix peut accepter.

Pour désactiver cette fonctionnalité, utilisez une limite à 0. Autrement, indiquez une limite au moins égale à la limite de sessions concurrentes par client, sinon des sessions légitimes seront rejetées.

ATTENTION : le but de cette fonctionnalité est de limiter les abus. Elle ne doit pas être utilisée pour réguler le trafic légitime.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

Exemple :

```
smtpd_client_new_tls_session_rate_limit = 100
```

smtpd_client_recipient_rate_limit (défaut : 0)

Nombre maximal d'adresses de destination qu'un client est autorisé à envoyer à ce service par unité de temps, suivant que Postfix accepte ou non ces destinataires. L'unité de temps est indiquée au paramètre de configuration anvil_rate_time_unit.

Par défaut, un client peut inscrire autant d'adresses de destination par unité de temps que Postfix est en mesure d'accepter.

Pour désactiver cette fonctionnalité, indiquer 0.

ATTENTION : le but de cette fonctionnalité n'est pas de limiter les abus. Il ne doit pas être utilisé pour réguler le trafic de message légitime.

Cette fonctionnalité est disponible dans les versions 2.2 et supérieures de Postfix.

Exemple :

```
smtpd_client_recipient_rate_limit = 1000
```

smtpd_client_restrictions (défaut : vide)

Restrictions d'accès optionnelles du serveur SMTP pour les requêtes de connexion au service SMTP.

Par défaut, toutes les requêtes de connexion sont autorisées.

Indiquez une liste de restrictions, séparé par des virgules et/ou des espaces. Continuez les lignes

longues en commençant la ligne suivante par des espaces. Les restrictions sont appliquées dans l'ordre indiqué, la première restriction qui correspond est appliquée et les suivantes sont ignorées.

Les restrictions suivantes sont spécifiques aux informations sur l'adresse réseau ou le nom de machine du client.

check_ccert_access type:table

Lorsqu'un certificat de client SMTP certificate est vérifié, utilise l'empreinte du certificat comme clef de consultation pour la table d'accès indiquée. Cette fonctionnalité est disponible à partir de la version 2.2 de Postfix.

check_client_access type:table

Recherche dans la table indiquée le nom de machine, l'adresse IP, le réseau ou les domaines parents du client. Reportez-vous à la page de manuel access(5) pour plus de détails.

permit_inet_interfaces

Autorise la requête si l'adresse IP du client correspond à \$inet_interfaces.

permit_mynetworks

Autorise la requête si l'adresse IP du client correspond à l'une des adresses ou l'un des réseaux listé dans \$mynetworks.

permit_sasl_authenticated

Autorise la requête lorsque le client est authentifié avec succès via le protocole AUTH (RFC 2554).

permit_tls_all_clientcerts

Autorise la requête lorsque le certificat du client SMTP est vérifié. Cette option ne doit être utilisée que si une autorité particulière délivre les certificats et que seule celle-ci est reconnue, sinon tous les clients avec un certificat reconnu seront autorisés à relayer le courrier.

permit_tls_clientcerts

Autorise la requête lorsque le certificat du client SMTP est vérifié et que son empreinte est listée dans \$relay_clientcerts.

reject_rbl_client rbl_domain=d.d.d.d

Rejette la requête lorsque la résolution inverse de l'adresse réseau du client est l'enregistrement de type A "d.d.d.d" du domaine rbl_domain (Postfix version 2.1 et supérieures uniquement). Si aucun "=d.d.d.d" n'est indiqué, rejette la requête lorsque la résolution inverse de l'adresse réseau du client correspond à un enregistrement de type A du domaine rbl_domain.

Le paramètre maps_rbl_reject_code indique le code de réponse des requêtes rejetées (défaut : 554), le paramètre default_rbl_reply indique la réponse par défaut du serveur et le paramètre rbl_reply_maps indique les tables de réponses indexées par domaine rbl_domain. Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

reject_rhsbl_client rbl_domain=d.d.d.d

Rejette la requête lorsque le nom de machine du client est correspond à l'enregistrement de type A "d.d.d.d" du domaine rbl_domain (Postfix version 2.1 et supérieures uniquement). Si aucun "=d.d.d.d" n'est indiqué, rejette la requête lorsque la résolution inverse de l'adresse réseau du client correspond à un enregistrement de type A du domaine rbl_domain. Voir le paragraphe reject_rbl_client ci-dessus pour les paramètres relatifs aux RBL (realtime black lists). Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

***reject_unknown_client_hostname* (with Postfix < 2.3: reject_unknown_client)**

Rejette la requête lorsque 1) la requête DNS adresse-IP->nom échoue, 2) la requête nom->adresse-IP échoue ou 3) la requête nom->adresse-IP ne correspond pas à l'adresse IP du client.

C'est une restriction plus forte que reject_unknown_reverse_client_hostname, qui ne vérifie

que la première condition ci-dessus.

Le paramètre unknown_client_reject_code indique le code de réponse pour les requêtes rejetées (défaut : 450). La réponse est toujours 450 dans le cas où les requêtes DNS adresse-IP->nom ou nom->adresse-IP échouent en raison d'un problème temporaire.

reject_unknown_reverse_client_hostname

Rejette la requête lorsque la requête DNS adresse-IP->nom échoue.

C'est une restriction plus faible que reject_unknown_client_hostname, qui requiert nom seulement la réussite des requêtes DNS adresse-IP->nom et nom->adresse, mais également que les deux requêtes retrouvent l'adresse IP du client.

Le paramètre unknown_client_reject_code indique le code de réponse pour les requêtes rejetées (défaut : 450). La réponse est toujours 450 dans le cas où les requêtes DNS adresse-IP->nom ou nom->adresse-IP échouent en raison d'un problème temporaire.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

En complément, vous pouvez utiliser n'importe quelle restriction générique. Elles sont applicables dans tous les contextes de commandes SMTP.

check_policy_service servername

Interroge le serveur de politique indiqué. Reportez-vous à la page

SMTPD_POLICY_README pour plus de détails. Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

defer

Retarde la requête. Le client est prié de réessayer plus tard. Cette restriction est pratique en fin de liste de restrictions pour rendre explicite la politique par défaut.

Le paramètre defer_code indique le code de réponse par défaut du serveur SMTP (défaut : 450).

defer_if_permit

Retarde la requête si une des restrictions suivantes renvoie une action PERMIT explicite ou implicite. C'est pratique lorsqu'une fonctionnalité "liste noire" échoue suite à un problème temporaire. Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

defer_if_reject

Retarde la requête si une des restrictions suivantes renvoie une action REJECT. C'est pratique lorsqu'une fonctionnalité "liste blanche" échoue suite à un problème temporaire. Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

permit

Autorise la requête. Cette restriction est pratique à la fin d'une liste de restrictions, pour rendre la politique par défaut explicite.

reject_multi_recipient_bounce

Rejette la requête lorsque l'expéditeur dans l'enveloppe est l'adresse nulle et que le message contient plusieurs destinataires dans l'enveloppe. Cet usage a de rares mais légitimes applications : sous certaines conditions, un message multi-destinataires ayant été posté avec l'option DSN "NOTIFY=DEFER" peut être transféré avec une adresse d'expédition nulle.

Note : cette restriction ne peut fonctionner correctement que si elle est utilisée dans smtpd_data_restrictions ou smtpd_end_of_data_restriction, car le nombre total de destinataires n'est pas connu à une étape précédente du protocole SMTP. L'utiliser à l'étape RCPT TO ne rejettera que les destinataires au delà du second.

Le paramètre multi_recipient_bounce_reject_code indique le code de réponse pour les requêtes rejetées (défaut : 550). Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

reject_plaintext_session

Rejette la requête lorsque la connexion n'est pas chiffrée. Cette restriction ne doit pas être utilisée avant que le client n'ait eu la possibilité de négocier le chiffrement avec les

commandes AUTH ou STARTTLS.

Le paramètre plaintext_reject_code indique le code de réponse pour les requêtes rejetées (défaut : 450). Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

reject_unauth_pipelining

Rejette la requête lorsque le client envoie des commandes SMTP en dehors des moments où il y est autorisé ou lorsque le client envoie des commandes SMTP avant de savoir que Postfix supporte la canalisation des commandes SMTP (*pipelining*). Ceci arrête les messages issus de logiciels mal écrits qui utilisent incorrectement la canalisation des commandes SMTP pour accélérer les livraisons.

NOTE : reject_unauth_pipelining n'est pas pratique en dehors de smtpd_data_restrictions lorsque : 1) le client utilise ESMTP (EHLO au lieu de HELO) et 2) avec "smtpd_delay_reject = yes" (valeur par défaut). L'utilisation de reject_unauth_pipelining dans les autres contextes de restriction n'est pas recommandé.

reject

Rejette la requête. Cette restriction est pratique à la fin d'une liste de restrictions pour rendre explicite la politique par défaut. Le paramètre de configuration reject_code indique le code de réponse pour les requêtes rejetées (défaut : 554).

sleep_seconds

Attend le délai indiqué (en secondes) et passe à la restriction suivante sur la liste s'il y en a une. Ceci peut stopper les messages zombies lorsqu'utilisé comme suit :

```
/etc/postfix/main.cf:  
smtpd_client_restrictions =  
    sleep 1, reject_unauth_pipelining  
smtpd_delay_reject = no
```

Cette fonctionnalité est disponible à partir de la version 2.3 de Postfix.

warn_if_reject

Change le sens de la restriction suivante : enregistre un avertissement au lieu de rejeter la requête (examinez les enregistrements dans les journaux contenant "reject_warning"). C'est pratique pour tester de nouvelles restrictions dans un environnement en production sans risquer de perdre du courrier.

Autres restrictions valides dans ce contexte :

- ◇ Restrictions spécifiques sur les commandes SMTP décrites aux paragraphes smtpd_helo_restrictions, smtpd_sender_restrictions ou smtpd_recipient_restrictions. Lorsque les restrictions sur HELO, l'expéditeur ou les destinataires sont listés dans smtpd_client_restrictions, ils n'ont d'effet qu'avec "smtpd_delay_reject = yes", ainsi \$smtpd_client_restrictions est évaluée au moment de la commande RCPT TO.

Exemple :

```
smtpd_client_restrictions = permit_mynetworks, reject_unknown_client  
smtpd_data_restrictions (défaut : vide)
```

Restrictions d'accès optionnelles que le serveur SMTP de Postfix applique dans le contexte d'une commande SMTP DATA.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

Indiquez une liste de restrictions, séparées par des virgules et/ou des espaces. Continuez les lignes longues en commençant la ligne suivante par des espaces. Les restrictions sont appliquées dans l'ordre indiqué ; la première restriction qui correspond gagne.

Les restrictions suivantes sont valides dans ce contexte :

- ◇ Restrictions génériques pouvant être utilisées dans tous les contextes de commandes SMTP, décrites au paragraphe smtpd_client_restrictions.
- ◇ Restrictions spécifiques aux commandes SMTP décrites au paragraphes smtpd_client_restrictions, smtpd_helo_restrictions, smtpd_sender_restrictions ou smtpd_recipient_restrictions.

Exemples :

```
smtpd_data_restrictions = reject_unauth_pipelining  
smtpd_data_restrictions = reject_multi_recipient_bounce
```

smtpd_delay_open_until_valid_rcpt (défaut : yes)

Retarde le début de la transaction SMTP jusqu'à la réception d'une commande RCPT TO valide. Indiquez "no" pour créer la transaction dès que le serveur SMTP reçoit une commande MAIL FROM valide.

Sur les sites qui rejettent un grand nombre de message, la valeur par défaut réduit l'utilisation des ressources disque, CPU et mémoire. L'inconvénient est que les destinataires rejetés sont enregistrés avec NOQUEUE au lieu d'un identifiant de transaction (ID). Ceci complique l'analyse des fichiers de log pour les messages multi-destinataires.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

smtpd_delay_reject (défaut : yes)

Attend la commande RCPT TO avant d'évaluer \$smtpd_client_restrictions, \$smtpd_helo_restrictions et \$smtpd_sender_restrictions, ou attend la la commande ETRN avant d'évaluer \$smtpd_client_restrictions et \$smtpd_helo_restrictions.

Cette fonctionnalité est activé par défaut car certains clients ne comprennent pas les rejets avant cette commande RCPT TO.

La valeur par défaut fournit un bénéfice majeur : elle autorise Postfix à enregistrer l'adresse de destination lorsqu'il rejette une adresse ou un nom de client ou une adresse d'expédition, facilitant ainsi l'analyse.

smtpd_discard_ehlo_keyword_address_maps (défaut : vide)

Tables de correspondances, indexées par adresses IP des clients SMTP extérieurs, contenant une liste insensible à la casse des mots-clefs EHLO (pipelining, starttls, auth, etc.) que le serveur SMTP n'enverra pas dans les réponses EHLO aux clients SMTP distants. Voir smtpd_discard_ehlo_keywords pour les détails. Cette table n'est pas consulté avec les noms de machine pour des raisons de robustesse.

Cette fonctionnalité est disponibles dans les versions 2.3 et supérieures de Postfix.

smtpd_discard_ehlo_keywords (défaut : vide)

Liste insensible à la casse des mots-clefs EHLO (pipelining, starttls, auth, etc.) que le serveur SMTP n'enverra pas dans les réponse EHLO aux clients SMTP distants.

Cette fonctionnalité est disponibles dans les versions 2.3 et supérieures de Postfix.

Notes :

- ◇ Utilisez le pseudo mot-clef **silent-discard** pour éviter l'enregistrement de ces actions dans les journaux.

- ◇ Utilisez le paramètre smtpd_discard_ehlo_keyword_address_maps pour écarter les mots-clefs EHLO selectivement.

smtpd_end_of_data_restrictions (défaut : vide)

Restrictions d'accès optionnelles que le serveur SMTP de Postfix applique dans le contexte de la commande SMTP END-OF-DATA.

Cette fonctionnalité est disponibles dans les versions 2.3 et supérieures de Postfix.

Voir smtpd_data_restrictions pour les détails à propos de la syntaxe.

smtpd_enforce_tls (défaut : no)

Annonce le support STARTTLS et requiert que les clients SMTP distants utilisent le chiffrement TLS. En accord avec la RFC 2487 ceci NE DOIT PAS être utilisé sur un serveur SMTP référencé public. Cette option est désactivée par défaut et ne doit que rarement être utilisée.

Note 1 : cette option implique "smtpd_tls_auth_only = yes".

Note 2 : lorsqu'il est invoqué par "sendmail -bs", Postfix n'offrira jamais STARTTLS en raison de privilèges insuffisants pour accéder à la clef privée du serveur.

smtpd_error_sleep_time (défaut : 1s)

Avec les versions 2.1 et supérieures de Postfix : la réponse du serveur SMTP n'est envoyée qu'après un délai lorsque le client a fait plus de \$smtpd_soft_error_limit et moins de \$smtpd_hard_error_limit erreurs, sans livrer de courrier.

Avec les versions 2.0 et antérieures de Postfix : le serveur SMTP attend avant d'envoyer une réponse de rejet (codes 4xx ou 5xx), lorsque le client a fait moins de \$smtpd_soft_error_limit erreurs sans livrer de courrier.

smtpd_etrn_restrictions (défaut : vide)

Restrictions d'accès optionnelles du serveur SMTP pour les clients présentant une requête ETRN.

L'implémentation de ETRN de Postfix accepte seulement les destinations éligibles au service "fast flush". Consultez la page ETRN README pour plus de détails.

Indiquez une liste de restrictions, séparé par des virgules et/ou des espaces. Continuez les lignes longues en commençant la ligne suivante par des espaces. Les restrictions sont appliquées dans l'ordre indiqué; la première restriction qui correspond gagne.

Les restrictions suivantes sont spécifiques à l'information de nom de domaine passée avec la commande ETRN.

check_etrn_access type:table

Cherche dans la base indiquée le nom du domaine ETRN ou l'un de ses domaines parents.

Consultez la page de manuel access(5) pour plus de détails.

Autres restrictions valides dans ce contexte :

- ◇ Restrictions génériques pouvant être utilisées dans tous les contextes de commandes SMTP, décrites au paragraphe smtpd_client_restrictions.
- ◇ Restrictions spécifiques aux commandes SMTP décrites aux paragraphes smtpd_client_restrictions et smtpd_helo_restrictions.

Exemple :

```
smtpd_etrn_restrictions = permit mynetworks, reject
```

smtpd_expansion_filter (défaut : voir la sortie de "postconf -d")

Caractères autorisés dans les substitutions \$name des modèles de réponse RBL. Les caractères non autorisés sont remplacés par "_". Utilisez les caractères d'échappement type C pour indiquer des caractères spéciaux tels les espaces.

Ce paramètre n'est pas sujet aux substitutions \$paramètre.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

smtpd_forbidden_commands (défaut : CONNECT, GET, POST)

Liste des commandes pour lesquelles le serveur SMTP de Postfix interrompt immédiatement la session avec un code 221. Ceci peut être utilisé pour déconnecter les clients qui tentent manifestement d'abuser du système. En complément des commandes listées dans ce paramètre, les commandes qui suivent le format "Label:" des en-têtes de message entraîneront également une déconnexion.

Cette fonctionnalité est disponibles dans les versions 2.3 et supérieures de Postfix.

smtpd_hard_error_limit (défaut : 20)

Nombre maximum d'erreurs qu'un client SMTP distant est autorisé à commettre sans livrer de message. Le serveur SMTP de Postfix coupe la connexion au delà de cette limite.

smtpd_helo_required (défaut : no)

Impose au client SMTP de démarrer la session SMTP par une commande HELO ou EHLO.

Exemple :

`smtpd_helo_required = yes`

smtpd_helo_restrictions (défaut : vide)

Restrictions optionnelles que le serveur SMTP de Postfix applique dans le contexte de la commande SMTP HELO.

Tout est autorisé par défaut.

Indiquez une liste de restrictions, séparé par des virgules et/ou des espaces. Continuez les lignes longues en commençant la ligne suivante par des espaces. Les restrictions sont appliquées dans l'ordre indiqué ; la première restriction qui correspond gagne.

Les restrictions suivantes s'appliquent au nom de machine passé avec la commande HELO or EHLO.

check_helo_access type:table

Cherche dans la base access(5) indiquée le nom de machine ou domaine parent et exécute l'action correspondante.

check_helo_mx_access type:table

Cherche dans la base access(5) indiquée la machine MX correspondant au nom de machine passé avec HELO ou EHLO et exécute l'action correspondante. Note : le résultat "OK" n'est pas autorisé pour des raisons de sécurité. A la place, utilisez DUNNO afin d'exclure des machines spécifiques depuis les listes noires. Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

check_helo_ns_access type:table

Cherche dans la base access(5) indiquée les serveurs DNS correspondant au nom de machine passé avec HELO ou EHLO et exécute l'action correspondante. Note : le résultat "OK" n'est pas autorisé pour des raisons de sécurité. A la place, utilisez DUNNO afin d'exclure des machines spécifiques depuis les listes noires. Cette fonctionnalité est disponible dans les

versions 2.1 et supérieures de Postfix.

reject_invalid_helo_hostname (avec Postfix < 2.3 *reject_invalid_hostname*)

Rejette les requêtes lorsque la syntaxe du nom de machine passé avec HELO ou EHLO est invalide.

Le code invalid_hostname_reject_code indique le code de réponse envoyé pour rejeter les requêtes (défaut : 501).

reject_non_fqdn_helo_hostname (avec Postfix < 2.3 *reject_non_fqdn_hostname*)

Rejette les requêtes lorsque le nom de machine passé avec HELO ou EHLO n'est pas sous la forme pleinement qualifiée, comme requis par la RFC.

Le paramètre non_fqdn_reject_code indique le code de réponse utilisé pour rejeter les requêtes (défaut : 504).

reject_unknown_helo_hostname (avec Postfix < 2.3 *reject_unknown_hostname*)

Rejette les requêtes lorsque le nom de machine passé avec HELO ou EHLO ne correspond à aucun enregistrement DNS de type A ou MX.

Le code unknown_hostname_reject_code indique le code de réponse utilisé pour rejeter les requêtes (défaut : 450).

Autres restrictions valides dans ce contexte :

- ◇ Restrictions génériques pouvant être utilisées dans tous les contextes de commandes SMTP, décrites au paragraphe smtpd_client_restrictions.
- ◇ Restrictions sur le nom de machine ou l'adresse réseau décrites au paragraphe smtpd_client_restrictions.
- ◇ Restrictions spécifiques aux commandes SMTP décrites aux paragraphes smtpd_sender_restrictions et smtpd_recipient_restrictions. Lorsque les restrictions sur l'expéditeur ou le destinataire restrictions sont listés dans smtpd_helo_restrictions, elles n'ont d'effet que si "smtpd_delay_reject = yes", ainsi \$smtpd_helo_restrictions est évaluée lors de la réception de la commande RCPT TO.

Exemples :

smtpd_helo_restrictions = permit_mynetworks, reject_invalid_hostname

smtpd_helo_restrictions = permit_mynetworks, reject_unknown_hostname

smtpd_history_flush_threshold (défaut : 100)

Nombre maximum de lignes dans l'historique des commandes du serveur SMTP de Postfix avant qu'il soit vidé par la réception d'une commande EHLO, RSET, ou fin de DATA.

smtpd_junk_command_limit (défaut : 100)

Nombre de commandes parasites (NOOP, VRFY, ETRN or RSET) qu'un client SMTP distant peut envoyer avant que le serveur SMTP de Postfix ne commence à incrémenter le compteur d'erreurs à chaque commande parasite. Ce compteur est remis à zéro lorsqu'un message est livré. Consultez également les descriptions des paramètres de configuration smtpd_error_sleep_time et smtpd_soft_error_limit.

smtpd_milters (défaut : vide)

Liste des applications Milter (mail filter) pour le courrier entrant via le serveur smtpd(8). Voir MILTER_README pour plus de détails.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

smtpd_noop_commands (défaut : vide)

Liste des commandes pour lesquelles le serveur SMTP de Postfix répond "250 Ok", sans faire aucun contrôle de syntaxe ni changer d'état. Cette liste surcharge toute commande intégrée au serveur SMTP de Postfix.

smtpd_null_access_lookup_key (défaut : <>)

Clef de consultation utilisée dans les tables d'accès en lieu et place de l'adresse d'expédition nulle.

smtpd_peername_lookup (défaut : yes)

Tente de rechercher le nom de machine du client SMTP, et vérifie que le nom correspond à l'adresse IP du client. Le nom A du client est mis à "unknown" lorsqu'il ne peut être recherché ou vérifié ou lorsque la recherche est désactivée. Désactiver la recherche des noms réduit le délai causé par les requêtes DNS et augmente le taux de livraison entrante.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

smtpd_policy_service_max_idle (défaut : 300s)

Temps au delà duquel une connexion inactive à un service de politique SMTPD est close.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

smtpd_policy_service_max_ttl (défaut : 1000s)

Temps au delà duquel une connexion active à un service de politique SMTPD est close.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

smtpd_policy_service_timeout (défaut : 100s)

Temps limite pour établir une connexion avec serveur de politique SMTPD délégué.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

smtpd_proxy_ehlo (défaut : \$myhostname)

Annonce envoyée par le serveur SMTP de Postfix au filtre proxy. Par défaut, le nom de machine est utilisé.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

smtpd_proxy_filter (défaut : vide)

Nom de machine et port TCP du serveur proxy de filtrage. Le proxy reçoit tout le courrier du serveur SMTP de Postfix et est supposé donner le résultat à un autre processus serveur SMTP de Postfix.

Indiquez machine:port. La machine peut être indiquée sous la forme d'une adresse IP ou d'un nom symbolique ; aucune recherche MX n'est effectuée. Lorsqu'aucune machine n'est indiquée (ie ":port") la machine locale est utilisée.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

Les préfixes "inet:" et "unix:" sont disponibles avec les versions 2.3 et supérieures de Postfix.

smtpd_proxy_timeout (défaut : 100s)

Temps limite pour établir une connexion au filtre proxy et pour envoyer ou recevoir une information. Lorsqu'une connexion échoue, le client reçoit un message d'erreur générique pendant que plus de détails sont enregistrés dans les journaux.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

smtpd_recipient_limit (défaut : 1000)

Nombre maximum de destinataires qu'un serveur SMTP de Postfix accepte par requête de livraison de message.

smtpd_recipient_overshoot_limit (défaut : 1000)

Nombre de destinataires qu'un client SMTP distant peut envoyer au delà de la limite indiquée par le paramètre \$smtpd_recipient_limit, avant que le serveur SMTP de Postfix incrémente le compteur d'erreur par session pour chaque destinataire en excès.

smtpd_recipient_restrictions (défaut : permit_mynetworks, reject_unauth_destination)

Restrictions d'accès que le serveur SMTP de Postfix applique dans le contexte d'une commande RCPT TO.

Par défaut, le serveur SMTP de Postfix accepte :

- ◇ le courrier des clients dont l'adresse IP correspond à \$mynetworks, ou :
- ◇ le courrier pour les destinations extérieures qui correspondent à \$relay_domains, à l'exception des adresses qui contiennent un routage envoyé par l'expéditeur (user@elsewhere@domain), ou :
- ◇ le courrier à destination locale qui correspond à \$inet_interfaces ou \$proxy_interfaces, \$mydestination, \$virtual_alias_domains ou \$virtual_mailbox_domains.

IMPORTANT : si vous changez la valeur de ce paramètre, vous devez indiquer au moins une des restrictions suivantes, sinon Postfix refusera de recevoir du courrier :

reject, defer, defer_if_permit, reject_unauth_destination

Indiquez une liste de restrictions, séparé par des virgules et/ou des espaces. Continuez les lignes longues en commençant la ligne suivante par des espaces. Les restrictions sont appliquées dans l'ordre indiqué ; la première restriction qui correspond gagne.

Les restrictions suivantes sont spécifiques aux adresses de destination reçues avec la commande RCPT TOd.

check_recipient_access type:table

Cherche dans la table d'accès indiquée l'adresse, le domaine ou ses domaines parents ou la partie locale et exécute l'action correspondante.

check_recipient_mx_access type:table

Cherche dans la table d'accès indiquée la machine MX correspondant à l'adresse reçue avec RCPT TO et exécute l'action correspondante. Note : le résultat "OK" n'est pas autorisé pour des raisons de sécurité. A la place, utilisez DUNNO pour exclure des machines spécifiques d'une liste noire. Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

check_recipient_ns_access type:table

Cherche dans la table d'accès indiquée les serveurs DNS correspondant à l'adresse reçue avec RCPT TO et exécute l'action correspondante. Note : le résultat "OK" n'est pas autorisé pour des raisons de sécurité. A la place, utilisez DUNNO pour exclure des machines spécifiques d'une liste noire. Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

permit_auth_destination

Autorise la requête lorsque l'une des propositions suivantes est vraie :

- Postfix est un relai : l'adresse résolue RCPT TO correspond à \$relay_domains ou l'un de leurs sous-domaines et l'adresse ne contient pas de routage expéditeur (user@elsewhere@domain),
- Postfix est la destination finale : l'adresse résolue RCPT TO correspond à \$mydestination, \$inet_interfaces, \$proxy_interfaces, \$virtual_alias_domains ou \$virtual_mailbox_domains et l'adresse ne contient pas de routage expéditeur (user@elsewhere@domain).

permit_mx_backup

Autorise la requête lorsque le système de messagerie locale est la machine MX pour l'adresse RCPT TO, ou lorsque l'adresse est une destination autorisée (voir permit_auth_destination

pour la définition).

- Sûreté : `permit_mx_backup` n'accepte pas les adresses contenant des informations de routage (exemple : `user@elsewhere@domain`).
- Sûreté : `permit_mx_backup` peut être vulnérable aux abus si l'accès n'est pas restreint avec `permit_mx_backup_networks`.
- Sûreté : depuis la version 2.3 de Postfix, `permit_mx_backup` n'accepte plus l'adresse lorsque le système local est la machine MX primaire du domaine de destination. Exception : `permit_mx_backup` accepte l'adresse lorsqu'elle correspond à une destination autorisée (voir `permit_auth_backup` pour la définition).
- Limitation : le courrier pourrait être rejeté en cas de problème DNS temporaire avec les versions de Postfix antérieures à la 2.0.

reject_non_fqdn_recipient

Rejette la requête lorsque l'adresse RCPT TO n'est pas de forme pleinement qualifiée, comme requis par la RFC.

Le paramètre `reject_non_fqdn_reject_code` indique le code de réponse pour les requêtes rejetées (défaut : 504).

reject_rhsbl_recipient rbl_domain=d.d.d.d

Rejette la requête lorsque le domaine RCPT TO est listé avec l'enregistrement A "`d.d.d.d`" du domaine `rbl_domain` (Postfix version 2.1 et supérieures seulement). Si aucun "`d.d.d.d`" n'est indiqué, rejette la requête lorsque la résolution de l'adresse réseau du client est listée avec un enregistrement de type A dans le domaine `rbl_domain`.

Le paramètre `maps_rbl_reject_code` indique le code de réponse pour les requêtes rejetées (défaut : 554) ; le paramètre `default_rbl_reply` indique la réponse par défaut du serveur et le paramètre `rbl_reply_maps` indique les tables contenant les réponses indexées par `rbl_domain`. Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

reject_unauth_destination

Rejette la requête sauf si l'une des propositions suivantes est vraie :

- Postfix transfère les messages : l'adresse RCPT TO résolue correspond à `$relay_domains` ou l'un de leurs sous-domaines et ne contient pas de routage expéditeur (`user@elsewhere@domain`),
- Postfix est la destination finale : l'adresse RCPT TO résolue correspond à `$mydestination`, `$inet_interfaces`, `$proxy_interfaces`, `$virtual_alias_domains`, ou `$virtual_mailbox_domains`, et ne contient pas de routage expéditeur (`user@elsewhere@domain`).

Le paramètre `relay_domains_reject_code` indique le code de réponse pour les requêtes rejetées (défaut : 554).

reject_unknown_recipient_domain

Rejette la requête lorsque l'adresse RCPT TO ne correspond à aucun enregistrement DNS de type A ou MX et Postfix n'est pas la destination finale de l'adresse de destination.

Le paramètre `unknown_address_reject_code` indique le code de réponse pour les requêtes rejetées (défaut : 450). La réponse est toujours 450 en cas d'erreur DNS temporaire.

***reject_unlisted_recipient* (avec Postfix 2.0 : `check_recipient_maps`)**

Rejette la requête lorsque l'adresse RCPT TO n'est pas listée dans la liste des destinataires valides pour ses classes de domaines. Reportez-vous au paragraphe concernant le paramètre `smtpd_reject_unlisted_recipient` pour plus de détails. Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

reject_unverified_recipient

Rejette la requête lorsque le courrier à destination de l'adresse RCPT TO est réputé non-livrable ou lorsque l'adresse de destination est réputée non-joignable. Les informations de vérification d'adresse sont gérées par le serveur `verify(8)` ; reportez-vous à la page [ADDRESS VERIFICATION README](#) pour plus de détails.

Le paramètre unverified_recipient_reject_code indique le code de réponse pour une adresse réputée non-livrable (défaut : 450, changez le en 550 si vous savez ce que vous faites). Postfix répond toujours avec le code 450 lorsqu'une vérification d'adresse échoue suite à un problème temporaire. Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

Autres restrictions valides dans ce contexte :

- ◇ Restrictions génériques pouvant être utilisées dans tous les contextes de commandes SMTP, décrites au paragraphe smtpd_client_restrictions.
- ◇ Restrictions spécifiques aux commandes SMTP décrites aux pages smtpd_client_restrictions, smtpd_helo_restrictions et smtpd_sender_restrictions.

Exemple :

```
smtpd_recipient_restrictions = permit_mynetworks, reject_unauth_destination  
smtpd_reject_unlisted_recipient (défaut : yes)
```

Requiert que le serveur SMTP de Postfix rejette le courrier des adresses de destination inconnues, même si aucune restriction d'accès reject_unlisted_recipient n'est indiquée. Ceci évite des surcharger la file d'attente avec des avis de rejet MAILER-DAEMON.

- ◇ Le domaine du destinataire correspond à \$mydestination, \$inet_interfaces ou \$proxy_interfaces, mais le destinataire n'est pas listé dans \$local_recipient_maps et \$local_recipient_maps n'est pas nul.
- ◇ Le domaine du destinataire correspond à \$virtual_alias_maps mais le destinataire n'est pas listé dans \$virtual_alias_maps.
- ◇ Le domaine du destinataire correspond à \$virtual_mailbox_domains mais le destinataire n'est pas listé dans \$virtual_mailbox_maps et \$virtual_mailbox_maps n'est pas nul.
- ◇ Le domaine du destinataire correspond à \$relay_domains mais le destinataire n'est pas listé dans \$relay_recipient_maps et \$relay_recipient_maps n'est pas nul.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

```
smtpd_reject_unlisted_sender (défaut : no)
```

Requiert que le serveur SMTP de Postfix rejette le courrier des adresses d'expédition inconnues, même si aucune restriction d'accès reject_unlisted_sender n'est indiquée. Ceci peut ralentir une explosion de messages forgés par des vers ou virus.

- ◇ Le domaine de l'expéditeur correspond à \$mydestination, \$inet_interfaces ou \$proxy_interfaces, mais le destinataire n'est pas listé dans \$local_recipient_maps et \$local_recipient_maps n'est pas nul.
- ◇ Le domaine de l'expéditeur correspond à \$virtual_alias_maps mais le destinataire n'est pas listé dans \$virtual_alias_maps.
- ◇ Le domaine de l'expéditeur correspond à \$virtual_mailbox_domains mais le destinataire n'est pas listé dans \$virtual_mailbox_maps et \$virtual_mailbox_maps n'est pas nul.
- ◇ Le domaine de l'expéditeur correspond à \$relay_domains mais le destinataire n'est pas listé dans \$relay_recipient_maps et \$relay_recipient_maps n'est pas nul.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

```
smtpd_restriction_classes (défaut : vide)
```

Alias définis par l'utilisateur pour les groupes de restriction d'accès. Les alias peuvent être indiqué dans smtpd_recipient_restrictions etc., et dans la partie droite d'une table d'accès de Postfix.

Une des applications majeures est l'implémentation de contrôle anti-spam par destinataire. Voyez la page RESTRICTION CLASS README pour les autres exemples.

```
smtpd_sasl_application_name (défaut : smtpd)
```

Documentation de Postfix en français

Nom de l'application utilisée pour l'initialisation du serveur SASL. Ceci contrôle le nom du fichier de configuration SASL. La valeur par défaut est **smtpd**, correspondant au fichier de configuration SASL nommé **smtpd.conf**.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

smtpd_sasl_auth_enable (défaut : no)

Active l'authentification SASL dans le serveur SMTP de Postfix. Par défaut, le serveur SMTP de Postfix n'utilise pas d'authentification.

Si un client SMTP distant est authentifié, la restriction d'accès permit_sasl_authenticated peut être utilisée pour permettre le relai, comme ci-dessous :

```
smtpd_recipient_restrictions =  
    permit_mynetworks, permit_sasl_authenticated, ...
```

Pour rejeter toutes les connections des clients non authentifiés, utilisez "smtpd_delay_reject = yes" (valeur par défaut) et utilisez :

```
smtpd_client_restrictions = permit_sasl_authenticated, reject
```

Reportez-vous à la page SASL README pour plus de détails.

smtpd_sasl_authenticated_header (défaut : no)

Reporte le nom d'utilisateur SASL authentifié dans l'en-tête de message Received du serveur smtpd(8).

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

smtpd_sasl_exceptions_networks (défaut : vide)

Clients SMTP pour lesquels Postfix ne proposera pas le support AUTH.

Certains clients (Netscape 4 entre autres) ont un bug qui impose un login et un mot-de-passe dès lors que AUTH est proposé, même si ce n'est pas nécessaire. Pour contourner ce problème, indiquez par exemple \$mynetworks pour ne pas proposer AUTH aux clients locaux.

Indiquez une liste d'expression adresse/réseau, séparé par des virgules et/ou des espaces. Vous pouvez également indiquer des noms de machines ou des noms de domaines (s'ils sont précédés d'un "." tous les sous-domaines sont pris en compte), des "/nom/de/fichier" ou des expressions "type:table". Un "/nom/de/fichier" de correspondances est remplacé par son contenu; une table de correspondances "type:table" correspond lorsqu'une entrée de la table correspond (le résultat de la consultation est ignoré). Continuez les lignes longues en commençant la ligne suivante par des espaces.

Note : les adresses IP version 6 doivent être indiquées entre [] dans smtpd_sasl_exceptions_networks et dans les fichier indiqués avec "/nom/de/fichier". Les adresses IP version 6 contiennent le caractère ":" et pourraient être confondues avec une expression "type:table".

Exemple :

```
smtpd_sasl_exceptions_networks = $mynetworks
```

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

smtpd_sasl_local_domain (défaut : vide)

Nom de royaume d'authentification SASL local.

Par défaut, cette valeur est vide.

Exemples :

```
smtpd_sasl_local_domain = $mydomain  
smtpd_sasl_local_domain = $myhostname
```

smtpd_sasl_path (défaut : smtpd)

Information spécifique à l'implémentation passée au plug-in SASL sélectionné par **smtpd_sasl_type**. Généralement, ceci indique le nom d'un fichier de configuration ou d'un point de rendez-vous.

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

smtpd_sasl_security_options (défaut : noanonymous)

Options de sécurité SASL ; depuis la version 2.3 de Postfix 2.3, la liste des options dépend de l'implémentation du serveur SASL sélectionné par **smtpd_sasl_type**.

Les options de sécurité suivantes sont définies pour l'implémentation SASL **cyrus** :

noplaintext

Interdit les méthodes utilisant les mots de passe en clair.

noactive

Interdit les méthodes sujettes à une attaque active (sans dictionnaire).

nodictionary

Interdit les méthodes sujettes à une attaque passive (par dictionnaire).

noanonymous

Interdit les méthodes qui autorisent l'authentification anonyme.

mutual_auth

N'autorise que les méthodes fournissant une authentification mutuelle (non disponible avec SASL version 1).

Par défaut, le serveur SMTP de Postfix accepte les mots de passe en clair mais pas les logins anonymes.

Attention : il apparaît que les clients essaient les méthodes d'authentification dans l'ordre indiqué par le serveur (e.g., PLAIN ANONYMOUS CRAM-MD5), ce qui signifie que si vous désactivez les mots-de-passe en clair, ils tenteront l'authentification anonyme même s'ils sont aptes à utiliser CRAM-MD5. Ainsi, si vous désactivez les mots de passe en clair, désactivez les logins anonymes également. Postfix traite ces derniers comme s'il n'y avait pas eu d'authentification.

Exemple :

```
smtpd_sasl_security_options = noanonymous, noplaintext
```

smtpd_sasl_tls_security_options (défaut : \$smtpd_sasl_security_options)

Options de sécurité de l'authentification SASL que le serveur SMTP de Postfix utilise pour les sessions SMTP chiffrées par TLS.

smtpd_sasl_type (défaut : cyrus)

Type de plug-in SASL que le serveur SMTP de Postfix doit utiliser pour l'authentification. Les types disponibles sont affichés par la commande "**postconf -a**".

Cette fonctionnalité est disponible dans les versions 2.3 et supérieures de Postfix.

smtpd_sender_login_maps (défaut : vide)

Tables optionnelles de correspondances entre le login SASL et les adresses d'expéditeur (MAIL FROM).

Indiquez zéro ou plus de table de correspondances "type:table". Avec des tables indexées telles les fichiers DB or DBM, ou des tables réseau telles NIS, LDAP or SQL, les opérations de recherche sont effectuées avec l'adresse d'expédition of *user@domain*:

1) *user@domain*

Cette recherche est toujours faite et à la préférence la plus élevée.

2) *user*

Cette recherche est faite seulement lorsque la partie *domaine* de l'adresse d'expédition correspond à \$myorigin, \$mydestination, \$inet_interfaces ou \$proxy_interfaces.

3) *@domain*

Cete recherche est faite en dernier avec la préférence la plus faible.

Dans tous les cas, le résultat de la consultation doit toujours être "non trouvé" ou une liste des logins SASL séparés par des virgules et/ou des espaces.

smtpd_sender_restrictions (défaut : vide)

Restrictions optionnelles que le serveur SMTP de Postfix applique dans le contexte des commandes MAIL FROM.

Tout est permit par défaut.

Indiquez une liste de restrictions, séparées par des virgules et/ou des espaces. Continuez les lignes longues en commençant la ligne suivante par des espaces. Les restrictions sont appliquées dans l'ordre indiqué; la première restriction qui correspond gagne.

Les restrictions suivantes s'appliquent à l'adresse d'expédition reçue avec la commande MAIL FROM.

check_sender_access type:table

Recherche dans la table d'accès indiquée l'adresse MAIL FROM, son domaine ou un domaine parent, ou la parit locale@, et exécute l'action correspondante.

check_sender_mx_access type:table

Recherche dans la table d'accès(5) indiquée les machines MX correspondant à l'adresse MAIL FROM et exécute l'action correspondante. Note : le résultat "OK" n'est pas autorisé pour des raisons de sécurité. A la palce, utilisez DUNNO pour exclure certaines machines des listes noires. Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

check_sender_ns_access type:table

Recherche dans la table d'accès(5) indiquée for the serveurs DNS for the adresse MAIL FROM, and exécute l'action correspondante. Note : le résultat "OK" n'est pas autorisé pour des raisons de sécurité. A la place, utilisez DUNNO pour exclure des machines specifiques des listes noires. Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

reject_authenticated_sender_login_mismatch

Renforce la restriction reject_sender_login_mismatch pour les clients authentifiés uniquement. Cette fonctionnalité est disponible est disponible dans les versions 2.1 et supérieures de Postfix.

reject_non_fqdn_sender

Rejette la requête lorsque l'adresse MAIL FROM n'est pas sous la forme pleinement qualifiée requise par la RFC.

Le paramètre non_fqdn_reject_code indique le code de réponse pour les requêtes rejetées (défaut : 504).

reject_rhsbl_sender rbl_domain=d.d.d.d

Rejette la requête lorsque le domaine MAIL FROM est listé avec l'enregistrement A "*d.d.d.d*" du domaine *rbl_domain* (Postfix version 2.1 et supérieures seulement). Si aucun "*d.d.d.d*" n'est indiqué, rejette la requête lorsque la résolution inverse de l'adresse réseau du client est listée avec un enregistrement de type A dans le domaine *rbl_domain*.

Le paramètre maps_rbl_reject_code indique le code de réponse pour les requêtes rejetées (défaut : 554); Le paramètre default_rbl_reply indique la réponse par défaut du serveur ; et le paramètre rbl_reply_maps indique les tables contenant les réponses du serveur indexées par domaine *rbl_domain*. Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

reject_sender_login_mismatch

Rejette la requête lorsque \$smtpd_sender_login_maps indique un propriétaire pour l'adresse MAIL FROM, mais le client n'est pas logué (SASL) avec ce compte ou lorsque le client est logué (SASL) mais que le login utilisé ne possède pas l'adresse MAIL FROM au regard de \$smtpd_sender_login_maps.

reject_unauthenticated_sender_login_mismatch

Renforce la restriction reject_sender_login_mismatch pour les utilisateurs non authentifié seulement. Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

reject_unknown_sender_domain

Rejette la requête lorsque Postfix n'est pas la destination finale de l'adresse d'expédition et que l'adresse MAIL FROM n'a pas d'enregistrement DNS A ou MX correspondant, ou lorsque cet enregistrement MX est malformé comme un nom MX de longueur nulle (Postfix versions 2.3 et supérieures).

Le paramètre unknown_address_reject_code indique le code de réponse pour les requêtes rejetées (défaut : 450). La réponse est toujours 450 en cas d'erreur temporaire DNS.

reject_unlisted_sender

Rejette la requête lorsque l'adresse MAIL FROM n'est pas listée dans la liste des destinataires valides de sa classe de domaines. Reportez-vous à la description du paramètre smtpd_reject_unlisted_sender pour plus de détails. Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

reject_unverified_sender

Rejette la requête lorsque le courrier à destination de l'adresse MAIL FROM est connu pour être rejeté, ou lorsque la destination de l'adresse d'expédition destination n'est pas joignable. Les informations de vérification d'adresse sont gérées par le serveur verify(8) ; lisez la page ADDRESS VERIFICATION README pour plus de détails.

Le paramètre unverified_sender_reject_code indique le code de réponse lorsque l'adresse est connue pour être rejetée (défaut : 450, changez le en 550 si vous savez ce que vous faites). Postfix répond avec le code 450 lorsque la vérification d'adresse échoue en raison d'un problème temporaire. Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

Autres restrictions valides dans ce contexte :

- ◇ Restrictions génériques pouvant être utilisées dans tous les contextes de commandes SMTP, décrites au paragraphe smtpd_client_restrictions.
- ◇ Restrictions spécifiques aux commandes SMTP décrites aux paragraphes smtpd_client_restrictions et smtpd_helo_restrictions.
- ◇ Restrictions spécifiques aux commandes SMTP décrites aux paragraphes smtpd_recipient_restrictions. Lorsque les restrictions sur les destinataires sont listés dans le paramètre smtpd_sender_restrictions, elles n'ont d'effet que si "smtpd_delay_reject = yes", ainsi \$smtpd_sender_restrictions est évaluée à la réception de la commande RCPT TO.

Exemples :

Documentation de Postfix en français

```
smtpd_sender_restrictions = reject_unknown_sender_domain  
smtpd_sender_restrictions = reject_unknown_sender_domain,  
check_sender_access hash:/etc/postfix/access
```

smtpd_soft_error_limit (défaut : 10)

Nombre d'erreurs qu'un client SMTP distant est autorisé à faire sans livrer de messages avant d'être ralenti par le serveur SMTP de Postfix.

◇ Avec Postfix version 2.1 et supérieures, le serveur SMTP de Postfix ralentit toutes les réponses de \$smtpd_error_sleep_time secondes.

◇ Avec Postfix version 2.0 et antérieures, le serveur SMTP de Postfix ralentit toutes les réponses de (nombre d'erreurs) secondes.

smtpd_starttls_timeout (défaut : 300s)

Temps limite pour les opérations de lecture et d'écriture du serveur SMTP de Postfix durant les procédures de démarrage et d'arrêt TLS.

smtpd_timeout (défaut : 300s)

Temps limite pour que le serveur SMTP de Postfix envoie une réponse et pour que le client SMTP envoie une requête.

Note : si vous indiquez des temps limites très élevés, vous devrez augmenter le paramètre global ipc_timeout.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

smtpd_tls_CAfile (défaut : vide)

Fichier contenant le certificat de l'autorité de certification de laquelle est issu le certificat du serveur SMTP de Postfix. Ce n'est nécessaire que si ce certificat d'autorité n'est pas déjà présent dans le fichier du certificat du serveur. Ce fichier peut également contenir les certificats des autres autorités reconnues. Vous devez utiliser ce fichier pour lister les autorités de confiance si vous voulez utiliser la mise en cage chroot.

Exemple :

```
smtpd_tls_CAfile = /etc/postfix/CAcert.pem
```

smtpd_tls_CApath (défaut : vide)

Répertoire contenant les certificats des autorités que le serveur SMTP de Postfix offrira aux clients SMTP distants pour la vérification du certificat client. N'oubliez pas de créer les liens "hash" nécessaires, avec par exemple "\$OPENSSL_HOME/bin/c_rehash /etc/postfix/certs".

Pour utiliser cette option en mode chroot, ce répertoire (ou une copie) doit être dans la cage chroot. Notez que dans ce cas, les certificats des autorités ne sont pas offerts au client, ainsi les clients type Netscape ne proposeront pas les certificats issus de ces autorités. L'utilisation de cette fonctionnalité n'est généralement pas recommandée.

Exemple :

```
smtpd_tls_CApath = /etc/postfix/certs
```

smtpd_tls_ask_ccert (défaut : no)

Demande au client SMTP distant un certificat client. Cette information est nécessaire pour le relais de messages basé sur les certificats, par exemple avec l'option permit_tls_clientcerts.

Certains clients tels Netscape se plaindront si aucun certificat n'est disponible (pour la liste des autorités de /etc/postfix/certs) ou offriront plusieurs certificats client au choix. Pour éviter les gênes,

cette option est désactivée par défaut.

smtpd_tls_auth_only (défaut : no)

Lorsque le chiffrement TLS est optionnel dans le serveur SMTP de Postfix, n'annonce ou n'accepte pas les authentifications SASL dans une connexion non chiffrée.

smtpd_tls_ccert_verifydepth (défaut : 5)

Profondeur de vérification des certificats des client SMTP extérieurs. Une profondeur 1 est suffisante si l'autorité est listée dans un fichier local d'autorités. La valeur par défaut devrait suffir pour les chaînes plus longues (l'autorité racine certifie une autorité particulière qui elle-même certifie le certificat présenté...).

smtpd_tls_cert_file (défaut : vide)

Fichier contenant le certificat RSA du serveur SMTP de Postfix au format PEM. Ce fichier peut également contenir la clef privée du serveur.

Les certificats RSA et DSA sont tous deux supportés. Lorsque les deux types sont présents, le chiffrement utilisé détermine lequel présenter au client. Pour les clients Netscape et les clients ouverts SSL sans choix spécifique de chiffrement, le certificat RSA est choisit.

Pour vérifier un certificat, le certificat de l'autorité (dans le cas d'une chaîne de certification, tous les certificats d'autorité) doivent être disponibles. Vous devrez ajouter ces certificats au fichier du certificat du serveur, le certificat du serveur en premier puis les autorités reconnues.

Exemple : le certificat de "server.dom.ain" est issu de l'autorité "intermediate CA" issue elle-même de l'autorité "root CA". Créez le fichier server.pem avec
"cat server_cert.pem intermediate_CA.pem root_CA.pem > server.pem".

Si vous voulez accepter les certificats issus de ces autorités, vous pouvez également ajouter ces certificats d'autorité dans le fichier smtpd_tls_CAfile, auquel cas il n'est pas nécessaire de les copier dans le fichier smtpd_tls_dcrt_file ou smtpd_tls_cert_file.

Un certificat copié ici doit être utilisable comme certificat serveur SSL et donc passer le test "openssl verify -purpose sslserver ...".

Exemple :

```
smtpd_tls_cert_file = /etc/postfix/server.pem
```

smtpd_tls_cipherlist (défaut : vide)

Contrôle le schéma de sélection du chiffrement TLS du serveur SMTP de Postfix. Pour plus de détails, lisez la documentation d'OpenSSL. Note : n'utilisez pas de guillemets "" autour de la valeur de ce paramètre.

smtpd_tls_dcrt_file (défaut : vide)

Fichier contenant le certificat DSA du serveur SMTP de Postfix au format PEM. Ce fichier peut également contenir la clef privée du serveur.

Reportez-vous au paragraphe smtpd_tls_cert_file pour plus de détails.

Exemple :

```
smtpd_tls_dcrt_file = /etc/postfix/server-dsa.pem
```

smtpd_tls_dh1024_param_file (défaut : vide)

Fichier contenant les paramètres DH (Diffie-Hellmann) que le serveur SMTP de Postfix doit utiliser avec le chiffrement EDH.

Documentation de Postfix en français

Au lieu d'utiliser exactement le même ensemble de paramètres que celui distribué avec les autres packages TLS, il est plus sécurisé de générer votre propre ensemble avec une commande ressemblant à :

```
openssl gendh -out /etc/postfix/dh_1024.pem -2 -rand /var/run/egd-pool 1024
```

Votre source actuelle d'entropie peut différer. Certains systèmes disposent de `/dev/random` ; sur d'autres, vous devrez utiliser le démon "Entropy Gathering Daemon EGD" disponible à l'adresse suivante : <http://www.lothar.com/tech/crypto/>.

Exemple :

```
smtpd_tls_dh1024_param_file = /etc/postfix/dh_1024.pem
```

***smtpd_tls_dh512_param_file* (défaut : vide)**

Fichier contenant les paramètres DH (Diffie–Hellmann) que le serveur SMTP de Postfix doit utiliser avec le chiffrement EDH.

Reportez-vous à la description du paramètre de configuration smtpd_tls_dh1024_param_file ci-dessus.

Exemple :

```
smtpd_tls_dh512_param_file = /etc/postfix/dh_512.pem
```

***smtpd_tls_dkey_file* (défaut : \$smtpd_tls_dcert_file)**

Fichier contenant la clef privée du serveur SMTP de Postfix au format PEM. Ce fichier peut être le même que le fichier contenant le certificat \$smtpd_tls_dcert_file.

La clef privée ne doit pas être chiffrée. En d'autres termes, la clef doit être accessible sans mot-de-passe.

***smtpd_tls_key_file* (défaut : \$smtpd_tls_cert_file)**

Fichier contenant la clef privée RSA du serveur SMTP de Postfix au format PEM. Ce fichier peut être le même que celui contenant le certificat \$smtpd_tls_cert_file.

La clef privée ne doit pas être chiffrée. En d'autres termes, la clef doit être accessible sans mot-de-passe.

***smtpd_tls_loglevel* (défaut : 0)**

Active l'enregistrement additionnel de l'activité TLS du serveur SMTP de Postfix. Chaque niveau inclut également les informations des niveaux inférieurs.

- 0 Désactive l'enregistrement de l'activité TLS.
- 1 Enregistre les informations concernant la négociation et les certificat.
- 2 Enregistre les niveaux durant la négociation TLS.
- 3 Enregistre la copie hexadécimale et ASCII du processus de négociation TLS.
- 4 Enregistre également la retranscription complète hexadécimale et ASCII de la session après STARTTLS.

Utilisez "smtpd_tls_loglevel = 3" seulement en cas de problèmes. L'utilisation du niveau 4 est vivement déconseillée.

***smtpd_tls_received_header* (défaut : no)**

Requiert que le serveur SMTP de Postfix produise des en-têtes de message Received: qui incluent les informations à propos du protocole et du chiffrement utilisé ainsi que les champs CommonName des certificats client et de l'autorité dont il est issu. Ceci est désactivé par défaut, l'information étant susceptible d'être modifiée pendant le transit dans d'autres serveurs de messagerie. Seules

l'information enregistrée pas la destination finale est fiable.

smtpd_tls_req_ccert (défaut : no)

Lorsque le chiffrement TLS est imposé, requiert un certificat du client SMTP pour autoriser la connexion TLS. Ceci implique "smtpd_tls_ask_ccert = yes".

Lorsque le chiffrement TLS est optionnel, les clients SMTP distants peuvent outrepasser cette restriction simplement en n'utilisant pas STARTTLS. Pour cette raison, une connexion TLS sera utilisée ainsi seulement si "smtpd_tls_ask_ccert = yes".

smtpd_tls_session_cache_database (défaut : vide)

Nom du fichier contenant le cache optionnel des sessions TLS du serveur SMTP de Postfix. Utilisez un type de table qui supporte l'énumération tel **btree** ou **sdbm** ; il n'est pas nécessaire de supporter les accès concurrents. Le fichier est créé s'il n'existe pas déjà.

Note : les bases de données **dbm** ne sont pas utilisables ici car les objets TLS sont trop gros.

Exemple :

```
smtpd_tls_session_cache_database = btree:/var/postfix/smtpd_scache
```

smtpd_tls_session_cache_timeout (défaut : 3600s)

Délai d'expiration des informations du cache de session TLS du serveur SMTP de Postfix. Le nettoyage du cache est effectué périodiquement toutes les \$smtpd_tls_session_cache_timeout secondes.

smtpd_tls_wrappermode (défaut : no)

Lance le serveur SMTP de Postfix dans le mode non-standard "wrapper", au lieu d'utiliser la commande STARTTLS.

Si vous voulez supporter ce service, activez un port particulier dans le fichier master.cf, et indiquez "-o smtpd_tls_wrappermode=yes" sur la ligne de commande du serveur SMTP. Le port 465 (smtps) a été choisi dans ce but.

Note du traducteur : certains clients Microsoft n'ont pas une implémentation correcte de TLS et ne savent utiliser que ce mode.

smtpd_use_tls (défaut : no)

Mode opportuniste : annonce le support STARTTLS aux clients SMTP, mais ne l'exige pas.

Note : lorsqu'il est appelé par "**sendmail -bs**", Postfix n'offrira jamais STARTTLS en raison de privilèges insuffisants par accéder à la clef privée du serveur.

soft_bounce (défaut : no)

Filet de sécurité maintenant en file d'attente les messages qui devraient être retournés. Ce paramètre désactive les générations locales d'avis de rejet et évite au serveur SMTP de Postfix de rejeter toujours les messages en changeant les codes 5xx en 4xx. Toutefois, soft_bounce n'est pas un moyen de corriger les erreurs de réécriture d'adresses ou de routage.

Exemple :

```
soft_bounce = yes
```

stale_lock_time (défaut : 500s)

Temps au delà duquel un verrou exclusif de boîte-aux-lettres est supprimé. Utilisé pour la livraison en fichier ou dossier.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

strict_7bit_headers (défaut : no)

Rejette les messages contenant des en-têtes de message 8-bit. Bloque le courrier des applications pauvrement écrites.

Cette fonctionnalité ne doit pas être activée sur un serveur standard car cela peut rejeter du courrier légitime.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

strict_8bitmime (défaut : no)

Active strict_7bit_headers et strict_8bitmime_body.

Cette fonctionnalité ne doit pas être activée sur un serveur standard car cela peut rejeter du courrier légitime.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

strict_8bitmime_body (défaut : no)

Rejette les messages dont le corps est encodé en 8-bit sans informations d'encodage MIME. Ceci bloque les messages des applications pauvrement écrites.

Malheureusement, ceci rejette également les requêtes d'approbation du logiciel Majordomo lorsque les requêtes inclues contiennent un message MIME 8-bit valide, et rejette les avis de rejet des serveurs de messagerie qui n'encapsulent pas en MIME les contenus 8-bit (par exemple, les avis de rejet de qmail et des vieilles versions de Postfix).

Cette fonctionnalité ne doit généralement pas être activée car cela peut rejeter du courrier légitime.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

strict_mime_encoding_domain (défaut : no)

Rejette les messages contenant une information Content-Transfer-Encoding: invalide pour les types de contenu message/* ou multipart/*. Ceci bloque les messages des logiciels mal écrits.

Cette fonctionnalité ne doit généralement pas être activée car elle rejette le courrier après une simple violation.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

strict_rfc821_envelopes (défaut : no)

Requiert que les adresses reçues avec les commandes SMTP "MAIL FROM" et "RCPT TO" soient encadrées par <> et qu'elles contiennent des commentaires style RFC 822. Ceci stoppe le courrier des logiciels mal écrits.

Par défaut, le serveur SMTP de Postfix accepte la syntaxe RFC 822 dans les adresses MAIL FROM et RCPT TO.

sun_mailtool_compatibility (défaut : no)

Fonctionnalité obsolète de compatibilité avec l'outil mail SUN. A la place, utilisez "mailbox_delivery_lock = dotlock".

swap_bangpath (défaut : yes)

Active la réécriture de "site!user" en "user@site". Ceci est nécessaire si votre machine est connectée à des réseaux networks. Activé par défaut.

Note 2 : avec les versions 2.2 et supérieures de Postfix, la réécriture des adresses dans les en-têtes de message ne se produit que lorsque l'une des conditions suivantes est réalisée :

- ◇ le message est soumis par la commande sendmail(1) de Postfix,
- ◇ le message provient d'un client réseau qui correspond à \$local_header_rewrite_clients,
- ◇ le message provient du réseau et le paramètre remote_header_rewrite_domain contient une valeur non vide.

Pour retrouver le comportement des versions de Postfix antérieures à la 2.2, utilisez "local_header_rewrite_clients = static:all".

Exemple :

swap_banqpath = no

syslog_facility (défaut : mail)

Facilité syslog pour les journaux Postfix. Indiquez une facilité définie dans syslog.conf(5). La facilité par défaut est "mail".

Attention : une valeur syslog_facility différente de la valeur par défaut ne prend effet qu'après la fin de l'initialisation d'un processus Postfix. Les erreurs arrivant avant sont enregistré avec la facilité par défaut (problèmes de lecture du fichier main.cf, erreurs sur la ligne de commande,...).

syslog_name (défaut : postfix)

Nom du système de messagerie précédent le nom de processus dans les enregistrements syslog, ainsi "smtpd" devient "postfix/smtpd".

Attention : une valeur syslog_name différente de la valeur par défaut ne prend effet qu'après la fin de l'initialisation d'un processus Postfix. Les erreurs arrivant avant sont enregistré avec la facilité par défaut (problèmes de lecture du fichier main.cf, erreurs sur la ligne de commande,...).

tls_daemon_random_bytes (défaut : 32)

Nombre d'octets que les processus smtpd(8) et smtpd(8) demandent au server tlsmgr(8) pour égrainer son générateur de nombres pseudo-aléatoires (PRNG). La valeur par défaut de 32 octets (équivalent à 256 bits) est suffisante pour générer une clef de session de 128bit (ou 168bit pour 3DES).

tls_random_bytes (défaut : 32)

Nombre d'octets que tlsmgr(8) lit depuis la source \$tls_random_source lors de la (re)mise à jour du pseudo générateur de nombres pseudo-aléatoire en-mémoire (pseudo random number generator PRNG). La valeur par défaut de 32 octets (256 bits) est suffisante pour générer des clefs symétriques de 128bit. Si vous utilisez EGD ou un fichier dev, un maximum de 255 octets est lu.

tls_random_exchange_name (défaut : \${config_directory}/prng_exch)

Nom fichier d'échange du générateur de nombres pseudo aléatoire (pseudo random number generator PRNG) maintenu par tlsmgr(8). Le fichier est créé s'il n'existe déjà et sa longueur est fixée à 1024 octets.

Tant que ce fichier est modifié par Postfix, il doit probablement est gardé dans le système de fichier /var au lieu du répertoire \$config_directory. Cet emplacement ne devrait pas être dans la cage chroot.

tls_random_prng_update_period (défaut : 60s)

Délai maximal entre deux tentatives de sauvegarde de l'état du générateur de nombres pseudo-aléatoires (pseudo random number generator PRNG) par tlsmgr(8) dans le fichier \$tls_random_exchange_name.

tls_random_reseed_period (défaut : 3600s)

Délai maximal entre deux tentatives de réinitialiser (graine) le générateur en mémoire de nombres pseudo aléatoires (pseudo random number generator PRNG) depuis les sources externes par tlsmgr(8). Le délai entre deux tentatives est calculé en utilisant le PRNG, et est compris entre 0 et le

délicia indique.

tls_random_source (défaut : vide)

Source externe d'entropie pour le gestionnaire tlsmgr(8) du pool de générateurs en mémoire de nombres pseudo-aléatoires (pseudo random number generator PRNG). Assurez-vous d'indiquer une source non bloquante. Si la source n'est pas un fichier régulier, le nom de la source d'entropie doit être préfixé de son type : `egd:/path/to/egd_socket` pour un source disposant une interface socket compatible EGD, ou `dev:/path/to/device` pour un fichier dev.

Note : sur les systèmes OpenBSD, indiquez `/dev/arandom` lorsque `/dev/urandom` produit des erreurs de timeouts.

trace_service_name (défaut : trace)

Nom du service trace. Ce service est implémenté par le démon bounce(8) maintient un enregistrement des livraisons de message et produit un rapport de livraison lorsque la livraison verbuse est demandée avec `"sendmail -v"`.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

transport_maps (défaut : vide)

Tables optionnelles de correspondances entre adresse de destination et (transporteur, destination suivante). Lisez la page de manuel transport(5) pour plus de détails.

Pour des raisons de sécurité, depuis la version 2.3 de Postfix, cette fonctionnalité n'autorise pas les substitutions \$nombre dans les tables d'expressions régulières.

Indiquez zéro ou plus de tables `"type:table"` de correspondances. Si vous utilisez cette fonctionnalité avec des fichiers locaux, lancez `"postmap /etc/postfix/transport"` après un changement.

Exemples :

```
transport_maps = dbm:/etc/postfix/transport
transport_maps = hash:/etc/postfix/transport
```

transport_retry_time (défaut : 60s)

Temps entre deux tentatives du gestionnaire des files d'attente de Postfix pour contacter un transporteur de messages defectueux.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

trigger_timeout (défaut : 10s)

Temps limite pour envoyer un trigger à un démon de Postfix (par exemple, le démon pickup(8) ou qmgr(8)). Cette limite évite aux programmes de rester bloqué lorsque le système de messagerie est fortement chargé.

Unités de temps: s (secondes), m (minutes), h (heures), d (jours), w (semaines). L'unité de temps par défaut est la seconde.

undisclosed_recipients_header (défaut : To: undisclosed-recipients:;)

En-tête de message que le serveur cleanup(8) de Postfix insert lorsqu'un message ne contient pas d'en-tête de message To: ou Cc:.

unknown_address_reject_code (défaut : 450)

Code numérique de réponse du serveur SMTP de Postfix lorsqu'une adresse d'expéditeur ou une adresse de destination est rejetée par les restrictions reject_unknown_sender_domain ou reject_unknown_recipient_domain.

Ne changez pas ceci avant d'avoir bien compris la [RFC 821](#).

unknown_client_reject_code (défaut : 450)

Code numérique de réponse du serveur SMTP de Postfix lorsqu'un client sans nom d'adresse valide <=> est rejeté par la restriction [reject_unknown_client](#). Le serveur SMTP répond toujours avec un code 450 lorsque le mapping échoue pour un problème temporaire.

Ne changez pas ceci avant d'avoir bien compris la [RFC 821](#).

unknown_hostname_reject_code (défaut : 450)

Code numérique de réponse du serveur SMTP de Postfix lorsque le nom de machine indiqué par la commande HELO ou EHLO est rejeté par la restriction [reject_unknown_hostname](#).

Ne changez pas ceci avant d'avoir bien compris la [RFC 821](#).

unknown_local_recipient_reject_code (défaut : 550)

Code numérique de réponse du serveur SMTP de Postfix lorsque l'adresse de destination est locale et que [\\$local_recipient_maps](#) indique une liste de tables de correspondances et que le destinataire n'est pas trouvé. Une adresse de destination est locale lorsque son domaine correspond à [\\$mydestination](#), [\\$proxy_interfaces](#) ou [\\$inet_interfaces](#).

La valeur par défaut est 550 (reject mail) mais peut être mise à 450 (try again later) pour vous permettre de contrôler votre [local_recipient_maps](#).

Exemple :

```
unknown\_local\_recipient\_reject\_code = 450
```

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

unknown_relay_recipient_reject_code (défaut : 550)

Code de réponse numérique du serveur SMTP de Postfix lorsqu'une adresse de destination correspond à [\\$relay_domains](#) et que [relay_recipient_maps](#) indique une liste de tables de correspondances sans que l'adresse de destination n'y soit trouvée.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

unknown_virtual_alias_reject_code (défaut : 550)

Code de réponse numérique du serveur SMTP de Postfix lorsqu'une adresse de destination correspond à [\\$virtual_alias_domains](#) et que [\\$virtual_alias_maps](#) indique une liste de tables de correspondances sans que l'adresse de destination n'y soit trouvée.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

unknown_virtual_mailbox_reject_code (défaut : 550)

Code de réponse numérique du serveur SMTP de Postfix lorsqu'une adresse de destination correspond à [\\$virtual_mailbox_domains](#) et que [\\$virtual_mailbox_maps](#) indique une liste de tables de correspondances sans que l'adresse de destination n'y soit trouvée.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

unverified_recipient_reject_code (défaut : 450)

Code de réponse numérique du serveur SMTP de Postfix lorsqu'une adresse de destination est rejetée par la restriction [reject_unverified_recipient](#).

Vous pouvez indiquer 250 pour accepter tout de même l'adresse.

Ne changez pas ceci avant d'avoir bien compris la [RFC 821](#).

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

unverified_sender_reject_code (défaut : 450)

Code numérique de réponse du serveur SMTP de Postfix lorsqu'une adresse de destination est rejetée par la restriction reject_unverified_sender.

Vous pouvez indiquer 250 pour accepter tout de même l'adresse.

Ne changez pas ceci avant d'avoir bien compris la RFC 821.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

verp_delimiter_filter (défaut : --=+)

Caractères que Postfix accepte comme délimiteur VERP sur la ligne de commande de la commande Postfix sendmail(1).

Cette fonctionnalité est disponible dans les versions 1.1 et supérieures de Postfix.

virtual_alias_domains (défaut : \$virtual_alias_maps)

Liste optionnelle de noms de domaines d'alias virtuels, c'est à dire les domaines pour lesquels toutes les adresses ont des alias pointant sur des adresses de domaines locaux ou extérieurs. Le serveur SMTP valide les adresses de destination avec les tables \$virtual_alias_maps et rejette les destinataires inexistantes. Voyez également la classe virtual alias domain à la page ADDRESS CLASS README

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix. La valeur par défaut est compatible avec Postfix 1.1.

La valeur par défaut est \$virtual_alias_maps ainsi vous pouvez conserver toutes les informations sur les domaines d'alias virtuels en un seul lieu. Si vous avez beaucoup d'utilisateurs, il est préférable de séparer les informations qui changent fréquemment (adresse virtuelle -> adresse correspondante locale ou distante) de celles qui changent moins souvent (liste des noms de domaines virtuels).

Indiquez une liste d'expression adresse/réseau, séparé par des virgules et/ou des espaces. Vous pouvez également indiquer des noms de machines ou des noms de domaines (s'ils sont précédés d'un "." tous les sous-domaines sont pris en compte), des "/nom/de/fichier" ou des expressions "type:table". Un "/nom/de/fichier" de correspondances est remplacé par son contenu ; une table de correspondances "type:table" correspond lorsqu'une entrée de la table correspond (le résultat de la consultation est ignoré). Continuez les lignes longues en commençant la ligne suivante par des espaces.

Lisez également les pages VIRTUAL README et ADDRESS CLASS README.

Exemple :

```
virtual_alias_domains = virtual1.tld virtual2.tld
```

virtual_alias_expansion_limit (défaut : 1000)

Nombre maximum d'adresses qu'une substitution d'alias virtuel peut produire pour chaque destinataire original.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

virtual_alias_maps (défaut : \$virtual_maps)

Tables d'optionnelles faisant correspondre des adresses ou des domaines spécifiés avec des adresses locales ou distantes. Le format de la table et les recherches sont documentées à la page virtual(5).

Documentation de Postfix en français

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix. La valeur par défaut est compatible avec Postfix 1.1.

Si vous utilisez cette fonctionnalité avec des fichiers indexés, lancez "**postmap /etc/postfix/virtual**" après toute modification.

Exemples :

```
virtual_alias_maps = dbm:/etc/postfix/virtual  
virtual_alias_maps = hash:/etc/postfix/virtual
```

virtual_alias_recursion_limit (défaut : 1000)

Profondeur maximale de substitutions des alias virtuels. Couramment la limite récursive est appliquée seulement à la partie gauche du graphe de substitution, ainsi la profondeur de l'arbre peut dans le pire des cas atteindre la somme des limites de récursivité et de substitution. Ceci changera ultérieurement.

Cette fonctionnalité est disponible dans les versions 2.1 et supérieures de Postfix.

virtual_destination_concurrency_limit (défaut : \$default_destination_concurrency_limit)

Nombre maximum de livraisons parallèles vers la même destination via le transporteur de courrier virtuel. Cette limite est utilisée par le gestionnaire des files d'attente. Le nom du transporteur de message correspond au premier champ du fichier master.cf.

virtual_destination_recipient_limit (défaut : \$default_destination_recipient_limit)

Nombre maximum de destinataires par livraison via le transporteur de courrier virtuel. Cette limite est utilisée par le gestionnaire des files d'attente. Le nom du transporteur de message correspond au premier champ du fichier master.cf.

Mettre ce paramètre à 1 change le sens de virtual_destination_concurrency_limit de concurrence par domaine en concurrence par destinataire.

virtual_gid_maps (défaut : vide)

Tables de correspondances entre destinataires et identifiants de groupe pour la livraison vers des boîtes virtuelles.

Dans une table de correspondances, insérez "@domain.tld" sur la partie gauche pour faire correspondre tous les utilisateurs de ce domaine qui n'ont pas d'entrée individuelle "user@domain.tld".

Lorsqu'une adresse de destination a une extension optionnelle d'adresse (user+foo@domain.tld), l'agent de livraison virtual(8) recherche l'adresse complète et lorsque la consultation échoue, il recherche l'adresse sans extension (user@domain.tld).

Note 1 : pour des raisons de sécurité, l'agent de livraison virtual(8) n'autorise pas les substitutions d'expression rationnelle telles \$1 etc.. dans les tables de correspondances d'expression rationnelle, car cela pourrait ouvrir un trou de sécurité.

Note 2 : pour des raisons de sécurité, l'agent de livraison virtual(8) n'autorise pas les consultations de table par le service proxymap(8), car cela pourrait ouvrir un trou de sécurité.

virtual_mailbox_base (défaut : vide)

Préfixe que l'agent de livraison virtual(8) ajoute devant tous les chemins résultats de la recherche dans les tables \$virtual_mailbox_maps. C'est une mesure de sûreté qui évite qu'une correspondance échappant au contrôle de l'administrateur ne pollue le système de fichier avec des boîtes aux lettres. Tant que virtual_mailbox_base peut désigner la racine "/", ce paramètre n'est pas recommandé.

Exemple :

```
virtual_mailbox_base = /var/mail  
virtual_mailbox_domains (défaut : $virtual_mailbox_maps)
```

Liste de domaine livrés via le transporteur de messages \$virtual_transport. Par défaut c'est l'agent de livraison virtual(8) de Postfix. Le serveur SMTP valide les adresses de destination avec \$virtual_mailbox_maps et rejette le courrier des destinataires inconnus. Étudiez également la classe virtual_mailbox_domain à la page ADDRESS CLASS README.

Ce paramètre requiert la même syntaxe que mydestination.

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix. La valeur par défaut est compatible with Postfix 1.1.

```
virtual_mailbox_limit (défaut : 51200000)
```

Taille maximale en octet d'un fichier ou répertoire individuel de boîte-aux lettres, ou zéro (pas de limite).

```
virtual_mailbox_lock (défaut : voir la sortie de "postconf -d")
```

Comment verrouiller une boîte-aux-lettres virtuelle type UNIX avant de tenter la livraison. Pour connaître le liste des méthodes de verrouillage disponibles, lancez la commande "**postconf -l**".

Ce paramètre est ignoré avec les livraisons type **maildir**, car ces livraisons sont sûres même sans verrou applicatif.

Note 1 : La méthode **dotlock** requiert que l'UID ou le GID du destinataire ait un droit d'accès en écriture sur le répertoire parent du fichier boîte-aux-lettres du destinataire.

Note 2: la valeur par défaut de ce paramètre dépend du système.

```
virtual_mailbox_maps (défaut : vide)
```

Tables de correspondances optionnelles contenant toutes les adresses valides des domaines correspondant à virtual_mailbox_domains.

Dans une table de correspondances, indiquez sur la partie gauche "@domain.tld" pour faire correspondre tous les utilisateurs du domaine spécifié qui n'ont pas d'entrée "user@domain.tld" spécifique.

L'agent de livraison virtual(8) utilise cette table pour rechercher le chemin de la boîte-aux-lettres (fichier ou répertoire). Si le résultat se termine par une barre de fraction ("/"), la livraison type maildir est choisie dans les autres cas, le chemin est censé désigner un fichier boîte-aux-lettres. Notez que \$virtual_mailbox_base est préposé systématiquement devant ce chemin.

Lorsqu'une adresse de destination contient une extension optionnelle (user+foo@domain.tld), l'agent de livraison virtual(8) recherche d'abord l'adresse complète et, si la recherche échoue, recherche l'adresse sans extension (user@domain.tld).

Note 1 : pour des raisons de sécurité, l'agent de livraison virtual(8) n'autorise pas les substitutions \$1 etc.. dans les expressions des tables de correspondances d'expressions rationnelles, car cela peut ouvrir un trou de sécurité.

Note 2 : pour des raisons de sécurité, l'agent de livraison virtual(8) n'autorise pas les recherches via le serveur proxymap(8), car cela peut ouvrir un trou de sécurité.

```
virtual_maps (défaut : vide)
```

Tables de correspondances optionnelles avec a) les noms de domaines pour lesquels toutes les adresses sont aliasées à des adresses locales ou distantes et b) les adresses qui sont aliasées à des adresses locales ou d'autres domaines extérieurs. Disponible sur les versions 2.0 et antérieures de Postfix. Sur les versions 2.1 et supérieures de Postfix, il est remplacé par des contrôles distinct : virtual_alias_domains et virtual_alias_maps.

virtual_minimum_uid (défaut : 100)

Valeur d'UID minimum que l'agent de livraison virtual(8) accepte comme résultat de la recherche dans les tables \$virtual_uid_maps. Les valeurs retournées inférieures sont rejetées et le message est retardé.

virtual_transport (défaut : virtual)

Transporteur de messages par défaut pour les domaines qui correspondent à la valeur du paramètre \$virtual_mailbox_domains. Cette information peut être surchargée par la table transport(5).

Indiquez une chaîne sous la forme *transport:nexthop*, où *transport* est le nom du transporteur de messages défini dans le fichier master.cf. La partie *:nexthop* est optionnelle. Pour plus de détails, consultez la page de manuel transport(5).

Cette fonctionnalité est disponible dans les versions 2.0 et supérieures de Postfix.

virtual_uid_maps (défaut : vide)

Tables de correspondances entre destinataire et UID que l'agent de livraison virtual(8) utilise en écrivant dans la boîte-aux-lettres du destinataire.

Dans une table de correspondances, indiquez dans la partie gauche "@domain.tld" pour faire correspondre tous les utilisateurs du domaine spécifié qui n'ont pas d'entrée spécifique "user@domain.tld".

Lorsqu'une adresse de destination contient une extension optionnelle (user+foo@domain.tld), l'agent de livraison virtual(8) recherche d'abord l'adresse complète et, si la recherche échoue, recherche l'adresse sans extension (user@domain.tld).

Note 1 : pour des raisons de sécurité, l'agent de livraison virtual(8) n'autorise pas les substitutions \$1 etc.. dans les expressions des tables de correspondances d'expressions rationnelles, car cela peut ouvrir un trou de sécurité.

Note 2 : pour des raisons de sécurité, l'agent de livraison virtual(8) n'autorise pas les recherches via le serveur proxymap(8), car cela peut ouvrir un trou de sécurité.

Pages de manuel de Postfix

Informations pour les nouveaux utilisateurs de Postfix

Les nouveaux utilisateurs de Postfix devront tout d'abord consulter les documents d'introduction suivantes. Ils contiennent des liens vers les pages plus précises et les pages de manuel type UNIX. Les pages de manuel type UNIX sont destinées aux personnes déjà familières avec Postfix.

- [Présentation de l'architecture de Postfix](#)
- [Configuration de base](#)
- [Résolution des problèmes](#)
- [Présentation de l'inspection du contenu](#)
- [Présentation du contrôle de relais/d'accès](#)
- [Présentation des tables de correspondances](#)

Organisation des pages de manuel de Postfix

Chaque page de manuel de Postfix est numérotée pour être attachée à une section des manuels UNIX : par exemple [mailq\(1\)](#) ou [access\(5\)](#). Malheureusement, il n'y a pas de méthode unique pour organiser les pages de manuel ; chaque famille UNIX semble différente. La documentation de Postfix adopte la convention suivante :

Section	Sujet
1	Commandes
3	Routines des libraries
5	Formats de fichier
8	Démons

Commandes

- [postalias\(1\)](#), creation/mise à jour/interrogation des bases de données d'alias
- [postcat\(1\)](#), examine les fichiers de file d'attente de Postfix
- [postconf\(1\)](#), Utilitaire de configuration Postfix
- [postfix\(1\)](#), Programme de contrôle de Postfix
- [postkick\(1\)](#), trigger Postfix daemon
- [postlock\(1\)](#), verrouillage compatible Postfix
- [postlog\(1\)](#), générateur de journaux compatible Postfix
- [postmap\(1\)](#), gestionnaire des tables de correspondances de Postfix
- [postqueue\(1\)](#), surveillant des files d'attentes de Postfix
- [postsuper\(1\)](#), Postfix housekeeping
- [mailq\(1\)](#), interface compatible Sendmail
- [newaliases\(1\)](#), interface compatible Sendmail
- [sendmail\(1\)](#), interface compatible Sendmail

Configuration de Postfix

- [bounce\(5\)](#), modèles de messages de rejet de Postfix
- [master\(5\)](#), syntaxe du fichier master.cf de Postfix
- [postconf\(5\)](#), syntaxe du fichier main.cf de Postfix

Construction des tables

- [access\(5\)](#), tables de contrôle d'accès SMTP de Postfix
- [aliases\(5\)](#), bases de données d'alias de Postfix
- [canonical\(5\)](#), réécriture des adresses entrantes
- [generic\(5\)](#), réécriture des adresses sortantes
- [header_checks\(5\)](#), [body_checks\(5\)](#), inspection du contenu
- [relocated\(5\)](#), Utilisateurs déplacés
- [transport\(5\)](#), tables de routage de Postfix
- [virtual\(5\)](#), alias virtuels de Postfix

Types de tables

- [cidr_table\(5\)](#), tables d'expressions CIDR
- [ldap_table\(5\)](#), client LDAP de Postfix LDAP
- [mysql_table\(5\)](#), client MYSQL de Postfix
- [nisplus_table\(5\)](#), client NIS+ de Postfix
- [pcre_table\(5\)](#), tables d'expressions PCRE
- [pgsql_table\(5\)](#), client PostgreSQL de Postfix
- [regexp_table\(5\)](#), tables d'expressions POSIX
- [tcp_table\(5\)](#), consultation de tables client–serveur de Postfix

Processus démons

- [anvil\(8\)](#), limite de connexions/taux de Postfix
- [bounce\(8\)](#), [defer\(8\)](#), [trace\(8\)](#), rapporte les statuts de livraison
- [cleanup\(8\)](#), traite et met en file d'attente les messages
- [discard\(8\)](#), agent de livraison retardée de Postfix
- [error\(8\)](#), agent de livraison en erreur de Postfix
- [flush\(8\)](#), service ETRN rapide de Postfix
- [lmtp\(8\)](#), client LMTP de Postfix
- [local\(8\)](#), agent de livraison local de Postfix
- [master\(8\)](#), démon principal de Postfix
- [oqmgr\(8\)](#), ancien gestionnaire des files d'attente de Postfix
- [pickup\(8\)](#), récupère les messages soumis localement
- [pipe\(8\)](#), livre les messages aux commandes non–Postfix
- [proxymap\(8\)](#), serveur mandataire (proxy) pour la consultation des tables
- [qmgr\(8\)](#), gestionnaire des files d'attente de Postfix
- [qmqpd\(8\)](#), serveur QMQP de Postfix
- [scache\(8\)](#), gère le cache des connexions
- [showq\(8\)](#), montre la file d'attente de Postfix
- [smtp\(8\)](#), client SMTP de Postfix
- [smtpd\(8\)](#), serveur SMTP de Postfix

- spawn(8), lance les serveurs non-Postfix
- tlsmgr(8), gère le cache TLS et le générateur aléatoire
- trivial-rewrite(8), réécrit les adresses
- verify(8), vérifie les adresses
- virtual(8), agent de livraison virtuel