Exim4 en Smarthost

Ce tutoriel développe la mise en place d'un serveur de messagerie Exim4 configuré en Smarthost (relaye vers un FAI) avec un antivirus Clamav, Spamassassin, fetchmail et Qpopper comme serveur POP sous Debian Sarge.
Les utilisateurs auront une adresse local (user@alex.fr) qui sera réécrite (user@fai.com) pour sortir vers internet.
Une liste de correspondance d'adresses mails entre les comptes locaux et FAI permettra de garder en interne les mails destinés aux utilisateurs du domaine pour éviter de relayer systématiquement vers le FAI.

Installation des packages

Sur le serveur :

exim4_4.50-4_all.deb
exim4-base_4.50-4_i386.deb
exim4-config_4.50-4_all.deb
exim4-daemon-heavy_4.50-4_i386.deb (intégre exiscan-acl par défaut)

clamav_0.83-5_i386.deb
clamav-base_0.83-5_all.deb
clamav-daemon_0.83-5_i386.deb
clamav-freshclam_0.83-5_i386.deb
clamav-testfiles_0.83-5_all.deb
libclamav1_0.83-5_i386.deb
arj_3.10.20-1_i386.deb
unzoo_4.4-2_i386.deb
unrar_0.0.1-1_i386.deb

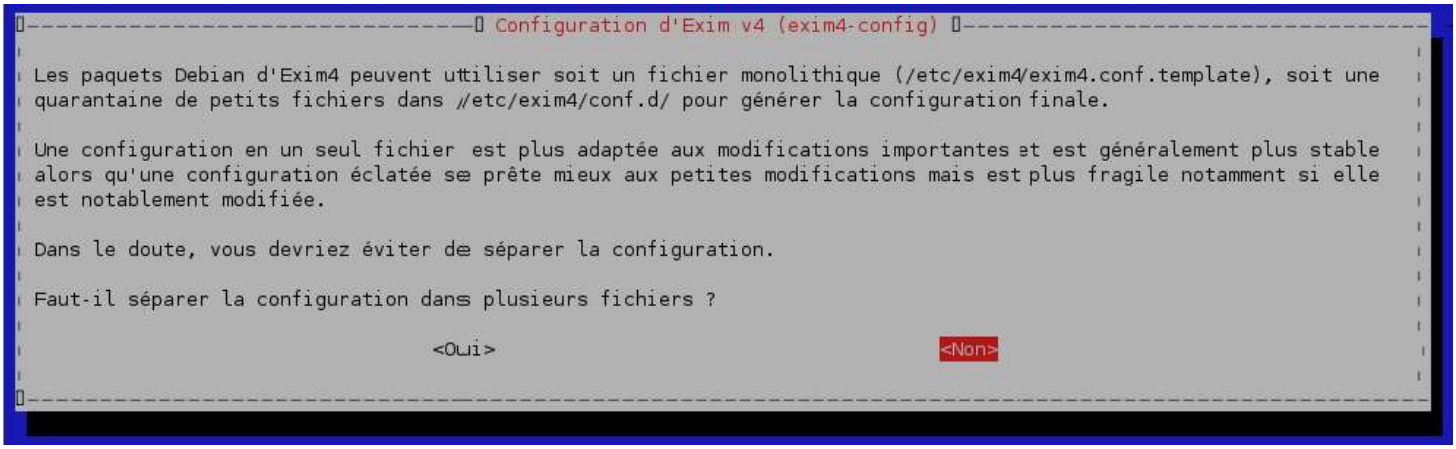
fetchmail_6.2.5-10_i386.deb

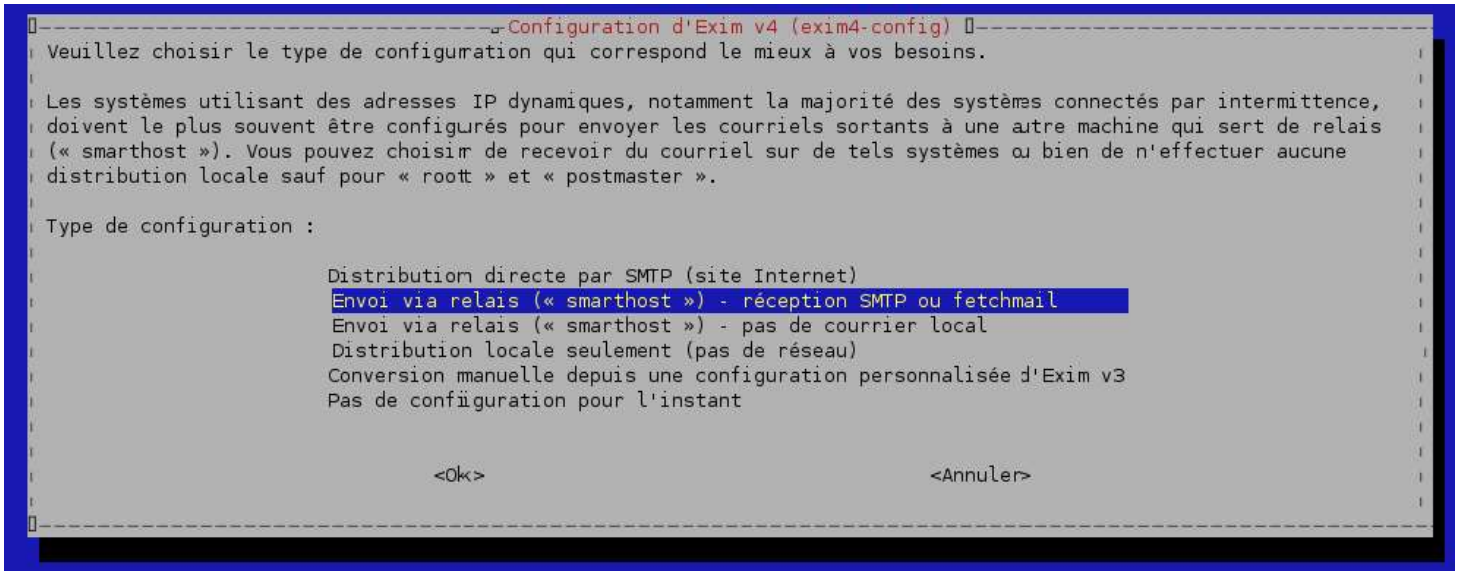
spamassassin_3.0.2-1_all.deb
spamc_3.0.2-1_i386.deb

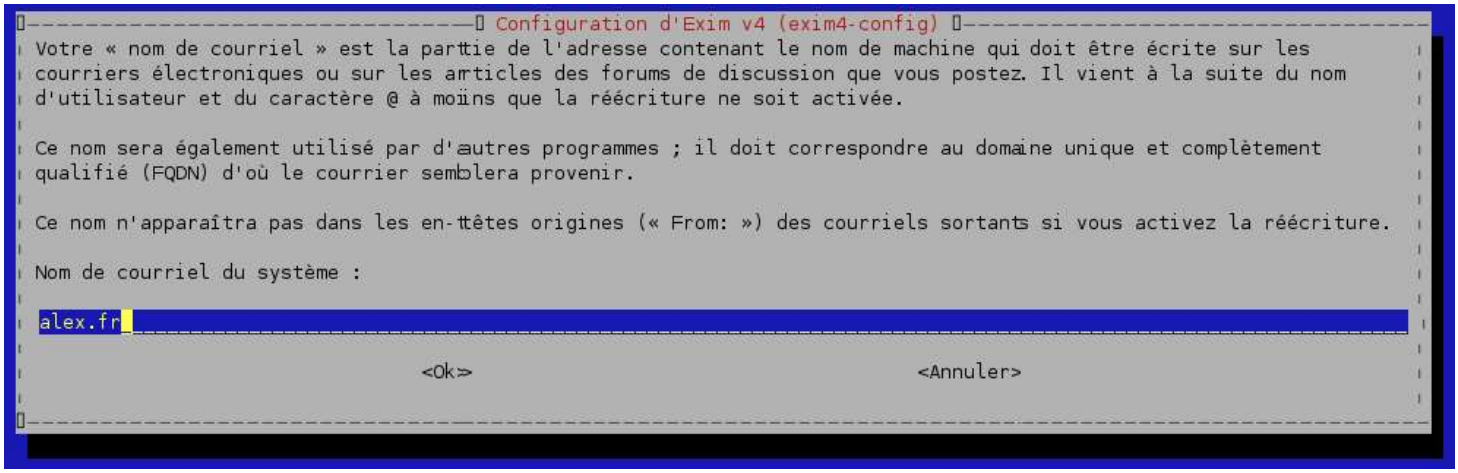
qpopper_4.0.5-4_i386.deb

Configuration de Exim4

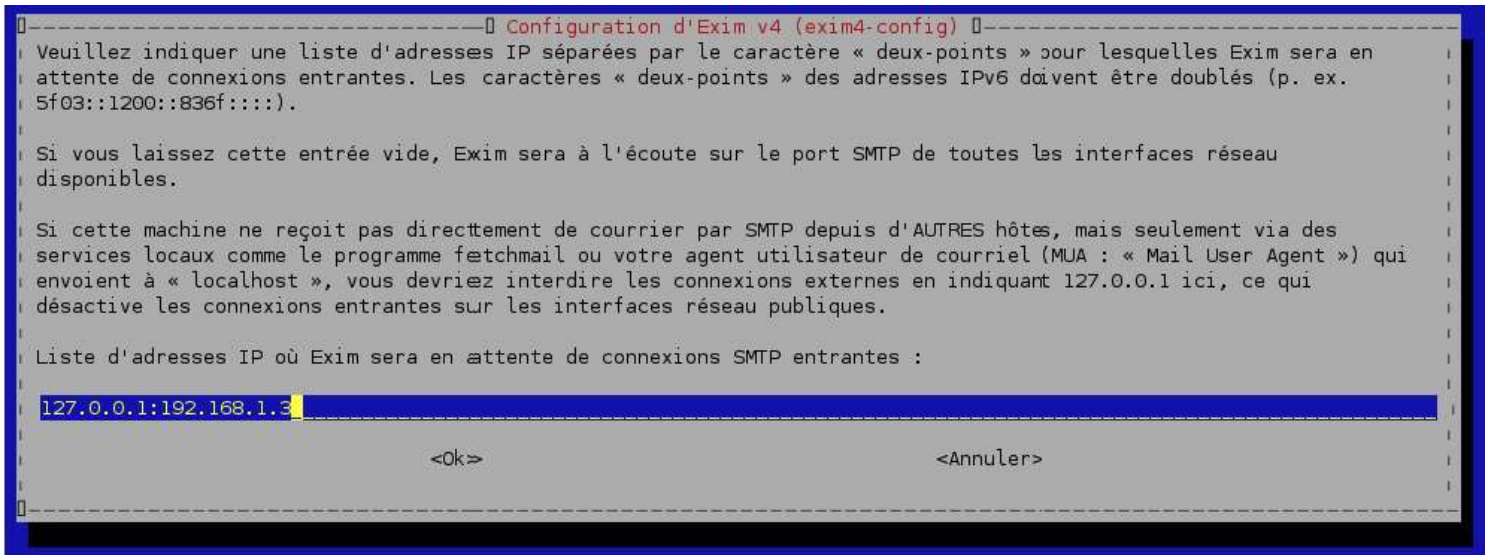
Lors de l'installation du package ou de sa reconfiguration (dpkg-reconfigure exim4-config) saisissez les informations suivantes :

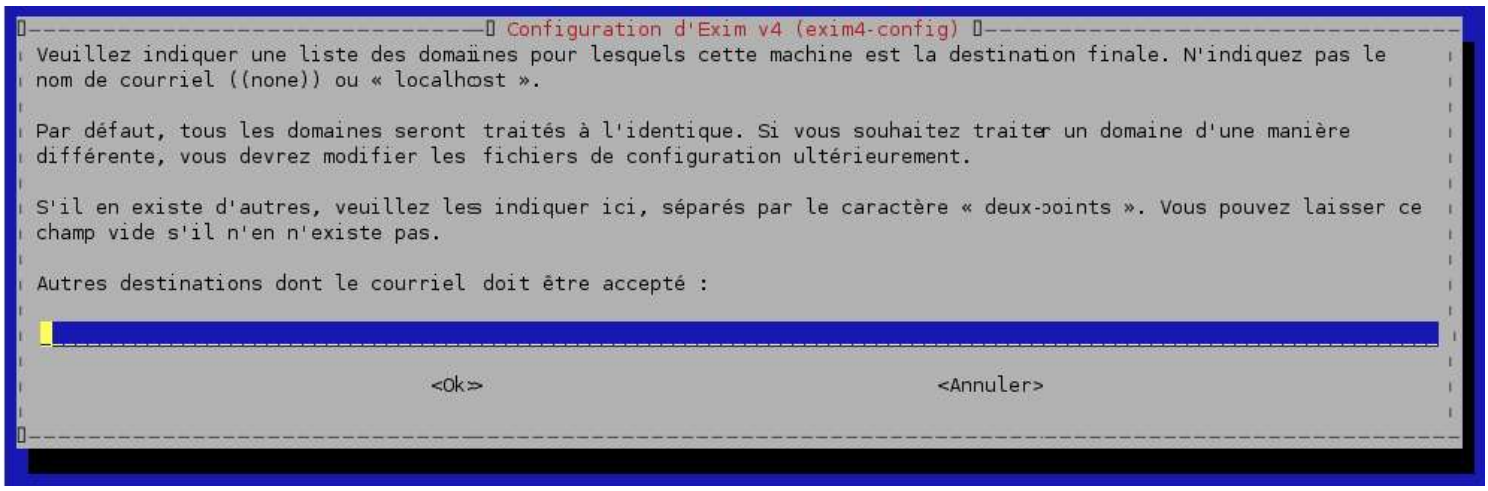


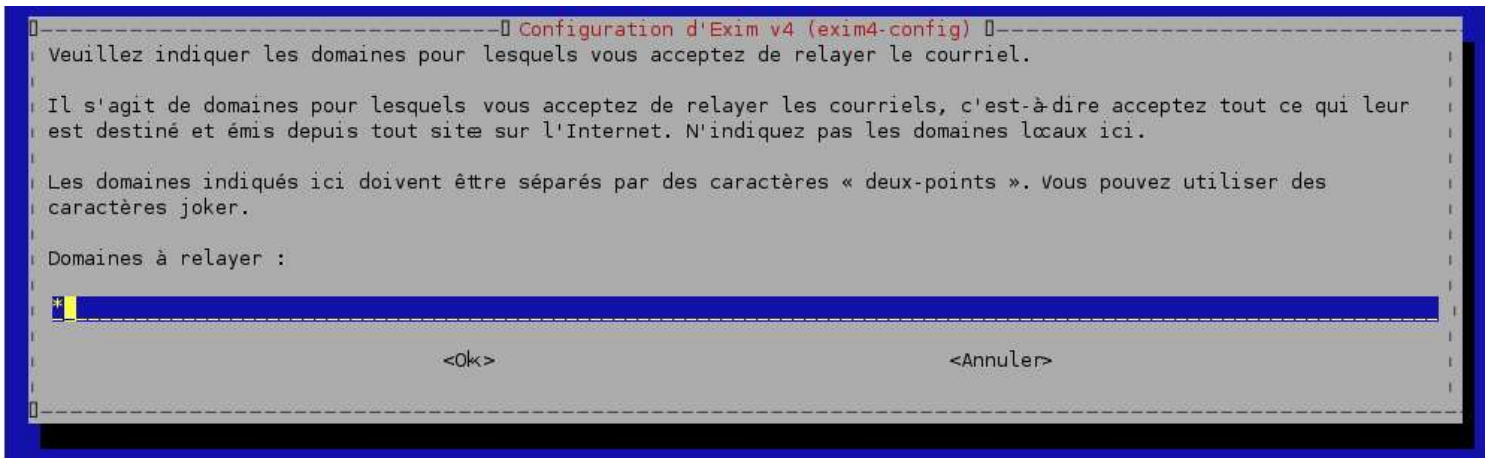


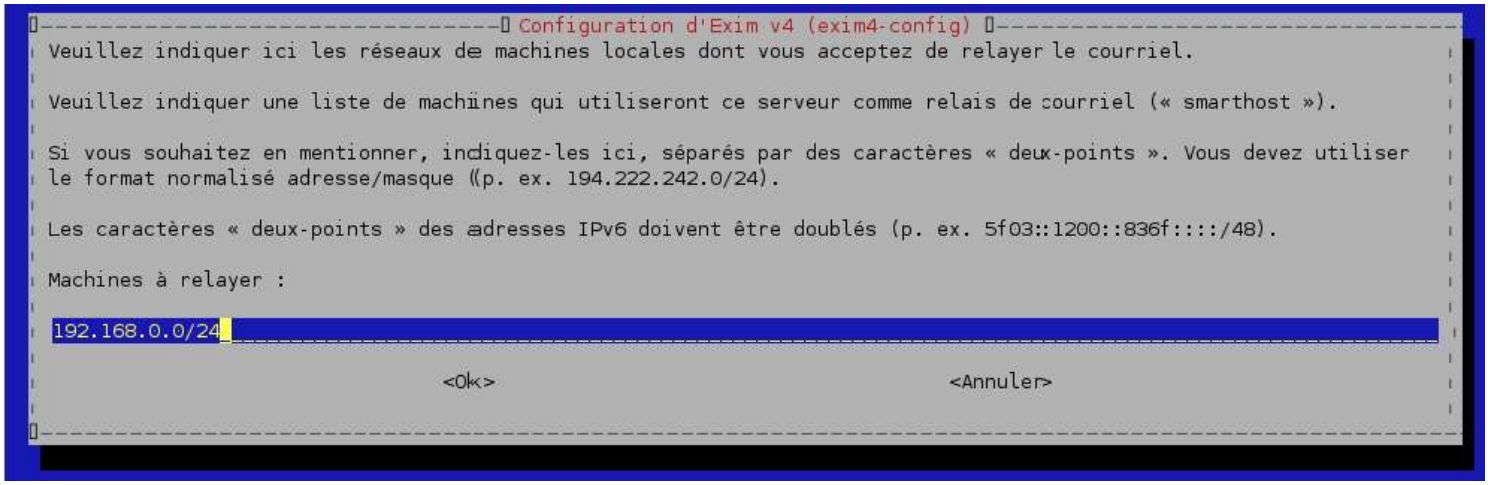


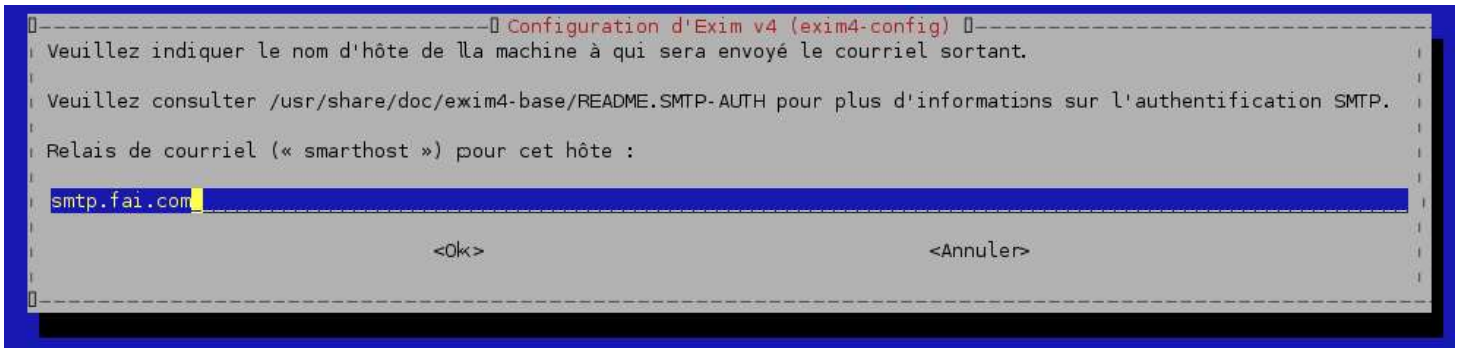
Indiquez la boucle locale et l'IP de votre serveur mail.

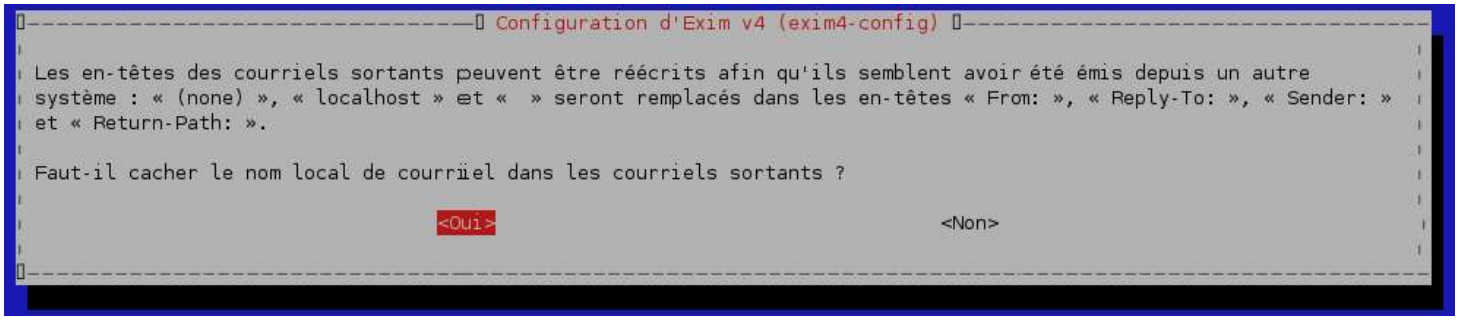


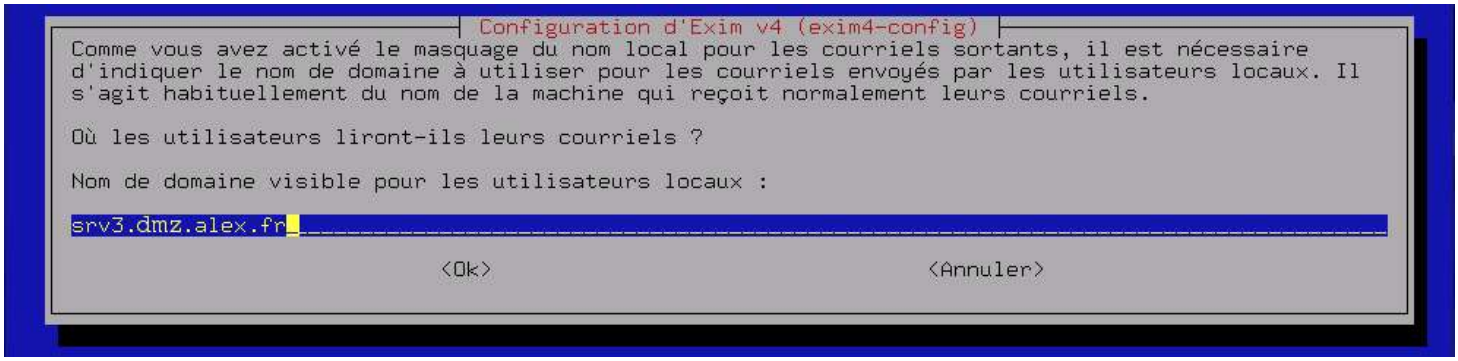


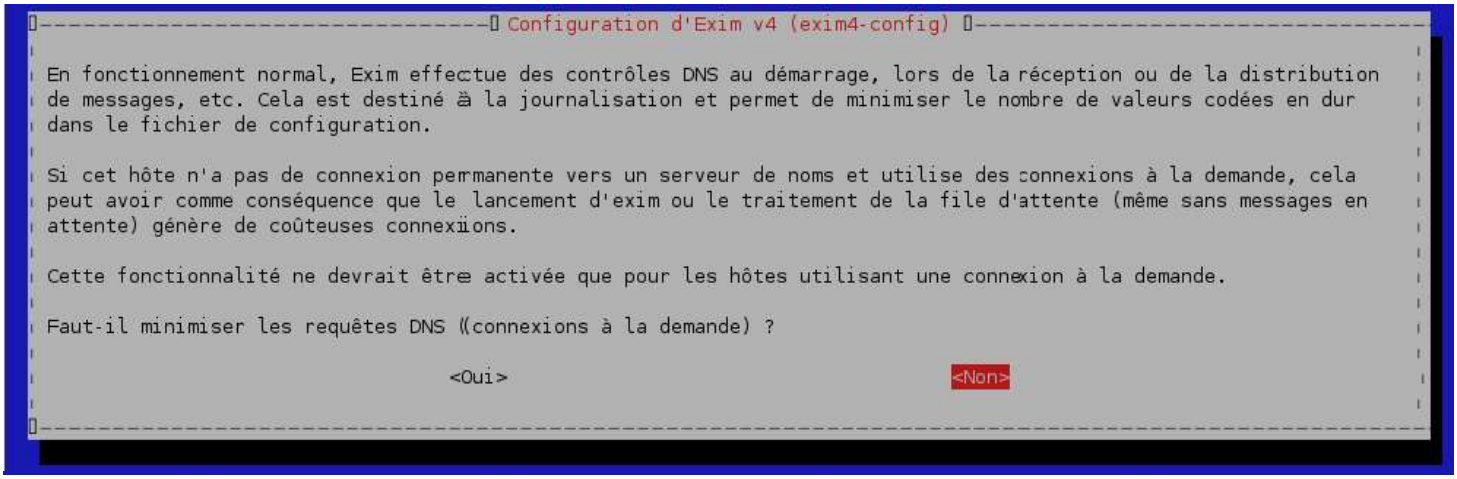



Configuration d'Exim v4 (exim4-config)
Veuillez indiquer ici les réseaux de machines locales dont vous acceptez de relayer le courriel.
Veuillez indiquer une liste de machines qui utiliseront ce serveur comme relais de courriel (« smarthost »).
Si vous souhaitez en mentionner, indiquez-les ici, séparés par des caractères « deux-points ». Vous devez utiliser le format normalisé adresse/masque ((p. ex. 194.222.242.0/24).
Les caractères « deux-points » des adresses IPv6 doivent être doublés (p. ex. 5f03::1200::836f::::/48).
Machines à relayer :
192.168.0.0/24
<Ok>
<Annuler>


Configuration d'Exim v4 (exim4-config)
Veuillez indiquer le nom d'hôte de la machine à qui sera envoyé le courriel sortant.
Veuillez consulter /usr/share/doc/exim4-base/README.SMTP-AUTH pour plus d'informations sur l'authentification SMTP.
Relais de courriel (« smarthost ») pour cet hôte :
smtp.fai.com
<Ok>
<Annuler>


Configuration d'Exim v4 (exim4-config)
Les en-têtes des courriels sortants peuvent être réécrits afin qu'ils semblent avoir été émis depuis un autre système : « (none) », « localhost » et « » seront remplacés dans les en-têtes « From: », « Reply-To: », « Sender: » et « Return-Path: ».
Faut-il cacher le nom local de courriel dans les courriels sortants ?
<Oui>
<Non>


Configuration d'Exim v4 (exim4-config)
Comme vous avez activé le masquage du nom local pour les courriels sortants, il est nécessaire d'indiquer le nom de domaine à utiliser pour les courriels envoyés par les utilisateurs locaux. Il s'agit habituellement du nom de la machine qui reçoit normalement leurs courriels.
Où les utilisateurs liront-ils leurs courriels ?
Nom de domaine visible pour les utilisateurs locaux :
srv3.dmz.alex.fr
<Ok>
<Annuler>



L'assistant vous créez le fichier **/etc/exim4/update-exim4.conf.conf** qui contient les variables utilisées dans **/etc/exim4/exim4.conf.template,** vérifiez son contenu :

```
# /etc/exim4/update-exim4.conf.conf
#
# Edit this file and /etc/mailname by hand and execute update-exim4.conf
# yourself or use 'dpkg-reconfigure exim4-config'

dc_eximconfig_configtype='smarthost'
dc_other_hostnames=''
dc_local_interfaces='127.0.0.1:192.168.1.3'
dc_readhost='srv3.dmz.alex.fr'
dc_relay_domains='*'
dc_minimaldns='false'
dc_relay_nets='192.168.0.0/24'
dc_smarthost='smtp.fai.com'
CFILEMODE='644'
dc_use_split_config='false'
dc_hide_mailname='true'
```

Modifiez le fichier de configuration d'Exim4 **/etc/exim4/exim4.conf.template**

```
##################################################
### main/01_exim4-config_listmacrosdefs
##################################################

##################################################
# Runtime configuration file for Exim #
##################################################

##################################################
# MAIN CONFIGURATION SETTINGS #
##################################################

# Just for reference and scripts, on debian, the main binary is
# installed as exim4
exim_path = /usr/sbin/exim4

# Macro defining the main configuration directory, we use no abolute
# paths.
CONFDIR = /etc/exim4

# Pour éviter les timeout de Exim en cas de surcharge.
deliver_queue_load_max = 1.0
smtp_load_reserve = 5.0

# Communication avec l'antivirus clamav par son socket.
av_scanner = clamd:/var/run/clamav/clamd.ctl

# Communication avec spamassassin par son socket.
# Voir les options à passer dans /etc/default/spamassassin pour l'activer.
spamd_address = /var/run/spamd.socket

# Define a macro DC_minimaldns if dc_minimaldns=true, to use in
# .ifdef-statements otherwise this expands to an empty line
DEBCONFminimaldnsDEBCONF

# The next three settings create two lists of domains and one list of
hosts.
# These lists are referred to later in this configuration using the syntax
# +local_domains, +relay_to_domains, and +relay_from_hosts, respectively.
They
# are all colon-separated lists:

# '@' refers to 'the name of the local host'

### EXPANSION-begins ######################
domainlist local_domains = DEBCONFlocal_domainsDEBCONF
domainlist relay_to_domains = DEBCONFrelay_domainsDEBCONF
hostlist relay_from_hosts = 127.0.0.1 : ::::1 : DEBCONFrelay_netsDEBCONF

# Specify the domain you want to be added to all unqualified addresses
# here. An unqualified address is one that does not contain an "@"
character
# followed by a domain. For example, "caesar@rome.example" is a fully
qualified
# address, but the string "caesar" (i.e. just a login name) is an
```

```
unqualified
# email address. Unqualified addresses are accepted only from local
callers by
# default. See the recipient_unqualified_hosts option if you want to
permit
# unqualified addresses from remote sources. If this option is not set,
the
# primary_hostname value is used for qualification.
qualify_domain = DEBCONFvisiblenameDEBCONF

# only used for satellite-system
.ifndef DCreadhost
DCreadhost = DEBCONFreadhostDEBCONF
.endif

#for satellite and smarthost-systems
.ifndef DCsmarthost
DCsmarthost = DEBCONFsmarthostDEBCONF
.endif

# listen on all all interfaces?
DEBCONFlistenonpublicDEBCONF
### EXPANSION-ends ######################

# The default delivery method. See CONFDIR/conf.d/transports/ for other
# possibilities
#LOCAL_DELIVERY=DEBCONFlocaldeliveryDEBCONF
LOCAL_DELIVERY=mail_spool

# The gecos field in /etc/passwd holds not only the name. see passwd(5).
gecos_pattern = ^([^,:]*)
gecos_name = $1

# define a macro DCconfig_smarthost, DCconfig_satellite, etc. we need this
# for .ifdef ... .endif
DCconfig_DEBCONFconfigtypeDEBCONF = 1
######################################################
### end main/01_exim4-config_listmacrosdefs
######################################################
######################################################
### main/02_exim4-config_options
######################################################

# This option defines the access control list that is run when an
# SMTP RCPT command is received.
#
acl_smtp_rcpt = acl_check_rcpt

# This option defines the access control list that is run when an
# SMTP DATA command is received.
#
acl_smtp_data = acl_check_data

# Define a message size limit. You can either change it here, or set the
# MESSAGE_SIZE_LIMIT macro. The default (used when MESSAGE_SIZE_LIMIT
# is unset and/or message_size_limit is unset) is 50 MB
.ifdef MESSAGE_SIZE_LIMIT
message_size_limit = 10M
```

```
.endif

# If you want unqualified recipient addresses to be qualified with a
different
# domain to unqualified sender addresses, specify the recipient domain
here.
# If this option is not set, the qualify_domain value is used.
#
# qualify_recipient =

# The following line must be uncommented if you want Exim to recognize
# addresses of the form "user@[10.11.12.13]" that is, with a "domain
literal"
# (an IP address) instead of a named domain. The RFCs still require this
form,
# but it makes little sense to permit mail to be sent to specific hosts by
# their IP address in the modern Internet. This ancient format has been
used
# by those seeking to abuse hosts by using them for unwanted relaying. If
you
# really do want to support domain literals, uncomment the following line,
and
# see also the "domain_literal" router.
#
# allow_domain_literals

.ifndef DC_minimaldns
# The setting below causes Exim to do a reverse DNS lookup on all incoming
# IP calls, in order to get the true host name. If you feel this is too
# expensive, you can specify the networks for which a lookup is done, or
# remove the setting entirely.
#
host_lookup = *
.endif

# For minimaldns try to guess the primary_hostname only once at startup,
when
# running update-exim4.conf
DEBCONF_hardcode_primary_hostname_DEBCONF

# The settings below, which are actually the same as the defaults in the
# code, cause Exim to make RFC 1413 (ident) callbacks for all incoming
SMTP
# calls. You can limit the hosts to which these calls are made, and/or
change
# the timeout that is used. If you set the timeout to zero, all RFC 1413
calls
# are disabled. RFC 1413 calls are cheap and can provide useful
information
# for tracing problem messages, but some hosts and firewalls have problems
# with them. This can result in a timeout instead of an immediate refused
# connection, leading to delays on starting up an SMTP session.
#
rfc1413_hosts = *
rfc1413_query_timeout = 30s

# By default, Exim expects all envelope addresses to be fully qualified,
that
```

```
# is, they must contain both a local part and a domain. If you want to
accept
# unqualified addresses (just a local part) from certain hosts, you can
specify
# these hosts by setting one or both of
#
# sender_unqualified_hosts =
# recipient_unqualified_hosts =
#
# to control sender and recipient addresses, respectively. When this is
done,
# unqualified addresses are qualified using the settings of qualify_domain
# and/or qualify_recipient (see above).

# If you want Exim to support the "percent hack" for certain domains,
# uncomment the following line and provide a list of domains. The "percent
# hack" is the feature by which mail addressed to x%y@z (where z is one of
# the domains listed) is locally rerouted to x@y and sent on. If z is not
one
# of the "percent hack" domains, x%y is treated as an ordinary local part.
This
# hack is rarely needed nowadays; you should not enable it unless you are
sure
# that you really need it.
#
# percent_hack_domains =

# When Exim can neither deliver a message nor return it to sender, it
"freezes"
# the delivery error message (aka "bounce message"). There are also other
# circumstances in which messages get frozen. They will stay on the queue
for
# ever unless one of the following options is set.

# This option unfreezes frozen bounce messages after two days, tries
# once more to deliver them, and ignores any delivery failures.
#
ignore_bounce_errors_after = 2d

# This option cancels (removes) frozen messages that are older than a
week.
#
timeout_frozen_after = 7d
freeze_tell = postmaster

# Only for interacting with other packages, to make it possible to use
# -DSPOOLDIR to override it on the command line
.ifndef SPOOLDIR
SPOOLDIR = /var/spool/exim4
.endif
spool_directory = SPOOLDIR

# uucp should be able to set envelope-from to arbitrary values
trusted_users = uucp

# uncomment this to get the Debian version in the SMTP dialog
# smtp_banner = "${primary_hostname} ESMTP Exim ${version_number} (Debian
package DEBCONFpackageversionDEBCONF) ${tod_full}"
```

```
######################################################
### end main/02_exim4-config_options
######################################################
######################################################
### main/03_exim4-config_tlsoptions
######################################################
# Example for TLS/SSL configuration.

# See /usr/share/doc/exim4-base/README.TLS* for explanations.
# Defines that you want to log what cipher your exim and the peer's mailer
# uses to encrypt the transaction. It also defines you want to log the
'DN'
# (Distinguished Name) of the certificate of the peer.
#
# log_selector = +tls_cipher +tls_peerdn
# Defines what hosts to 'advertise' STARTTLS functionality to. Setting
this
# to * will advertise to all hosts that connect with EHLO, and this is a
# good default
#
# tls_advertise_hosts = *
# Defines where your SSL-certificate and SSL-Private Key are located.
# This requires a full path. The files pointed to must be kept 'secret'
# and should be owned my root.Debian-exim mode 640 (-rw-r-----). Usually
the
# exim-gencert script takes care of these prerequisites.
#
# tls_certificate = CONFDIR/exim.crt
# tls_privatekey = CONFDIR/exim.key
# A file which contains the certificates of the trusted CAs (Certification
# Authorities) against which host certificates can be checked (through the
# `tls_verify_hosts' and `tls_try_verify_hosts' lists below).
# /etc/ssl/certs/ca-certificates.crt is generated by
# the "ca-certificates" package's update-ca-certificates(8) command.
#
#tls_verify_certificates = /etc/ssl/certs/ca-certificates.crt
# A list of hosts which are constrained by `tls_verify_certificates'. A
host
# that matches `tls_verify_host' must present a certificate that's
# verifyable through `tls_verify_certificates' in order to be accepted as
an
# SMTP client. If it does not, the connection is aborted.
#
#tls_verify_hosts =
# A weaker form of checking: if a client matches `tls_try_verify_hosts'
(but
# not `tls_verify_hosts'), request a certificate and check it against
# `tls_verify_certificates' but do not abort the connection if there is no
# certificate or if the certificate presented does not match. (This
# condition can be tested for in ACLs through `verify = certificate')
#
#tls_try_verify_hosts = *
######################################################
### end main/03_exim4-config_tlsoptions
######################################################
######################################################
### acl/00_exim4-config_header
```

```
#################################################

#################################################
# ACL CONFIGURATION #
# Specifies access control lists for incoming SMTP mail
#################################################
begin acl
#################################################
### end acl/00_exim4-config_header
#################################################
#################################################
### acl/20_exim4-config_whitelist_local_deny
#################################################
# This access control list is used to determine whitelisted senders and
# hosts. It checks for CONFDIR/local_host_whitelist and
# CONFDIR/local_sender_whitelist.
#
# It is meant to be used from some other acl entry.
#
# For example,
# deny message = local blacklist example
# !acl = acl_whitelist
# dnslist = some.dns.list.example
# will allow messages with envelope sender listed in
local_sender_whitelist
# or messages coming in from hosts listed in local_host_whitelist to be
# accepted even if the delivering host is listed in the dns list.
#
# Whitelisting can also be configured by including negative items in the
# black list. See /usr/share/doc/exim4-config/default_acl for details.
#
# If the files do not exist, the white list never matches, which is
# the desired behaviour.

acl_whitelist_local_deny:
 accept hosts = ${if exists{CONFDIR/local_host_whitelist}\
  {CONFDIR/local_host_whitelist}\
  {}}
 accept senders = ${if exists{CONFDIR/local_sender_whitelist}\
  {CONFDIR/local_sender_whitelist}\
  {}}

#################################################
### end acl/20_exim4-config_whitelist_local_deny
#################################################
#################################################
### acl/30_exim4-config_check_rcpt
#################################################
# This access control list is used for every RCPT command in an incoming
# SMTP message. The tests are run in order until the address is either
# accepted or denied.
#
acl_check_rcpt:

# Accept if the source is local SMTP (i.e. not over TCP/IP). We do this by
# testing for an empty sending host field.
accept hosts = :
```

```
# The following section of the ACL is concerned with local parts that
contain
# @ or % or ! or / or | or dots in unusual places.
#
# The characters other than dots are rarely found in genuine local parts,
but
# are often tried by people looking to circumvent relaying restrictions.
# Therefore, although they are valid in local parts, these rules lock them
# out, as a precaution.
#
# Empty components (two dots in a row) are not valid in RFC 2822, but Exim
# allows them because they have been encountered. (Consider local parts
# constructed as "firstinitial.secondinitial.familyname" when applied to
# someone like me, who has no second initial.) However, a local part
starting
# with a dot or containing /../ can cause trouble if it is used as part of
a
# file name (e.g. for a mailing list). This is also true for local parts
that
# contain slashes. A pipe symbol can also be troublesome if the local part
is
# incorporated unthinkingly into a shell command line.
#
# Two different rules are used. The first one is stricter, and is applied
to
# messages that are addressed to one of the local domains handled by this
# host. It blocks local parts that begin with a dot or contain @ % ! / or
|.
# If you have local accounts that include these characters, you will have
to
# modify this rule.
deny domains = +local_domains
local_parts = ^[.] : ^.*[@%!/|]
message = restricted characters in address

# The second rule applies to all other domains, and is less strict. This
# allows your own users to send outgoing messages to sites that use
slashes
# and vertical bars in their local parts. It blocks local parts that begin
# with a dot, slash, or vertical bar, but allows these characters within
the
# local part. However, the sequence /../ is barred. The use of @ % and !
is
# blocked, as before. The motivation here is to prevent your users (or
# your users' viruses) from mounting certain kinds of attack on remote
sites.
deny domains = !+local_domains
local_parts = ^[./|] : ^.*[@%!] : ^.*/\\.\\./
message = restricted characters in address

# Accept mail to postmaster in any local domain, regardless of the source,
# and without verifying the sender.
#
accept local_parts = postmaster
domains = +local_domains

# Deny unless the sender address can be verified.
#
```

```
# This is disabled by default so that DNSless systems don't break. If
# your system can do DNS lookups without delay or cost, you might want
# to enable the following line.
# deny message = Sender verification failed
# !acl = acl_whitelist_local_deny
# !verify = sender
# Warn if the sender host does not have valid reverse DNS.
#
# This is disabled by default so that DNSless systems don't break. If
# your system can do DNS lookups without delay or cost, you might want
# to enable the following lines.
# warn message = X-Broken-Reverse-DNS: no host name found for IP address
$sender_host_address
# !verify = reverse_host_lookup
# deny bad senders (envelope sender)
# CONFDIR/local_sender_blacklist holds a list of envelope senders that
# should have their access denied to the local host. Incoming messages
# with one of these senders are rejected at RCPT time.
#
# The explicit white lists are honored as well as negative items in
# the black list. See /usr/share/doc/exim4-config/default_acl for details.
deny message = sender envelope address $sender_address is locally
blacklisted here. If you think this is wrong, get in touch with postmaster
!acl = acl_whitelist_local_deny
senders = ${if exists{CONFDIR/local_sender_blacklist}\
  {CONFDIR/local_sender_blacklist}\
  {}}

# deny bad sites (IP address)
# CONFDIR/local_host_blacklist holds a list of host names, IP addresses
# and networks (CIDR notation) that should have their access denied to
# The local host. Messages coming in from a listed host will have all
# RCPT statements rejected.
#
# The explicit white lists are honored as well as negative items in
# the black list. See /usr/share/doc/exim4-config/default_acl for details.
deny message = sender IP address $sender_host_address is locally
blacklisted here. If you think this is wrong, get in touch with postmaster
!acl = acl_whitelist_local_deny
hosts = ${if exists{CONFDIR/local_host_blacklist}\
  {CONFDIR/local_host_blacklist}\
  {}}

#########################################################
# There are no checks on DNS "black" lists because the domains that
contain
# these lists are changing all the time. You can find examples of
# how to use dnslists in /usr/share/doc/exim4-config/examples/acl
#########################################################
# Accept if the address is in a local domain, but only if the recipient
can
# be verified. Otherwise deny. The "endpass" line is the border between
# passing on to the next ACL statement (if tests above it fail) or denying
# access (if tests below it fail).
#
accept domains = +local_domains
endpass
message = unknown user
```

```
verify = recipient

# Accept if the address is in a domain for which we are relaying, but
again,
# only if the recipient can be verified.
#
accept domains = +relay_to_domains
endpass
message = unrouteable address
verify = recipient

# If control reaches this point, the domain is neither in +local_domains
# nor in +relay_to_domains.
# Accept if the message comes from one of the hosts for which we are an
# outgoing relay. Recipient verification is omitted here, because in many
# cases the clients are dumb MUAs that don't cope well with SMTP error
# responses. If you are actually relaying out from MTAs, you should
probably
# add recipient verification here.
#
accept hosts = +relay_from_hosts

# Accept if the message arrived over an authenticated connection, from
# any host. Again, these messages are usually from MUAs, so recipient
# verification is omitted.
#
accept authenticated = *

# Reaching the end of the ACL causes a "deny", but we might as well give
# an explicit message.
#
deny message = relay not permitted

#########################################################
### end acl/30_exim4-config_check_rcpt
#########################################################
#########################################################
### acl/40_exim4-config_check_data
#########################################################
# 40_exim4-config_check_data

acl_check_data:
# On redirige les mails douteux sur un compte poubelle appelé:
badbox@alex.fr
# Cette boite sert dans un premier temps à tester notre
# config et une fois fonctionnelle il suffira de faire un lien symbolique
# de /var/mail/badbox vers /dev/null. (créez l'utilisateur badbox)
#
# Ces extensions (com:vbs:bat:cmd:pif:scr:exe) de fichiers joints au mails
sont redirigées.
warn message = X-Redirect-To: badbox@alex.fr
  demime = com:vbs:bat:cmd:pif:scr:exe

# On tag le Subject du mail avec le nom du virus détecté.
warn message = Subject: *VIRUS* [$malware_name] $h_Subject
malware = *

# On tag l'entête du mail vérolé avec le nom du virus détecté.
```

```
warn message = Nom du virus detecte ($malware_name)
  malware = *

# On redirige les mails contenant des types mine inconnus et ceux
contenant des virus.
warn message = X-Redirect-To: badbox@alex.fr
  demime = *
  malware = *

# On tag l'entête du mail spammé avec notre nom de domaine et le score de
Spamassassin.
warn message = X-Spam-Score: Alex.fr $spam_score ($spam_bar)
  spam=nobody:true

# On tag le Subject du mail avec *SPAM* pour bien l'identifier.
warn message = Subject: *SPAM* $h_Subject
  spam=nobody

# On redirige les mails ayant un score spam supérieur à 8 (à multiplier
par 10)
warn message = X-Redirect-To: badbox@alex.fr
  spam=nobody:true
  condition = ${if >{$spam_score_int}{80}{1}{0}}

# Add Message-ID if missing
warn condition = ${if !def:h_Message-ID: {1}}
hosts = +relay_from_hosts
message = Message-ID: <E$message_id@$primary_hostname>

# Deny unless the address list headers are syntactically correct.
#
# This is disabled by default because it might reject legitimate mail.
# If you want your system to insist on syntactically valid address
# headers, you might want to enable the following lines.
# deny message = Message headers fail syntax check
# !acl = acl_whitelist_local_deny
# !verify = header_syntax
# require that there is a verifiable sender address in at least
# one of the "Sender:", "Reply-To:", or "From:" header lines.
# deny message = No verifiable sender address in message headers
# !acl = acl_whitelist_local_deny
# !verify = header_sender
# accept otherwise
accept
#############################################################
### end acl/40_exim4-config_check_data
#############################################################
#############################################################
### router/00_exim4-config_header
#############################################################

#############################################################################
# ROUTERS CONFIGURATION #
# Specifies how addresses are handled #
#############################################################################
# THE ORDER IN WHICH THE ROUTERS ARE DEFINED IS IMPORTANT! #
# An address is passed to each router in turn until it is accepted. #
#############################################################################
```

```
begin routers

#####################################################
### end router/00_exim4-config_header
#####################################################
#####################################################
### router/100_exim4-config_domain_literal
#####################################################

# This router routes to remote hosts over SMTP by explicit IP address,
# when an email address is given in "domain literal" form, for example,
# <user@[192.168.35.64]>. The RFCs require this facility. However, it is
# little-known these days, and has been exploited by evil people seeking
# to abuse SMTP relays. Consequently it is commented out in the default
# configuration. If you uncomment this router, you also need to uncomment
# allow_domain_literals above, so that Exim can recognize the syntax of
# domain literal addresses.

# domain_literal:
# debug_print = "R: domain_literal for $local_part@$domain"
# driver = ipliteral
# domains = ! +local_domains
# transport = remote_smtp

#####################################################
### end router/100_exim4-config_domain_literal
#####################################################
#####################################################
### router/150_exim4-config_hubbed_hosts
#####################################################

# route specific domains manually.
#
# The most common application of this router is to handle relaying to
nonlocal
# domains that the local host is primary MX for. That means that local
# information needs to be present for a domain to be handled correctly.
#
# That information is put into the optional file /etc/exim4/hubbed_hosts
# which contains key-value pairs of domain pattern and route data.
#
# foo.example: internal.mail.example.com
# bar.example: 192.168.183.3
#
# will cause mail for foo.example to be sent to the host
# internal.mail.example (IP address derived from A record only), and
# mail to bar.example to be sent to 192.168.183.3.
#
# If the file /etc/exim4/hubbed_hosts does not exist, this router is a
# no-op.

hubbed_hosts:
debug_print = "R: hubbed_hosts for $domain"
driver = manualroute
domains = "${if exists{CONFDIR/hubbed_hosts}\
{partial-lsearch;CONFDIR/hubbed_hosts}\
fail}"
route_data = ${lookup{$domain}partial-lsearch{CONFDIR/hubbed_hosts}}
```

```
transport = remote_smtp
#####################################################
### end router/150_exim4-config_hubbed_hosts
#####################################################
#####################################################
### router/200_exim4-config_primary
#####################################################

# This file holds the primary router, responsible for nonlocal mails

.ifdef DCconfig_internet
# configtype=internet
#
# deliver mail to the recipient if recipient domain is a domain we
# relay for. We do not ignore any target hosts here since delivering to
# a site local or even a link local address might be wanted here, and if
# such an address has found its way into the MX record of such a domain,
# the local admin is probably in a place where that broken MX record
# could be fixed.

dnslookup_relay_to_domains:
debug_print = "R: dnslookup_relay_to_domains for $local_part@$domain"
driver = dnslookup
domains = ! +local_domains : +relay_to_domains
transport = remote_smtp
same_domain_copy_routing = yes
no_more

# deliver mail directly to the recipient. This router is only reached
# for domains that we do not relay for. Since we most probably can't
# have broken MX records pointing to site local or link local IP
# addresses fixed, we ignore target hosts pointing to these addresses.

dnslookup:
debug_print = "R: dnslookup for $local_part@$domain"
driver = dnslookup
domains = ! +local_domains
transport = remote_smtp
same_domain_copy_routing = yes

# ignore private rfc1918 and APIPA addresses
ignore_target_hosts = 0.0.0.0 : 127.0.0.0/8 : 192.168.0.0/16 :\
172.16.0.0/12 : 10.0.0.0/8 : 169.254.0.0/16
no_more

.endif

.ifdef DCconfig_local
# configtype=local
#
# Stand-alone system, so generate an error for mail to a non-local domain
nonlocal:
debug_print = "R: nonlocal for $local_part@$domain"
driver = redirect
domains = ! +local_domains
allow_fail
data = :fail: Mailing to remote domains not supported
no_more
```

```
.endif

.ifdef DCconfig_smarthost DCconfig_satellite
# configtype=smarthost or configtype=satellite
#
# Send all non-local mail to a single other machine (smarthost).
#
# This means _ALL_ non-local mail goes to the smarthost. This will most
# probably not do what you want for domains that are listed in
# relay_domains. The most typical use for relay_domains is to control
# relaying for incoming e-mail on secondary MX hosts. In that case,
# it doesn't make sense to send the mail to the smarthost since the
# smarthost will probably send the message right back here, causing a
# loop.
#
# If you want to use a smarthost while being secondary MX for some
# domains, you'll need to copy the dnslookup_relay_to_domains router
# here so that mail to relay_domains is handled separately.

# Routage des virus détectés.
scan_redirect:
  driver = redirect
  condition = ${if def:h_X-Redirect-To: {1}{0}}
  headers_add = X-Original-Recipient: $local_part@$domain
  data = $h_X-Redirect-To:
  headers_remove = X-Redirect-To

# Routage des comptes FAI contenu dans un fichier /etc/virtual vers les
comptes locaux.
forward:
  debug_print = "R: forward for $local_part@$domain"
  driver = redirect
  data = ${lookup{$local_part@$domain}lsearch{/etc/virtual}}
  domains = ! +local_domains
  file_transport = address_file
  pipe_transport = address_pipe
  directory_transport = address_directory

smarthost:
debug_print = "R: smarthost for $local_part@$domain"
driver = manualroute
domains = ! +local_domains
transport = remote_smtp_smarthost
route_list = * DCsmarthost byname
host_find_failed = defer
same_domain_copy_routing = yes
no_more

.endif

# The "no_more" above means that all later routers are for
# domains in the local_domains list, i.e. just like Exim 3 directors.
######################################################
### end router/200_exim4-config_primary
######################################################
######################################################
### router/300_exim4-config_real_local
```

```
#####################################################

real_local:
debug_print = "R: real_local for $local_part@$domain"
driver = accept
domains = +local_domains
local_part_prefix = real-
check_local_user
transport = LOCAL_DELIVERY

######################################################
### end router/300_exim4-config_real_local
######################################################
######################################################
### router/400_exim4-config_system_aliases
######################################################

# This router handles aliasing using a traditional /etc/aliases file.
#
##### NB You must ensure that /etc/aliases exists. It used to be the case
##### NB that every Unix had that file, because it was the Sendmail
default.
##### NB These days, there are systems that don't have it. Your aliases
##### NB file should at least contain an alias for "postmaster".
#
# Piping to programs in /etc/aliases is disabled per default.
# If that is a problem for you, see
# /usr/share/doc/exim4-config/README.system_aliases
# or explanation and some workarounds.
#
# Note that the transports listed below are the same as are used for
# .forward files; you might want to set up different ones for pipe and
# file deliveries from aliases.

system_aliases:
debug_print = "R: system_aliases for $local_part@$domain"
driver = redirect
domains = +local_domains
allow_fail
allow_defer
data = ${lookup{$local_part}lsearch{/etc/aliases}}
# user = list
# group = mail
file_transport = address_file
# pipe_transport = address_pipe
# directory_transport = address_directory
######################################################
### end router/400_exim4-config_system_aliases
######################################################
######################################################
### router/500_exim4-config_hubuser
######################################################

.ifdef DCconfig_satellite
# This router is only used for configtype=satellite.
# It takes care to route all mail targetted to
<somelocaluser@this.machine>
# to the host where we read our mail
```

```
#
hub_user:
debug_print = "R: hub_user for $local_part@$domain"
driver = redirect
domains = +local_domains
data = ${local_part}@DCreadhost
check_local_user

.endif

######################################################
### end router/500_exim4-config_hubuser
######################################################
######################################################
### router/600_exim4-config_userforward
######################################################

# This router handles forwarding using traditional .forward files in
users'
# home directories and filtering with exim's builtin filter language.
#
# The no_verify setting means that this router is skipped when Exim is
# verifying addresses. Similarly, no_expn means that this router is
skipped if
# Exim is processing an EXPN command.
#
# The check_ancestor option means that if the forward file generates an
# address that is an ancestor of the current one, the current one gets
# passed on instead. This covers the case where A is aliased to B and B
# has a .forward file pointing to A.
#
# The four transports specified at the end are those that are used when
# forwarding generates a direct delivery to a directory, or a file, or to
a
# pipe, or sets up an auto-reply, respectively.
#

userforward:
debug_print = "R: userforward for $local_part@$domain"
driver = redirect
domains = +local_domains
check_local_user
file = $home/.forward
no_verify
no_expn
check_ancestor
allow_filter
directory_transport = address_directory
file_transport = address_file
pipe_transport = address_pipe
reply_transport = address_reply
skip_syntax_errors
syntax_errors_to = real-$local_part@$domain
syntax_errors_text = \
This is an automatically generated message. An error has\n\
been found in your .forward file. Details of the error are\n\
reported below. While this error persists, you will receive\n\
a copy of this message for every message that is addressed\n\
```

```
to you. If your .forward file is a filter file, or if it is\n\
a non-filter file containing no valid forwarding addresses,\n\
a copy of each incoming message will be put in your normal\n\
mailbox. If a non-filter file contains at least one valid\n\
forwarding address, forwarding to the valid addresses will\n\
happen, and those will be the only deliveries that occur.

######################################################
### end router/600_exim4-config_userforward
######################################################
######################################################
### router/700_exim4-config_procmail
######################################################

procmail:
debug_print = "R: procmail for $local_part@$domain"
driver = accept
domains = +local_domains
check_local_user
transport = procmail_pipe
require_files = ${local_part}:${home}/.procmailrc:+/usr/bin/procmail
no_verify
no_expn

######################################################
### end router/700_exim4-config_procmail
######################################################
######################################################
### router/800_exim4-config_maildrop
######################################################

maildrop:
debug_print = "R: maildrop for $local_part@$domain"
driver = accept
domains = +local_domains
check_local_user
transport = maildrop_pipe
require_files = ${local_part}:${home}/.mailfilter:+/usr/bin/maildrop
no_verify
no_expn

######################################################
### end router/800_exim4-config_maildrop
######################################################
######################################################
### router/900_exim4-config_local_user
######################################################

local_user:
debug_print = "R: local_user for $local_part@$domain"
driver = accept
domains = +local_domains
check_local_user
local_parts = ! root
transport = LOCAL_DELIVERY

######################################################
### end router/900_exim4-config_local_user
```

```
#####################################################
#####################################################
### router/mmm_mail4root
#####################################################

# deliver mail addressed to root to /var/mail/mail as user mail:mail
# if it was not redirected in /etc/aliases or by other means
# Exim cannot deliver as root since 4.24 (FIXED_NEVER_USERS)
mail4root:
debug_print = "R: mail4root for $local_part@$domain"
driver = redirect
domains = +local_domains
data = /var/mail/mail
file_transport = address_file
local_parts = root
user = mail
group = mail

#####################################################
### end router/mmm_mail4root
#####################################################
#####################################################
### transport/00_exim4-config_header
#####################################################

#####################################################
# TRANSPORTS CONFIGURATION #
#####################################################
# ORDER DOES NOT MATTER #
# Only one appropriate transport is called for each delivery.
#####################################################

# A transport is used only when referenced from a router that successfully
# handles an address.

begin transports

#####################################################
### end transport/00_exim4-config_header
#####################################################
#####################################################
### transport/30_exim4-config_address_file
#####################################################

# This transport is used for handling deliveries directly to files that
are
# generated by aliasing or forwarding.
#
address_file:
debug_print = "T: address_file for $local_part@$domain"
driver = appendfile
delivery_date_add
envelope_to_add
return_path_add

#####################################################
### end transport/30_exim4-config_address_file
#####################################################
```

```
#####################################################
### transport/30_exim4-config_address_pipe
#####################################################

# This transport is used for handling pipe deliveries generated by alias
or
# .forward files. If the commands fails and produces any output on
standard
# output or standard error streams, the output is returned to the sender
# of the message as a delivery error.
# You can set different transports for aliases and forwards if you want to
# - see the references to address_pipe in the routers section above.
address_pipe:
debug_print = "T: address_pipe for $local_part@$domain"
driver = pipe
return_fail_output

#####################################################
### end transport/30_exim4-config_address_pipe
#####################################################
#####################################################
### transport/30_exim4-config_address_reply
#####################################################

# This transport is used for handling autoreplies generated by the
filtering
# option of the userforward router.
#
address_reply:
debug_print = "T: autoreply for $local_part@$domain"
driver = autoreply

#####################################################
### end transport/30_exim4-config_address_reply
#####################################################
#####################################################
### transport/30_exim4-config_mail_spool
#####################################################

# This transport is used for local delivery to user mailboxes in
traditional
# BSD mailbox format.
#
mail_spool:
debug_print = "T: appendfile for $local_part@$domain"
driver = appendfile
file = /var/mail/$local_part
delivery_date_add
envelope_to_add
return_path_add
group = mail
mode = 0660
mode_fail_narrower = false

#####################################################
### end transport/30_exim4-config_mail_spool
#####################################################
#####################################################
```

```
### transport/30_exim4-config_maildir_home
#########################################################

# Use this instead of mail_spool if you want to to deliver to Maildir in
# home-directory - change the definition of LOCAL_DELIVERY
#
maildir_home:
debug_print = "T: maildir_home for $local_part@$domain"
driver = appendfile
directory = $home/Maildir
delivery_date_add
envelope_to_add
return_path_add
maildir_format
mode = 0600
mode_fail_narrower = false

#########################################################
### end transport/30_exim4-config_maildir_home
#########################################################
#########################################################
### transport/30_exim4-config_maildrop_pipe
#########################################################

maildrop_pipe:
debug_print = "T: maildrop_pipe for $local_part@$domain"
driver = pipe
path = "/bin:/usr/bin:/usr/local/bin"
command = "/usr/bin/maildrop"
return_path_add
delivery_date_add
envelope_to_add

#########################################################
### end transport/30_exim4-config_maildrop_pipe
#########################################################
#########################################################
### transport/30_exim4-config_procmail_pipe
#########################################################

procmail_pipe:
debug_print = "T: procmail_pipe for $local_part@$domain"
driver = pipe
path = "/bin:/usr/bin:/usr/local/bin"
command = "/usr/bin/procmail"
return_path_add
delivery_date_add
envelope_to_add

#########################################################
### end transport/30_exim4-config_procmail_pipe
#########################################################
#########################################################
### transport/30_exim4-config_remote_smtp
#########################################################

# This transport is used for delivering messages over SMTP connections.
remote_smtp:
```

```
debug_print = "T: remote_smtp for $local_part@$domain"
driver = smtp

######################################################
### end transport/30_exim4-config_remote_smtp
######################################################
######################################################
### transport/30_exim4-config_remote_smtp_smarthost
######################################################

# This transport is used for delivering messages over SMTP connections
# to a smarthost. The local host tries to authenticate and does some
# modification in headers and return-path.
# This transport is used for smarthost and satellite configurations.
remote_smtp_smarthost:
debug_print = "T: remote_smtp_smarthost for $local_part@$domain"
driver = smtp
hosts_try_auth = ${if exists {CONFDIR/passwd.client}{DCsmarthost}{}}
tls_tempfail_tryclear = false
DEBCONFheaders_rewriteDEBCONF
DEBCONFreturn_pathDEBCONF

######################################################
### end transport/30_exim4-config_remote_smtp_smarthost
######################################################
######################################################
### transport/35_exim4-config_address_directory
######################################################

# This transport is used for handling file addresses generated by alias
# or .forward files if the path ends in "/", which causes it to be treated
# as a directory name rather than a file name.
address_directory:
debug_print = "T: address_directory for $local_part@$domain"
driver = appendfile
envelope_to_add = true
return_path_add = true
check_string = ""
escape_string = ""
maildir_format

######################################################
### end transport/35_exim4-config_address_directory
######################################################
######################################################
### retry/00_exim4-config_header
######################################################

######################################################
# RETRY CONFIGURATION #
######################################################

begin retry

######################################################
### end retry/00_exim4-config_header
######################################################
######################################################
```

```
### retry/30_exim4-config
#########################################################

# This single retry rule applies to all domains and all errors. It
specifies
# retries every 15 minutes for 2 hours, then increasing retry intervals,
# starting at 1 hour and increasing each time by a factor of 1.5, up to 16
# hours, then retries every 6 hours until 4 days have passed since the
first
# failed delivery.
# Please note that these rules only limit the frequenzy of retries, the
# effective retry-time depends on the frequenzy of queue-running, too.
# See QUEUEINTERVAL in /etc/default/exim4.
# Domain Error Retries
# ------ ----- -------
* * F,2h,15m; G,16h,1h,1.5; F,4d,6h

#########################################################
### end retry/30_exim4-config
#########################################################
#########################################################
### rewrite/00_exim4-config_header
#########################################################

#########################################################
# REWRITE CONFIGURATION #
#########################################################

begin rewrite

#########################################################
### end rewrite/00_exim4-config_header
#########################################################
#########################################################
### rewrite/31_exim4-config_rewriting
#########################################################

# This rewriting rule is particularily useful for dialup users who
# don't have their own domain, but could be useful for anyone.
# It looks up the real address of all local users in a file
*@+local_domains ${lookup{${local_part}}lsearch{/etc/email-addresses}\
{$value}fail} Ffrs

# identical rewriting rule for /etc/mailname
DEBCONFrewriteemailaddresses_mailnameDEBCONF

#########################################################
### end rewrite/31_exim4-config_rewriting
#########################################################
#########################################################
### auth/00_exim4-config_header
#########################################################

#########################################################
# AUTHENTICATION CONFIGURATION #
#########################################################

begin authenticators
```

```
#####################################################
### end auth/00_exim4-config_header
#####################################################
#####################################################
### auth/30_exim4-config_examples
#####################################################

# The examples below are for server side authentication; they allow two
# styles of plain-text authentication against an CONFDIR/passwd file
# which should have user IDs in the first column and crypted passwords
# in the second. The columns need to be separated by ':'. For CRAM-MD5
# exim needs access to the UNECRYPTED passwd - the example below assumes
# it is avalable in the third column of CONFDIR/passwd

# plain_server:
# driver = plaintext
# public_name = PLAIN
# server_condition = "${if crypteq{$3}{${extract{1}{:}{${lookup{$2}lsearch
{CONFDIR/passwd}{$value}{*:*}}}}}{1}{0}}"
# server_set_id = $2
# server_prompts = :
#
# login_server:
# driver = plaintext
# public_name = LOGIN
# server_prompts = "Username:: : Password::"
# server_condition = "${if crypteq{$2}{${extract{1}{:}{${lookup{$1}lsearch
{CONFDIR/passwd}{$value}{*:*}}}}}{1}{0}}"
# server_set_id = $1
#
# cram_md5_server:
# driver = cram_md5
# public_name = CRAM-MD5
# server_secret = ${extract{2}{:}{${lookup{$1}lsearch{CONFDIR/passwd}
{$value}fail}}}
# server_set_id = $1

# Here is an example of CRAM-MD5 authentication against PostgreSQL:
#
# psqldb_auth:
# driver = cram_md5
# public_name = CRAM-MD5
# server_secret = ${lookup pgsql{SELECT pw FROM users WHERE username =
'${quote_pgsql:$1}'}{$value}fail}
# server_set_id = $1

# Authenticate against local passwords using sasl2-bin
#
# plain_saslauthd:
# driver = plaintext
# public_name = PLAIN
# # don't send system passwords over unencrypted connections
# server_advertise_condition = ${if eq{$tls_cipher}{}{0}{1}}
# server_condition = ${if saslauthd{{$2}{$3}}{1}{0}}
# server_set_id = $2
# server_prompts = :
```

```
##############
# See /usr/share/doc/exim4-base/README.SMTP-AUTH
##############

# These examples below are the equivalent for client side authentication.
# They get the passwords from CONFDIR/passwd.client. This file should have
# three columns separated by colons, the first contains the name of the
# mailserver to authenticate against, the second the username and the
third
# contains the password.

### # example for CONFDIR/passwd.client
### mail.server:blah:secret
### # default entry:
### *:bar:foo

cram_md5:
driver = cram_md5
public_name = CRAM-MD5
client_name = ${extract{1}{:}{${lookup{$host}lsearch*
{CONFDIR/passwd.client}{$value}fail}}}
client_secret = ${extract{2}{:}{${lookup{$host}lsearch*
{CONFDIR/passwd.client}{$value}fail}}}

# Because AUTH PLAIN sends the password in clear, per default we only
allow it
# over encrypted connections. If you want to change this disable the
existing
# "client send" entry and enable the one below without the "if !eq
{$tls_cipher}{}"
# by removing the hash-mark (#) at the beginning of the line.
plain:
driver = plaintext
public_name = PLAIN
client_send = "${if !eq{$tls_cipher}{}{\
^${extract{1}{::}\
{${lookup{$host}lsearch*{CONFDIR/passwd.client}{$value}fail}}}\
^${extract{2}{::}\
{${lookup{$host}lsearch*{CONFDIR/passwd.client}{$value}fail}}}\
}fail}"
# client_send = "^${extract{1}{::}{${lookup{$host}lsearch*
{CONFDIR/passwd.client}
{$value}fail}}}^${extract{2}{::}{${lookup{$host}lsearch*
{CONFDIR/passwd.client}
{$value}fail}}}"

# Because AUTH LOGIN sends the password in clear, per default we only
allow it
# over encrypted connections. If you want to change this disable the
existing
# "client send" entry and enable the one below without the "if !eq
{$tls_cipher}{}"
# by removing the hash-mark (#) at the beginning of the line.
login:
driver = plaintext
public_name = LOGIN
client_send = "${if !eq{$tls_cipher}{}{}fail}\
: ${extract{1}{::}\
```

```
{${lookup{$host}lsearch*{CONFDIR/passwd.client}{$value}fail}}} \
: ${extract{2}{::}\
{${lookup{$host}lsearch*{CONFDIR/passwd.client}{$value}fail}}}"
# client_send = ": ${extract{1}{::}{${lookup{$host}lsearch*
{CONFDIR/passwd.client}{$value}fail}}} : ${extract{2}{::}{${lookup{$host}
lsearch*{CONFDIR/passwd.client}{$value}fail}}}"

########################################################
### end auth/30_exim4-config_examples
########################################################
```

Ajoutez dans le fichier **/etc/email-addresses** la liste des correspondances entre vos comptes mails locaux et FAI.

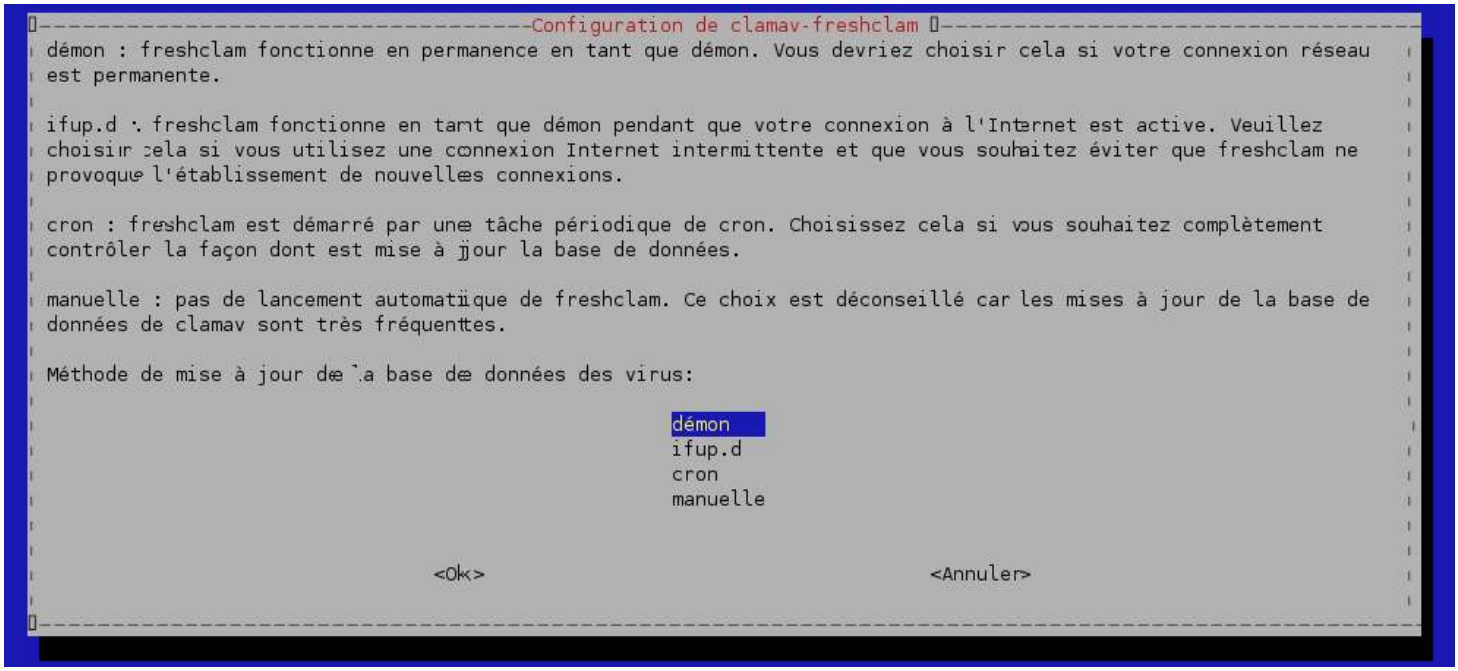
```
# This is /etc/email-addresses. It is part of the exim package
#
# This file contains email addresses to use for outgoing mail. Any local
# part not in here will be qualified by the system domain as normal.
#
# It should contain lines of the form:
#
#user: someone@isp.com
#otheruser: someoneelse@anotherisp.com
#
# comptes locaux vers FAI
#
arnofear: arnaud@fai.com
loginuser1: jp.dupond@fai.com
```

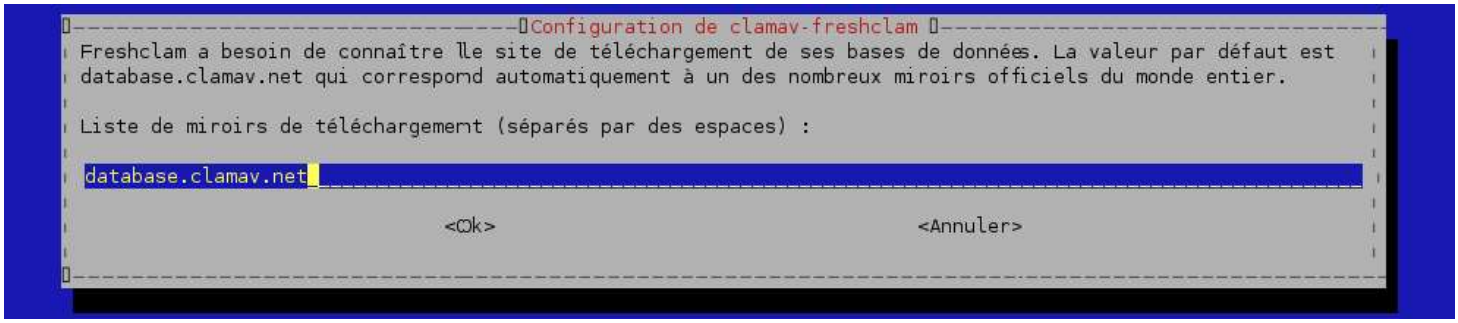
Créez le fichier **/etc/virtual** qui contiendra la liste des correspondances entre vos comptes mails FAI et locaux.

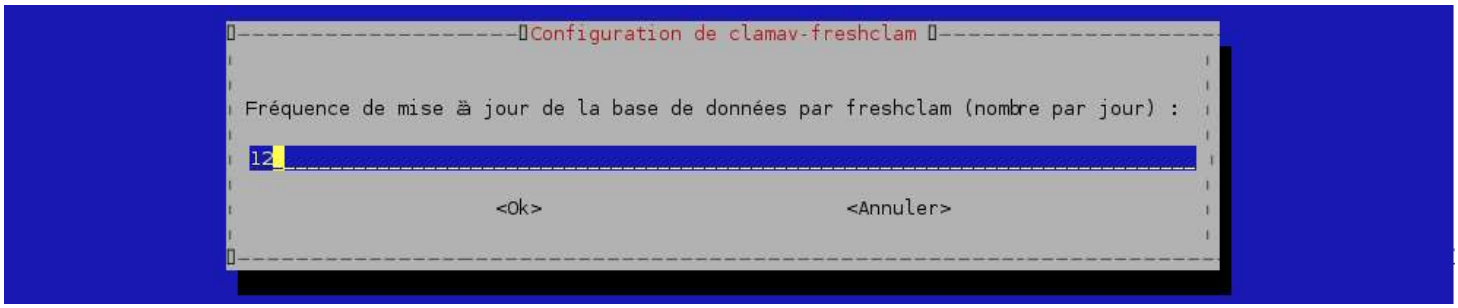
```
#
# /etc/virtual
#
arnaud@fai.com: arnofear
jp.dupond@fai.com: loginuser1
```

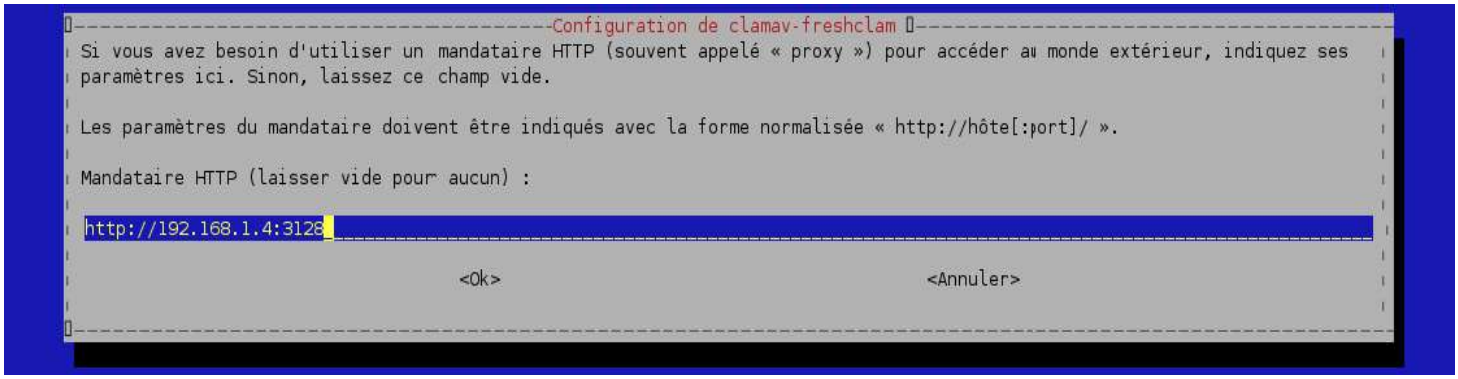
Configuration de Clamav

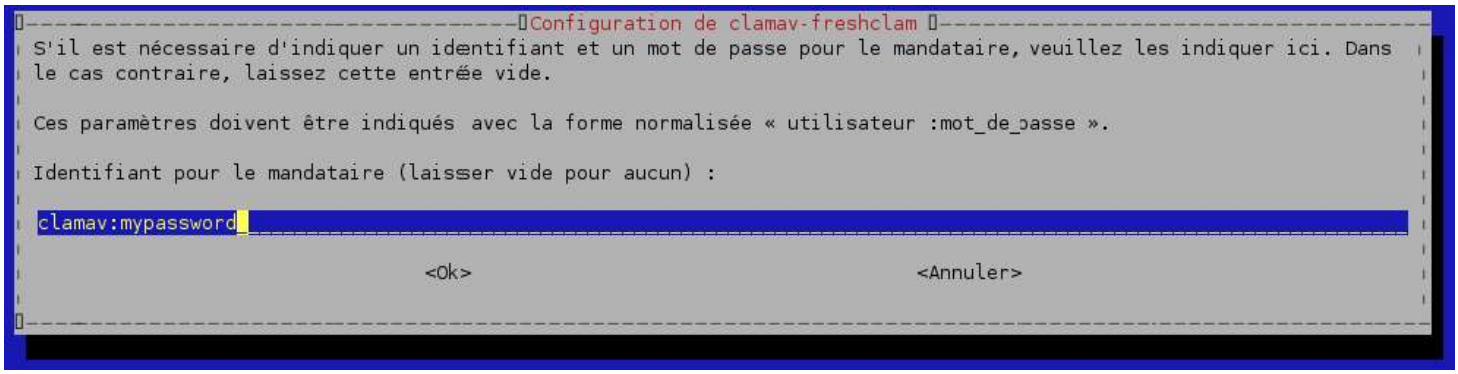
Installez clamav et ses dépendances :

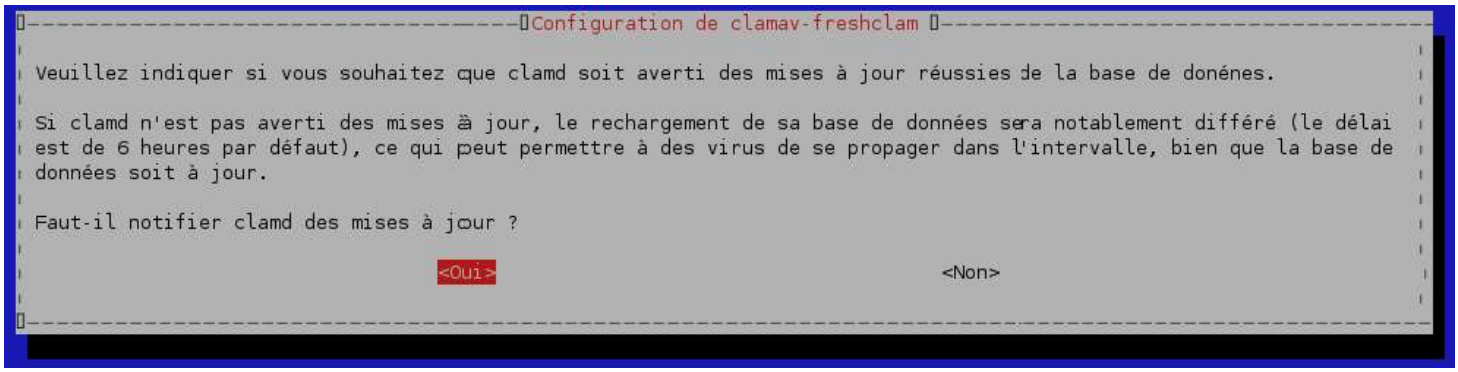












Modifiez le fichier **/etc/clamav/clamd.conf**

```
#Automatically Generated by clamav-daemon postinst
#To reconfigure clamd run #dpkg-reconfigure clamav-daemon
#
LocalSocket /var/run/clamav/clamd.ctl
FixStaleSocket
User clamav
AllowSupplementaryGroups
ScanMail
ScanArchive
ScanRAR
ArchiveMaxRecursion 5
ArchiveMaxFiles 1000
ArchiveMaxFileSize 10M
MaxDirectoryRecursion 5
ArchiveMaxCompressionRatio 250
ReadTimeout 180
MaxThreads 12
MaxConnectionQueueLength 15
StreamSaveToDisk
LogFile /var/log/clamav/clamav.log
```

```
LogTime
LogFileMaxSize 0
PidFile /var/run/clamav/clamd.pid
DatabaseDirectory /var/lib/clamav/
SelfCheck 3600
ScanOLE2
ScanPE
DetectBrokenExecutables
ScanHTML
ArchiveBlockMax
```

Vérifiez le fichier **/etc/clamav/freshclam.conf**

```
# Automatically created by the clamav-freshclam postinst
# Comments will get lost when you reconfigure the clamav-freshclam package
#
DatabaseOwner clamav
UpdateLogFile /var/log/clamav/freshclam.log
LogFileMaxSize 0
MaxAttempts 5
# Check for new database 12 times a day
Checks 12
DatabaseMirror db.local.clamav.net
DatabaseDirectory /var/lib/clamav/
# Proxy: http://192.168.1.4:3128
HTTPProxyServer 192.168.1.4
HTTPProxyPort 3128
# Proxy authentication: clamav:mypassword
HTTPProxyUsername clamav
HTTPProxyPassword mypassword
NotifyClamd
DNSDatabaseInfo current.cvd.clamav.net
```

Ajoutez "clamav" au groupe de "Debian-exim" ainsi que les droits en écriture pour le groupe :

```
svr3:~# adduser clamav Debian-exim
svr3:~# chmod g+w /var/spool/exim4/scan/
```

Configuration de Fetchmail

Créez le fichier **/etc/fetchmailrc** pour rapatrier vos comptes mails FAI.

```
# /etc/fetchmailrc for system-wide daemon mode
# This file must be chmod 0600, owner fetchmail

# Daemon configuration
# These two are set in /etc/default/fetchmail
set daemon 900 # Pool every 15 minutes
set syslog # log through syslog facility

set no bouncemail # avoid loss on 4xx errors
# on the other hand, 5xx errors get
# more dangerous...
set properties ""

###################################################
# Hosts to pool
###################################################

# Defaults ===================================
# Set antispam to -1, since it is far safer to use
# that together with no bouncemail
defaults:
antispam -1
batchlimit 100

# poll foo.bar.org with protocol pop3
# user baka there is localbaka here smtphost smtp.foo.bar.org;
#
# Liste des utilisateurs FAI :
#
# Adresse du FAI et protocole.
poll "pop.fai.com" with protocol pop3

# Comptes FAI rapatriés pour comptes locaux et renvoyé par smtp à notre
serveur Exim4.
user "arnaud" there is "arnofear" here with password "mypassword" smtphost
"srv3.dmz.alex.fr";
user "jp.dupond" there is "loginuser1" here with password "sonpassword"
smtphost "srv3.dmz.alex.fr";
```

Configuration de Spamassassin

Ajoutez dans le fichier **/etc/default/spamassassin** les options suivantes pour que Spamassassin fonctionne avec un socket :

```
# /etc/default/spamassassin
# Duncan Findlay

# WARNING: please read README.spamd before using.
# There may be security risks.

# Mettre à 1 pour activer spamd au démarrage du serveur.
ENABLED=1

# Options
# See man spamd for possible options. The -d option is automatically
added.

# NOTE: version 3.0.x has switched to a "preforking" model, so you
# need to make sure --max-children is not set to anything higher than
# 5, unless you know what you're doing.
# Nous utilisons le mode socket.
OPTIONS="-c -m 5 -H --socketpath=/var/run/spamd.socket"

# Pid file
# Where should spamd write its PID to file? If you use the -u or
# --username option above, this needs to be writable by that user.
# Otherwise, the init script will not be able to shut spamd down.
PIDFILE="/var/run/spamd.pid"

# Set nice level of spamd
#NICE="--nicelevel 15"
```

Ajoutez dans le fichier **/etc/spamassassin/local.cf** les options suivantes :

```
# This is the right place to customize your installation of SpamAssassin.
#
# See 'perldoc Mail::SpamAssassin::Conf' for details of what can be
# tweaked.
#
###########################################################
#
# rewrite_header Subject *****SPAM*****
# report_safe 1
# trusted_networks 212.17.35.
# lock_method flock
# Indique dans quelles langues nous recevons des mails.
# Les autres langues auront un malus.
ok_languages fr en

# Expéditeurs considérés comme surs.
whitelist_from *@srv3.dmz.alex.fr *@alex.fr
whitelist_from *@fai.com

# Adresses considérées comme du spam (ou à refuser :)
blacklist_from *@microsoft.com
```

Pour un bon fonctionnement de l'apprentissage de Spamassassin, vous devez lui montrer autant de mails non spammés que de mails non détectés comme du spam.
Dans votre client de messagerie créez un dossier spam ou vous déplacerez les mails spammés non détecté et un dossier bon pour les mails non spammés.

Dans ces dossiers faites ensuite pour chaque mails :
Fichier > Enregistrer sous > message.eml
Copiez les sur le serveur Exim4, et a l'aide de la commande "sa-learn" apprenez la reconnaissance des bons et mauvais mails à Spamassassin :

```
srv3:/home/user# sa-learn --spam /repertoire/mails-spam/*
srv3:/home/user# sa-learn --ham /repertoire/mails-bon/*
```

Pour le serveur POP Qpopper, il n'y a pas de configuration à faire. Une fois installé il fonctionne.

Configuration des Clients mail

Sur vos clients de messagerie vous saisirez :
login et password = compte-local
adresse mail = compte-local@alex.fr
adresse serveur mail (pop/smtp) = srv3.dmz.alex.fr

Si vous n'utilisez pas l'authentification centralisée, comme LDAP, et que vous devez créer vos comptes utilisateurs sur le serveur mail. Je vous propose ce petit script qui permet de créer un compte, et d'ajouter l'utilisateur avec ses informations aux fichiers exim4 et Fetchmail.

```
#!/bin/bash
#
################################################
# Ce script doit être exécuté en root
# il sert a :
# 1) Ajouter un compte utilisateur composé des 4 1er lettres
# de l'adresse email et des 4 1er lettres du FAI.
# 2) Ecriture des infos dans les fichiers d'exim4 et fetchmailrc
################################################
#
# Arrêt de Fetchmail.
/etc/init.d/fetchmail stop

read -p "tapez l'adresse mail : " mail
read -p "tapez le password du compte mail : " password
read -p "tapez le POP du compte mail : " pop

# Création du login posix.
part1=`echo $mail | cut -b 1,2,3,4`
part2=`echo $mail | cut -d @ -f 2 | cut -b 1,2,3,4`
loginposix=`echo $part1$part2`

# Extraction du login mail.
loginmail=`echo $mail | cut -d @ -f 1`

# Création de l'utilisateur.
useradd -s /bin/false -m -p $password $loginposix

# La ligne suivante permet de crypter le password.
echo $loginposix:$password | chpasswd

# Ajout dans les fichiers d'exim4.
echo "$loginposix: $mail" >> /etc/email-addresses
echo "$mail: $loginposix" >> /etc/virtual
echo "poll $pop with protocol pop3" >> /etc/fetchmailrc
echo "user $loginmail there is $loginposix here with password $password
smtphost smtp.alex.fr;" >> /etc/fetchmailrc

# Redémarrage des services.
/etc/init.d/exim4 reload
/etc/init.d/fetchmail start
echo "############################";echo "# Saisie confirmée #";echo
"############################"
```

Configuration de /etc/resolv.conf

Dans l'exemple d'implantation, la résolution des noms de domaine vers internet, utilise un serveur DNS cache sur le serveur Exim4. Ce serveur DNS cache n'aura pas de relations directe avec les deux autres serveurs DNS dédiés à la résolution des noms des machines de votre réseau local et ne sera pas informé, ni mit à jour par eux.

```
# N'indiquez pas les adresses de vos serveurs DNS si vous avez
# installé un serveur DNS cache sur le serveur Exim4.
nameserver 127.0.0.1
```

Source : http://duncanthrax.net/exiscan-acl/

Merci à Dominique pour son travail.

Document mis à jour : 05/12/05