

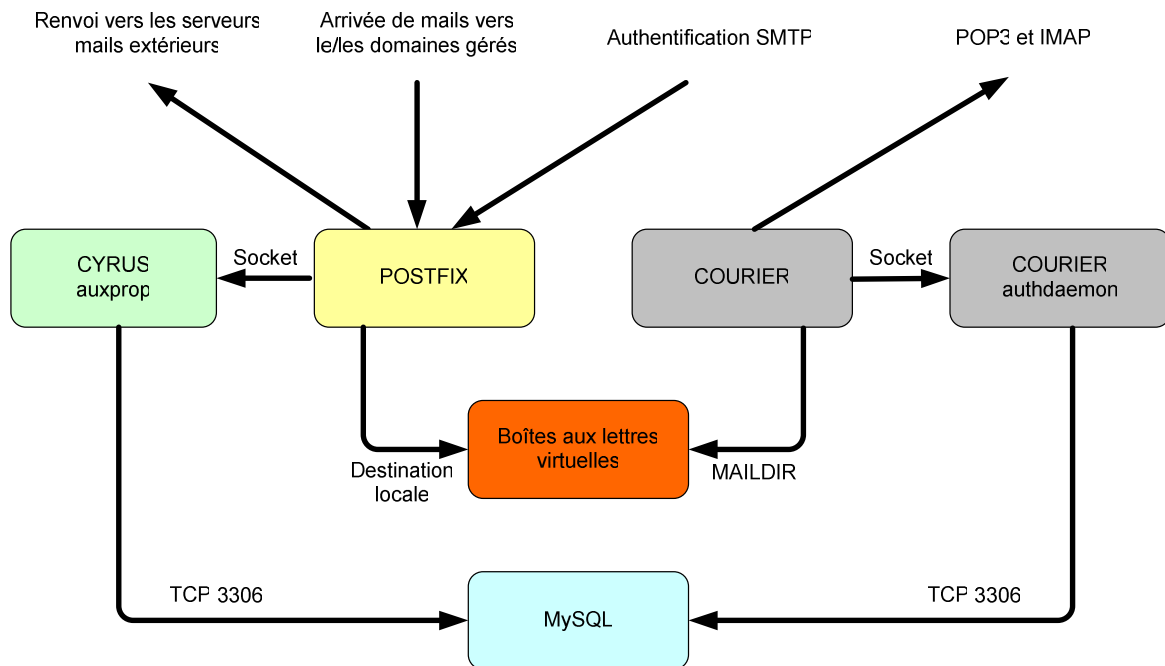
INSTALLATION DE POSTFIX + ANTIVIRUS/ANTISPAM + WEBMAIL IMP

Pierre PATAKI (pierre@pataki.nom.fr)
David KALMAR (dkalmar@free.fr)

Prérequis

- Debian 3.1 SARGE proprement installée

Explications du fonctionnement du système à installer



Installation des paquets

Après une installation clean de Debian 3.1 SARGE, on peut entamer l'installation des outils qui nous seront nécessaires pour le fonctionnement d'un serveur POSTFIX accompagné d'un WEBMAIL et des outils ANTISPAM et ANTIVIRUS.

```
aptitude install vim
aptitude install postfix (Choisir mode Local seulement)
aptitude install postfix-mysql
aptitude install postfix-doc
aptitude install mysql-client
aptitude install mysql-server
aptitude install courier-authdaemon
aptitude install courier-authmysql
aptitude install courier-pop
aptitude install courier-pop-ssl
aptitude install courier-imap
aptitude install courier-imap-ssl
```

```
aptitude install postfix-tls
aptitude install libsasl2
aptitude install libsasl2-modules
aptitude install libsasl2-modules-sql
aptitude install openssl
aptitude install apache
aptitude install apache-ssl
aptitude install php4
aptitude install php4-cgi
aptitude install php4-imap
aptitude install php4-pear-log
aptitude install phpmyadmin
aptitude install php4-pear
aptitude install php4-pear-log
aptitude install php4-domxml
aptitude install php-mail-mime
aptitude install amavisd-new
aptitude install spamassassin
aptitude install clamav
aptitude install clamav-daemon
aptitude install zoo
aptitude install unzip
aptitude install unarj
```

VIM est à installer par coquetterie, il s'agit là simplement d'activer la coloration syntaxique, utile lors de l'édition de fichiers de configuration.

POSTFIX est le serveur de messagerie que nous désirons installer et configurer. C'est lui qui aura la tâche de réception des emails et leur redistribution dans les boîtes mail.

MySQL est le système de base de donnée que nous allons utiliser pour recenser les domaines gérés par POSTFIX ainsi que les utilisateurs, les redirections et les mots de passes.

COURIER est un système alternatif à POSTFIX. Nous n'utiliserons ici que ses fonctionnalités de POP3/IMAP pour autoriser les utilisateurs à accéder à leurs comptes email.

SASL si un utilisateur se connecte depuis une IP autre que celle de votre réseau, SASL se chargera d'activer l'identification de l'utilisateur par identifiant et mot de passe à votre serveur SMTP.

AMAVIS est un filtre ANTIVIRUS et ANTISPAM de POSTFIX. Il est aussi connu sous le nom de SPAMASSASSIN.

Les outils de décompression comme ZOO UNZIP et UNARJ sont utiles pour l'antivirus qui pourra ainsi vérifier dans les archives envoyés par email s'ils ne contiennent pas de virus.

PHPMYADMIN est utile pour vérifier les informations entrées dans la base de donnée. Utile également pour l'ajout des informations dans la base MySQL, pour ceux qui ont des problèmes à taper leurs requêtes en ligne de commande.

APACHE et PHP nous serviront pour le webmail IMP de HORDE.

Configuration du serveur MySQL

Après l'installation du paquet MySQL-Server, le mot de passe de l'utilisateur root est à définir.

```
mysqladmin -u root password motdepasse-root
```

Ensuite, il nous faut créer la base MySQL nommée 'postfix_db'.

```
mysqladmin -u root -p create postfix_db
```

Il faut également créer un utilisateur appelé 'postfix' avec un mot de passe au choix (ici ce sera 'banane'). Il est important que celui-ci ai les droits pour se connecter depuis '127.0.0.1', car il sera configuré ainsi par la suite dans ce tutoriel.

```
echo "grant select on postfix_db.* to postfix@127.0.0.1 identified \
    by 'banane';" | mysql -u root -p
```

Pour prendre en compte les modifications des utilisateurs MySQL il faut effectuer un flush de celle-ci.

```
echo "flush privileges;" | mysql -u root -p
```

Enfin, il faut importer la structure des tables de la base 'postfix_db' pour que POSTFIX puisse fonctionner.

```
echo "CREATE TABLE domains ( \
    domain varchar(50) NOT NULL, \
    PRIMARY KEY (domain) ) \
    TYPE=MyISAM; \
CREATE TABLE forwardings ( \
    source varchar(80) NOT NULL, \
    destination TEXT NOT NULL, \
    PRIMARY KEY (source) ) \
    TYPE=MyISAM; \
CREATE TABLE users ( \
    email varchar(80) NOT NULL, \
    password varchar(20) NOT NULL, \
    PRIMARY KEY (email) \
) TYPE=MyISAM;" | mysql -u root -p postfix_db
```

Configuration de POSTFIX pour MySQL

Nous devons créer des fichiers de configuration pour indiquer les informations de connexion à la base MySQL de POSTFIX. Ici nous retrouvons entre autres, les identifiants et mots de passes pour accéder à MySQL, mais également les requêtes à effectuer par POSTFIX pour retrouver un utilisateur et ainsi de suite.

Note : Il est important de conserver la variable 'hosts' à '127.0.0.1' car si nous indiquons à POSTFIX la valeur 'localhost', il va tenter de se connecter au serveur MySQL via le socket, ce que nous ne voulons pas dans notre cas.

Important : Remplacez évidemment le mot de passe par défaut 'banane' par le mot de passe que vous aurez défini dans la rubrique précédente. Il en sera de même pour tout le reste du tutoriel.

```
/etc/postfix/mysql-virtual_domains.cf
```

```
user = postfix
password = banane
dbname = postfix_db
table = domains
select_field = 'virtual'
where_field = domain
hosts = 127.0.0.1
```

```
/etc/postfix/mysql-virtual_forwardings.cf
```

```
user = postfix
password = banane
dbname = postfix_db
table = forwardings
select_field = destination
where_field = source
hosts = 127.0.0.1
```

```
/etc/postfix/mysql-virtual_mailboxes.cf
```

```
user = postfix
password = banane
dbname = postfix_db
table = users
select_field = CONCAT(SUBSTRING_INDEX(email, '@', -
1), '/', SUBSTRING_INDEX(email, '@', 1), '/')
where_field = email
hosts = 127.0.0.1
```

```
/etc/postfix/mysql-virtual_email2email.cf
```

```
user = postfix
password = banane
dbname = postfix_db
table = users
select_field = email
where_field = email
hosts = 127.0.0.1
```

Création de la configuration Virtual Mail

Pour éviter de devoir créer sur le système un utilisateur par compte email, nous optons pour la création d'un répertoire VMAIL dans lequel seront stockés tous les utilisateurs, sans que nous ayons à les ajouter sur l'OS.

Nous définissons volontairement l'UID et le GID d'utilisateur et du groupe VMAIL à 5000. Cette valeur est importante à retenir car sera utilisée par la suite. Pour faciliter la tâche, conservez notre configuration tel quel.

```
groupadd -g 5000 vmail
useradd -g vmail -u 5000 vmail -d /home/vmail -m
```

Configuration de POSTFIX

POSTFIX peut être complexe à configurer. Il existe une multitude de fonctions, de commandes et de paramètres à personnaliser. Ici nous irons droit au but, pour une configuration standard fonctionnelle. Libre à vous de consulter la documentation sur <http://www.postfix.org/> et optimiser POSTFIX selon vos besoins particuliers.

Nous retrouvons ci-dessous notre fichier de configuration. La variable 'myhostname' est à changer avec le nom d'hôte de votre serveur POSTFIX. Pour notre part, nous avons configuré ce nom d'hôte dans /etc/hosts. Nous définissons ici le chemin vers les différents fichiers de configuration que nous avons créé auparavant, nous permettant l'accès à la base de donnée MySQL.

En fin de fichier, les paramètres 'content_filter' et 'receive_override_options' indiquent les adresses et options pour transmettre les emails vers AMAVIS pour un traitement ANTIVIRUS et ANTISPAM.

/etc/postfix/main.cf

```
smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff = no
append_dot_mydomain = no
myhostname = mail.madservers.cz
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = localhost.localdomain
relayhost =
mynetworks = 127.0.0.0/8
mailbox_command = procmail -a "$EXTENSION"
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
virtual_alias_domains =
virtual_alias_maps = mysql:/etc/postfix/mysql-virtual_forwardings.cf
mysql:/etc/postfix/mysql-virtual_email2email.cf
virtual_mailbox_domains = mysql:/etc/postfix/mysql-virtual_domains.cf
virtual_mailbox_maps = mysql:/etc/postfix/mysql-virtual_mailboxes.cf
virtual_mailbox_base = /home/vmail
virtual_uid_maps = static:5000
virtual_gid_maps = static:5000
smtpd_sasl_auth_enable = yes
broken_sasl_auth_clients = yes
smtpd_recipient_restrictions = permit_mynetworks,
permit_sasl_authenticated, reject_unauth_destination
smtpd_use_tls = yes
smtpd_tls_cert_file = /etc/postfix/smtpd.cert
smtpd_tls_key_file = /etc/postfix/smtpd.key
content_filter = amavis:[127.0.0.1]:10024
receive_override_options = no address mappings
```

Concernant le fichier smtpd.conf, nous devons activer le plugin MySQL pour qu'il puisse être interfacé avec notre serveur de base de donnée. C'est la variable 'auxprop_plugin' qui nous le permet.

/etc/postfix/sasl/smtpd.conf

```
pwcheck_method: auxprop
auxprop_plugin: sql
mech_list: plain login cram-md5 digest-md5
sql_engine: mysql
sql_hostnames: 127.0.0.1
sql_user: postfix
sql_passwd: banane
sql_database: postfix_db
sql_select: select password from users where email='%u@%r'
```

Enfin, il ne nous reste plus qu'à configurer les informations générales de POSTFIX dans le fichier de configuration 'master.cf'. Il est nécessaire d'ajouter ce qui suit dans l'encadré en fin de fichier, afin d'activer la gestion d'AMAVIS.

/etc/postfix/master.cf

```
amavis unix - - - - 2 smtp
    -o smtp_data_done_timeout=1200
    -o smtp_send_xforward_command=yes

127.0.0.1:10025 inet n - - - - smtpd
    -o content_filter=
    -o local_recipient_maps=
    -o relay_recipient_maps=
    -o smtpd_restriction_classes=
    -o smtpd_client_restrictions=
    -o smtpd_helo_restrictions=
    -o smtpd_sender_restrictions=
    -o smtpd_recipient_restrictions=permit_mynetworks,reject
    -o mynetworks=127.0.0/8
    -o strict_rfc821_envelopes=yes
    -o
receive override options=no unknown_recipient_checks,no_header_body_checks
```

Génération du certificat SSL

Pour gérer les connexions sécurisées aux serveurs SMTP, POP et IMAP, nous devons générer un certificat SSL. Vous pouvez le faire générer par un organisme tiers de confiance ou le faire vous-même, solution moins chère mais qui peut poser problèmes, notamment lors de l'utilisation d'Outlook Express qui grince des dents lors d'utilisation de certificats auto-validés.

Le certificat doit être déposé dans le même répertoire que POSTFIX et est généré avec une clé RSA de 2048 bits. En France, il est interdit de créer des clés dépassant les 1024 bits. Faites donc attention pour ne pas rentrer en conflit avec la législation de votre pays.

```
openssl req -new -outform PEM -out /etc/postfix/smtpd.cert \
    -newkey rsa:2048 -nodes -keyout /etc/postfix/smtpd.key \
    -keyform PEM -days 3650 -x509
```

Une fois exécutée, la commande va vous demander quelques informations vous concernant. N'hésitez pas à remplir le plus consciencieusement possible celles-ci, elles seront visibles aux utilisateurs qui voudront avoir des informations sur votre certificat SSL.

Configuration du POP3/IMAP

Il faut désormais autoriser les modules d'authentification à accéder à la base de donnée MySQL. Ceci se fait au travers des fichiers de configurations énumérés ci-dessous.

/etc/courier/authdaemonrc

```
authmodulelist="authmysql"  
authmodulelistorig="authcustom authcram authuserdb authldap authpgsql  
authmysql authpam"  
daemons=5  
version=""  
authdaemonvar=/var/run/courier/authdaemon
```

Dans le fichier qui suit, nous indiquons les informations de connexion au serveur MySQL ainsi que le chemin vers le répertoire VMAIL que nous avons créé précédemment. Gardez toujours à l'esprit que le serveur MySQL doit être renseigné comme '127.0.0.1' et non pas 'localhost' pour des questions d'authentification de notre utilisateur 'postfix'.

/etc/courier/authmysqlrc

```
MYSQL_SERVER          127.0.0.1  
MYSQL_USERNAME        postfix  
MYSQL_PASSWORD        banane  
MYSQL_PORT            0  
MYSQL_OPT              0  
MYSQL_DATABASE        postfix_db  
MYSQL_USER_TABLE      users  
MYSQL_CLEAR_PWFIELD   password  
MYSQL_UID_FIELD       5000  
MYSQL_GID_FIELD       5000  
MYSQL_LOGIN_FIELD     email  
MYSQL_HOME_FIELD      "/home/vmail"  
MYSQL_MAILDIR_FIELD   CONCAT(SUBSTRING_INDEX(email,'@',-  
1),'/',SUBSTRING_INDEX(email,'@',1),'/')
```

Ajout des domaines & adresses email dans POSTFIX

Nous arrivons au stade où il est bon de commencer à remplir la base de donnée MySQL avec des valeurs de test pour pouvoir débiter un maximum notre configuration. Nous décidons d'ajouter un domaine intitulé 'madservers.cz' ainsi qu'une adresse email 'banane@madservers.cz' qui s'authentifiera avec le mot de passe 'secret'.

```
echo "INSERT INTO `domains` (`domain`) VALUES \  
('madservers.cz');" | mysql -u root -p postfix_db  
  
echo "INSERT INTO `users` (`email`,`password`) \  
VALUES ('banane@madservers.cz','secret');" | mysql \  
-u root -p postfix_db
```

N'hésitez pas à remplacer ces données par vos propres exemples. De toute manière ce ne sont que des paramètres destinés aux tests et pourront aisément être supprimés de la base de donnée au moment de passer le serveur POSTFIX en production.

Information sur la configuration des comptes et domaines en mode production

Cette catégorie ne concerne que la mise en production de notre solution. Elle récapitule les quelques informations à configurer dans votre base de donnée MySQL pour pouvoir faire gérer des domaines et adresses email par postfix.

Dans la table MySQL 'domains', indiquez tous les domaines gérés par le serveur de mail postfix.

Exemple : madservers.cz

Dans la table MySQL 'users', indiquez toutes les adresses emails des domaines gérés par le serveur de mail postfix.

Exemple : postmaster@madservers.cz

Dans la table MySQL 'forwarding', indiquez les redirections et alias des adresses emails gérées par le serveur de mail postfix.

Exemples :

Source : postmaster@madservers.cz
Destination : pierre@madservers.cz
Bilan : Redirigera les mails de postmaster@madservers.cz vers pierre@madservers.cz

Source : @madservers.cz
Destination : pierre@madservers.cz
Bilan : Redirige tous les utilisateurs non déclarés dans la table 'users' et concernant le domaine madservers.cz vers l'adresse pierre@madservers.cz

Source : @madservers.cz
Destination : @madservers.us
Bilan : Redirige tous les utilisateurs non déclarés dans la table 'users' et concernant le domaine madservers.cz vers leur équivalent sur le domaine madservers.us. Exemple : pierre@madservers.cz vers pierre@madservers.us

Source : info@madservers.cz
Destination : pierre@madservers.cz, david@madservers.cz
Bilan : Redirige les emails envoyés à info@madservers.cz vers les deux destinataires listés, à savoir pierre@madservers.cz et david@madservers.cz

Configuration AMAVIS/SPAMASSASSIN

AMAVIS est destiné à faire le filtrage du SPAM et des VIRUS sur les mails que traitera POSTFIX. Pour qu'il fonctionne convenablement, ouvrez le fichier 'amavisd.conf' et vérifiez les valeurs des variables énoncées ci-dessous. Si besoin est, modifiez les comme indiqué.

La variable @lookup_sql_dsn doit être impérativement modifiée avec vos informations de connexion à la base de donnée MySQL. N'oubliez pas de garder l'adresse '127.0.0.1' sur le port '3306', sauf si vous avez modifié la configuration par défaut de votre serveur MySQL, ce qui n'est pas notre cas.

/etc/amavis/amavisd.conf

```
$mydomain = 'localhost';
@bypass_virus_checks_acl = qw( . );
@bypass_spam_checks_acl = qw( . );
@lookup_sql_dsn =
( ['DBI:mysql:postfix_db;host=127.0.0.1;port=3306', 'postfix', banane'] );
$sql_select_policy = 'SELECT "Y" as local FROM domains WHERE
CONCAT("@",domain) IN (%k)';
$final_virus_destiny = D_DISCARD;
$final_banned_destiny = D_REJECT;
$final_spam_destiny = D_PASS;
$sa_tag_level_deflt = -1000;
$sa_tag2_level_deflt = 5.0;
$sa_kill_level_deflt = 10;
$sa_spam_subject_tag = '***SPAM*** ';
$sa_local_tests_only = 0;
```

Enfin, il faut créer l'utilisateur qu'AMAVIS va utiliser pour effectuer ses traitements. Nous le créons ainsi sous le nom de 'clamav' dans le groupe 'amavis'.

```
adduser clamav amavis
```

Droits d'accès aux fichiers

Il est nécessaire au bon fonctionnement de POSTFIX de lui accorder des droits d'accès à ses fichiers de configuration. Cette phase est la plus critique dans l'ensemble de l'installation de POSTFIX. Si vous avez des problèmes d'envoi ou de réception de vos emails de test, n'hésitez pas à revenir dans cette partie pour effectuer des modifications, car la raison viendra probablement un problème d'accès aux fichiers de configuration.

```
chown root:postfix -R /etc/postfix
chmod 750 -R /etc/postfix
```

Note : Dans sa configuration de base, il est recommandé de garder l'utilisateur 'root' comme propriétaire des fichiers de configuration de POSTFIX. Le groupe 'postfix' lui n'aura accès à ces fichiers qu'en lecture et exécution. Les autres utilisateurs ne doivent avoir aucun accès à ces fichiers dits critiques, car y sont stockées des informations de connexion à la base MySQL de POSTFIX. Pour vérifier que tout est bien configuré, il ne faut pas hésiter à utiliser la commande 'check' de postfix avec '/etc/init.d/postfix check'. Il vous indiquera éventuellement les problèmes de droits qu'il trouvera.

Redémarrage des services

Pour appliquer tous les changements effectués sur les fichiers de configuration de POSTFIX, AMAVIS et les démons d'authentification, il faut redémarrer ceux-ci.

Redémarrage service AMAVIS :

```
/etc/init.d/amavis restart
```

Redémarrage du démon d'authentification :

```
/etc/init.d/courier-authdaemon restart
```

Redémarrage de POSTFIX :

```
/etc/init.d/postfix restart
```

Notre configuration devrait être prête à être utilisée. Il ne nous reste plus qu'à installer HORDE et IMP qui nous permettront d'accéder à nos mails via un webmail ergonomique.

Première phase de tests

Nous allons vérifier que jusque là, nous n'avons aucun problème de configuration. Si vous avez bien suivi pas à pas les instructions données, vous ne devriez pas avoir de soucis avec cette étape.

Pour tester le service nous allons ouvrir deux consoles en 'root' sur le serveur. Dans la première, nous allons entrer la commande suivante :

```
tail -f /var/log/mail.log
```

Celle-ci nous permettra de suivre en direct ce qui se passe sur notre serveur POSTFIX. Dans la seconde console nous allons effectuer un envoi d'email. Celui-ci se fait en quelques étapes. Les commandes à envoyer seront représentées avec le caractère '>' et les réponses du serveur seront représentées par le caractère '<'.

La première est la connexion en TELNET au port 25 de la machine hébergeant le service POSTFIX. Ici, il s'agit de 'localhost'.

```
telnet localhost 25
```

```
< 220 mail.madservers.cz ESMTP Postfix
```

Ensuite, nous devons dire 'bonjour' à POSTFIX en lui indiquant d'où nous provenons. Dans notre exemple nous lui disons que nous provenons du domaine 'supinfo.com'.

```
> ehlo supinfo.com
< 250-mail.madservers.cz
< 250-PIPELINING
< 250-SIZE 10240000
< 250-VERFY
< 250-ETRN
< 250-STARTTLS
< 250-AUTH LOGIN PLAIN DIGEST-MD5 CRAM-MD5
< 250-AUTH=LOGIN PLAIN DIGEST-MD5 CRAM-MD5
< 250 8BITMIME
```

Après, nous devons donner l'adresse email de l'émetteur du mail. Dans notre cas, nous ferons un envoi depuis '46500@supinfo.com'.

```
> mail from:<46500@supinfo.com>
< 250 Ok
```

Puis nous lui disons à qui nous voulons envoyer un message. Ici, ce sera le compte de test que nous avons créé tout à l'heure, à savoir 'banane@madservers.cz'.

```
> rcpt to:<banane@madservers.cz>
< 250 Ok
```

Il ne nous reste plus qu'à indiquer les données que nous souhaitons envoyer.

```
> data
< 354 End data with <CR><LF>.<CR><LF>
> test de message
> .
< 250 Ok: queued as CAC7A9E08B
```

Enfin, nous quittons la connexion au serveur.

```
> quit
< 221 Bye
< Connection closed by foreign host.
```

Si vous voyez dans les fichiers de log déroulant sur la première console des messages semblables à ce qui suit, vous avez convenablement configuré votre postfix. Sinon, lisez attentivement les messages d'erreur et tentez d'effectuer les rectifications nécessaires. Comme il a été dit plus haut, les problèmes les plus récurrents sont ceux des droits d'accès aux fichiers de configuration de postfix. Reportez vous à la section précédente en cas de problème.

```
postfix/smtpd[14861]: BF6509E08C: client=localhost.localdomain[127.0.0.1]
postfix/cleanup[14858]: BF6509E08C: message-
    id=<20060412032647.CAC7A9E08B@mail.madservers.cz>
postfix/qmgr[14731]: BF6509E08C: from=<46500@supinfo.com>, size=769,
    nrcpt=1 (queue active)
amavis[12814]: (12814-02) Passed, <46500@supinfo.com> ->
    <banane@madservers.cz>, Message-ID:
    <20060412032647.CAC7A9E08B@mail.madservers.cz>, Hits: -
postfix/smtp[14859]: CAC7A9E08B: to=<banane@madservers.cz>,
    relay=127.0.0.1[127.0.0.1], delay=135, status=sent (250 2.6.0 Ok,
    id=12814-02, from MTA: 250 Ok: queued as BF6509E08C)
postfix/qmgr[14731]: CAC7A9E08B: removed
postfix/smtpd[14861]: disconnect from localhost.localdomain[127.0.0.1]
postfix/virtual[14865]: BF6509E08C: to=<banane@madservers.cz>,
    relay=virtual, delay=0, status=sent (delivered to maildir)
postfix/qmgr[14731]: BF6509E08C: removed
postfix/smtpd[14852]: disconnect from localhost.localdomain[127.0.0.1]
```

Configuration de PHP et des modules adéquats

Nous avons installé PHP via aptitude tout à l'heure. Il faut maintenant configurer les modules nécessaires à PHP pour que le webmail ainsi que phpMyAdmin puissent avoir accès à l'interpréteur PHP et entre autres, aux modules de base de données MySQL.

Il faudra donc ajouter les lignes suivantes dans la section '[PHP]' des trois fichiers énoncés ci-dessous.

```
/etc/php4/apache/php.ini
/etc/php4/cli/php.ini
/etc/php4/cgi/php.ini
```

```
extension=mysql.so
extension=domxml.so
```

Une fois ces modifications effectuées, sauvegardez vos fichiers et redémarrez le service APACHE.

```
/etc/init.d/apache restart
```

Nous n'avons plus qu'à tester si PHP fonctionne bien avec APACHE. Pour ce faire, il nous suffit d'accéder en HTTP à l'adresse de la machine dans le répertoire de phpMyAdmin.

```
http://ipmachine/phpmyadmin/
```

Si vous arrivez à vous connecter à la base MySQL convenablement, c'est que tout fonctionne correctement.

Installation du framework HORDE

Le framework HORDE est disponible en libre téléchargement à partir de www.horde.org. Il est nécessaire à l'utilisation de notre webmail IMP, également produit de HORDE.

La première étape consiste dans le téléchargement du tarball adéquat. A l'heure de l'écriture de ce tutoriel, nous en sommes à la version 3.1.1. L'installation doit se faire dans le répertoire racine du serveur APACHE.

```
cd /usr/share
wget http://ftp.horde.org/pub/horde/horde-3.1.1.tar.gz
tar zxvf horde-3.1.1.tar.gz
mv horde-3.1.1 horde
```

Maintenant il faut renommer les fichiers templates en fichiers de configuration standard. Pour ceux qui veulent faire la configuration manuelle de HORDE via le panel d'administration, il faudra donner les droits d'écriture sur les fichiers de configuration. Ceux-ci seront à retirer par la suite, à la fin de l'installation et configuration.

```
cd horde/config
for f in *.dist; do cp $f `basename $f .dist`; done
chmod 777 /usr/share/horde/config/conf.php [ACCESSOIRE]
chown www-data:www-data -R /usr/share/horde
```

Il faut ajouter à la base MySQL les informations concernant HORDE. Le dump à insérer créé la base, les tables et l'utilisateur nommé 'horde' avec comme mot de passe 'horde'. Celui-ci sera à modifier par la suite.

```
mysql -u root -p < /usr/share/horde/scripts/sql/create.mysql.sql
```

Il faut désormais modifier les fichiers de configuration de HORDE. Nous avons effectué une configuration par défaut mais fonctionnelle pour faciliter la tâche.

```
/usr/share/horde/config/conf.php
```

```
<?php
$conf['debug_level'] = E_ERROR;
$conf['max_exec_time'] = 0;
$conf['use_ssl'] = 2;
$conf['server']['name'] = $_SERVER['SERVER_NAME'];
$conf['server']['port'] = $_SERVER['SERVER_PORT'];
$conf['compress_pages'] = true;
$conf['umask'] = 077;
$conf['session']['name'] = 'Horde';
$conf['session']['cache_limiter'] = 'nocache';
$conf['session']['timeout'] = 0;
$conf['cookie']['domain'] = $_SERVER['SERVER_NAME'];
$conf['cookie']['path'] = '/';
$conf['sql']['persistent'] = false;
$conf['sql']['hostspec'] = '127.0.0.1';
$conf['sql']['username'] = 'horde';
$conf['sql']['password'] = 'horde';
$conf['sql']['port'] = 3306;
$conf['sql']['protocol'] = 'tcp';
$conf['sql']['database'] = 'horde';
$conf['sql']['charset'] = 'iso-8859-1';
$conf['sql']['phptype'] = 'mysql';
$conf['auth']['admins'] = array('Administrator');
$conf['auth']['checkip'] = true;
$conf['auth']['checkbrowser'] = true;
$conf['auth']['alternate_login'] = false;
$conf['auth']['redirect_on_logout'] = false;
$conf['auth']['params']['username'] = 'Administrator';
$conf['auth']['params']['requestuser'] = false;
$conf['auth']['driver'] = 'imap';
$conf['signup']['allow'] = false;
$conf['log']['priority'] = PEAR_LOG_NOTICE;
$conf['log']['ident'] = 'HORDE';
$conf['log']['params'] = array();
$conf['log']['name'] = '/tmp/horde.log';
$conf['log']['params']['append'] = true;
$conf['log']['type'] = 'file';
$conf['log']['enabled'] = true;
$conf['log_accesskeys'] = false;
$conf['prefs']['driver'] = 'none';
$conf['datatree']['driver'] = 'null';
$conf['group']['driver'] = 'datatree';
$conf['cache']['default_lifetime'] = 1800;
$conf['cache']['params']['dir'] = Horde::getTempDir();
$conf['cache']['params']['gc'] = 86400;
$conf['cache']['driver'] = 'file';
$conf['token']['driver'] = 'none';
$conf['mailer']['params']['sendmail_path'] = '/usr/lib/sendmail';
$conf['mailer']['params']['sendmail_args'] = '-oi';
$conf['mailer']['type'] = 'sendmail';
```

```

$conf['vfs']['params']['vfsroot'] = '/tmp';
$conf['vfs']['type'] = 'file';
$conf['sessionhandler']['type'] = 'none';
$conf['geoip']['datafile'] = 'fr';
$conf['problems']['email'] = 'postmaster@madservers.cz';
$conf['problems']['maildomain'] = 'madservers.cz';
$conf['problems']['tickets'] = false;
$conf['menu']['apps'] = array('imp');
$conf['menu']['always'] = false;
$conf['menu']['links']['help'] = 'all';
$conf['menu']['links']['help_about'] = true;
$conf['menu']['links']['options'] = 'authenticated';
$conf['menu']['links']['problem'] = 'all';
$conf['menu']['links']['login'] = 'all';
$conf['menu']['links']['logout'] = 'authenticated';
$conf['hooks']['permsdenied'] = false;
$conf['hooks']['username'] = false;
$conf['hooks']['preauthenticate'] = false;
$conf['hooks']['postauthenticate'] = false;
$conf['hooks']['authldap'] = false;
$conf['portal']['fixed_blocks'] = array();
$conf['accounts']['driver'] = 'null';
$conf['imsp']['enabled'] = false;
$conf['kolab']['enabled'] = false;
?>

```

Modifions les informations suivantes pour que HORDE utilise IMP pour l'identification par défaut.

/usr/share/horde/config/prefs.php

```

$_prefs['initial_application'] = array(
    'value' => 'imp',
    'locked' => false,
    'shared' => true,
    'type' => 'select',
    'desc' => sprintf(_("What application should %s display after
login?"), $GLOBALS['registry']->get('name'))
);

```

Pour pouvoir accéder à horde, nous devons créer un Alias dans APACHE. Il vous suffit de créer le fichier suivant.

/etc/apache/conf.d/horde.conf

```
Alias /horde /usr/share/horde
```

Redémarrons APACHE pour prendre en compte les modifications.

```
/etc/init.d/apache restart
```

Configuration d'APACHE pour le virtual host

Dans notre cas nous supposons que le webmail sera accédé depuis 'http://webmail.madservers.cz'. Il faut pour cela créer un VirtualHost dans APACHE.

C'est très simple. Créons le fichier suivant.

```
/etc/apache/conf.d/imp.conf
```

```
<VirtualHost *:80>
  ServerAdmin info@madservers.cz
  ServerName webmail.madservers.cz

  DocumentRoot /usr/share/horde/imp

  ErrorLog /var/log/apache/webmail-error_log
  CustomLog /var/log/apache/webmail-access_log combined

  <Directory "/usr/share/horde/imp">
    Options Indexes FollowSymLinks
    AllowOverride All
    Order allow,deny
    Allow from all
  </Directory>
</VirtualHost>
```

Redémarrons APACHE pour prendre en compte les modifications.

```
/etc/init.d/apache restart
```

Installation et configuration d'IMP

Tout comme HORDE, IMP est librement distribué sur le site www.horde.org. A l'heure de l'écriture de ce tutoriel, nous en sommes à la version 3.4.1.

Les étapes d'installation et de configuration sont similaires à celles de HORDE, indiquées dans la partie précédente.

```
cd /usr/share/horde
wget ftp://ftp.horde.org/pub/imp/imp-h3-4.1.tar.gz
tar zxvf imp-h3-4.1.tar.gz
mv imp-h3-4.1 imp
```

Comme précédemment, nous renommons les fichiers templates en fichiers de configuration standard. Comme expliqué pour HORDE, si vous désirez personnaliser votre configuration IMP, n'oubliez pas de mettre les droits en écriture sur ces fichiers.

```
cd imp/config
for foo in *.dist; do cp $foo `basename $foo .dist`; done
```

Retrouvez ci-dessous le fichier de configuration généré par nos soins.

/usr/share/horde/imp/config/conf.php

```
<?php
$conf['utils']['gnupg_keyserver'] = array('wwwkeys.pgp.net');
$conf['utils']['gnupg_timeout'] = '10';
$conf['menu']['apps'] = array();
$conf['user']['allow_view_source'] = true;
$conf['user']['allow_resume_all'] = false;
$conf['user']['allow_resume_all_in_drafts'] = false;
$conf['user']['select_sentmail_folder'] = false;
$conf['user']['allow_folders'] = true;
$conf['user']['alternate_login'] = false;
$conf['user']['redirect_on_logout'] = false;
$conf['server']['server_list'] = 'hidden';
$conf['server']['sort_limit'] = '0';
$conf['server']['cache_folders'] = false;
$conf['server']['cache_msgbody'] = false;
$conf['mailbox']['show_attachments'] = false;
$conf['mailbox']['show_preview'] = false;
$conf['mailbox']['show_xpriority'] = false;
$conf['fetchmail']['show_account_colors'] = false;
$conf['fetchmail']['size_limit'] = '4000000';
$conf['msgsettings']['filtering']['words'] = './config/filter.txt';
$conf['msgsettings']['filtering']['replacement'] = '****';
$conf['spam']['reporting'] = false;
$conf['notspam']['reporting'] = false;
$conf['msg']['prepend_header'] = true;
$conf['msg']['append_trailer'] = true;
$conf['compose']['allow_cc'] = true;
$conf['compose']['allow_bcc'] = true;
$conf['compose']['allow_receipts'] = true;
$conf['compose']['special_characters'] = true;
$conf['compose']['use_vfs'] = false;
$conf['compose']['link_all_attachments'] = false;
$conf['compose']['link_attachments_notify'] = true;
$conf['compose']['link_attachments'] = true;
$conf['compose']['add_maildomain_to_unexpandable'] = false;
$conf['compose']['attach_size_limit'] = '0';
$conf['compose']['attach_count_limit'] = '0';
$conf['hooks']['vinfo'] = false;
$conf['hooks']['signature'] = false;
$conf['hooks']['trailer'] = false;
$conf['hooks']['fetchmail_filter'] = false;
$conf['hooks']['mbox_redirect'] = false;
$conf['hooks']['mbox_icon'] = false;
$conf['hooks']['spam_bounce'] = false;
$conf['maillog']['use_maillog'] = true;
$conf['tasklist']['use_tasklist'] = true;
$conf['notepad']['use_notepad'] = true;
?>
```

Pour finir, il ne nous reste plus qu'à configurer les serveurs utilisés par IMP pour se connecter au IMAP. Tout ceci se fait dans le fichier 'conf.php' d'IMP. Retrouvez ci-dessous notre fichier de configuration, à modifier selon vos besoins, évidemment.

/var/www/horde/imp/config/servers.php

```
<?
$servers['imap'] = array(
    'name' => 'IMAP Server',
    'server' => 'mail.madservers.cz',
    'hordeauth' => false,
    'protocol' => 'imap/ssl/novalidate-cert',
    'port' => '993',
    'maildomain' => 'madservers.cz',
    'smtpport' => 'mail.madservers.cz',
    'smtpport' => 25,
    'realm' => '',
    'preferred' => '1',
);
?>
```

Vous pouvez maintenant tenter une connexion au webmail pour voir si vous avez reçu vos emails de test.

<http://webmail.madservers.cz/>

Si cela fonctionne, félicitations, le webmail est installé avec succès. Sinon, reportez vous à la page de début d'IMP à l'url suivante.

<http://webmail.madservers.cz/test.php>

FIN (13/04/2006)