OUDAR Nicolas SIO2 2023/2024

Mise en place d'un service mail interne

Préambule :

Le système mis en place devra fonctionner sous Linux. C'est un travail individuel. Démonstration des solutions fonctionnelles avant le 27/11/2023 à 16h. Remise des comptes rendus (envoi sur monge.vf@gmail.com) avant le 01/12/2023.

L'internet sera simulé par le réseau de la salle, ce dernier sera également la zone DNS .monge.

Contexte :

Le groupe lyonnais Lise Charmel a été victime d'un ransomware le 8 novembre 2019. La société a refusé de payer et son système informatique complet est resté bloqué pendant près d'un mois, ralentissant considérablement l'activité du groupe et de ses 1 150 employés au niveau mondial.

Le manque à gagner est chiffré par ses dirigeants à plusieurs millions d'euros.

Le vers qui a intégralement crypté les données de l'ensemble des machines du réseau, serveurs de sauvegarde compris, est arrivé via un mail reçu par le service commercial.

Pour éviter de se trouver à nouveau dans une situation semblable, plusieurs mesures ont été mises en œuvre. La plus radicale, pour laquelle vous êtes recruté, est la mise en place d'un service mail interne à l'entreprise. Les machines reliées à l'Internet seront uniquement celles nécessaires à l'activité de l'entreprise et le seront via des sandbox. L'ensemble des communications internes se fera par ce nouveau service de courrier interne.

Demande du client :

L'entreprise possède un nom de domaine pour chaque lieu de production. Tous dépendent de la zone .monge. Il s'agit de mettre en place sur le LAN de chaque lieu de production un serveur mail afin de pouvoir communiquer avec les autres sites de l'entreprise.

Travail réaliser :

1. Installation des paquets
2. Enregistrement DNS
3. Utilisateur vmail
4. Préparer les certificats
5. Configuration de Postfixadmin
6. Configuration de Postfix
7. Configuration de Dovecot
8. Teste de la configuration
9. Partie explicative
10. Conclusion

---

1. Installation des paquets

Pour commencer on fait des mises à jour sur la VM (je ne montre pas comment j'ai fait la VM car je l'ai déjà montré dans d'autres compte rendus puis j'ai utilisé la même VM que j'ai utilisé pour faire le serveur DNS) :

Apt update

Apt upgrade

Puis on installer tout ce dont on a besoin :

Apt install php8.2-mysql php8.2-bstring php8.2-imap php8.2-xml php8.2-curl php8.2 -y

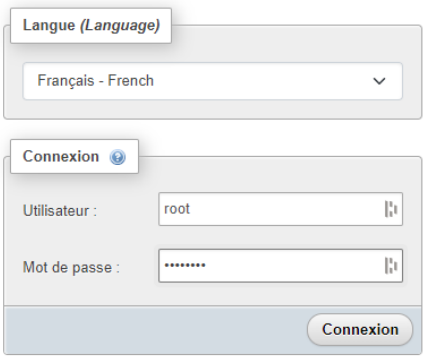
A2enmod php8.2

/etc/init.d/apache2 force-reload

Cd /var/www/html

Wget https://files.phpmyadmin.net/phpMyAdmin/5.2.1/phpMyAdmin-5.2.1-all-languages.zip

On teste si phpMyAdmin fonctionne :



Apt install unzip

Unzip phpMyAdmin-5.2.1-all-languages.zip

Cp -r /var/www/html/phpMyAdmin-5.2.1-all-languages /var/www/html/pma

rm -r phpMyAdmin-5.2.1-all-languages

rm phpMyAdmin-5.2.1-all-languages.zip

apt install postfixadmin

apt-get install postfix postfix-mysql libsasl2-modules sasl2-bin

apt-get install dovecot-mysql dovecot-pop3d dovecot-imapd dovecot-managesieved

---

2. Enregistrement DNS

Ensuite on configure le fichier du DNS qu'on avait créer auparavant sur le tp DNS :

```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns.lunar.monge. root.lunar.monge. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns.lunar.monge.
NS      IN      A       172.30.5.50
@       IN      A       172.30.5.50
@       IN      MX      10 mail.lunar.monge.
mail    IN      A       172.30.5.50
```

---

3. Utilisateur vmail

Pour la suite on crée un groupe et un utilisateur pour recevoir les mails :

groupadd -g 5000 vmail

useradd -g vmail -u 5000 vmail -d /var/vmail -m

---

4. Préparer les certificats

Puis on crée un certificat pour le mail soit bien envoyé :

openssl req -new -x509 -days 3650 -nodes -newkey rsa:4096 -out /etc/ssl/certs/mailserver.pem -keyout /etc/ssl/private/mailserver.pem

Répondez aux questions qui sont posées. Si vous ne voulez rien mettre dans le champ, mettez un point ".".
Dans le champ Organization Name, mettez votre nom exemple : lunar.
Dans le champ Common Name, mettez votre nom de domaine exemple : lunar.monge.

Votre certificat est généré et placé dans /etc/ssl/certs/mailserver.pem.

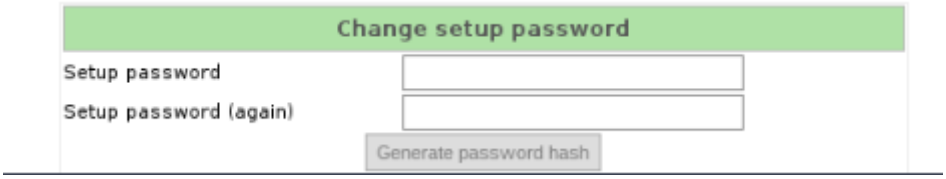
---

5. Configuration de Postfixadmin

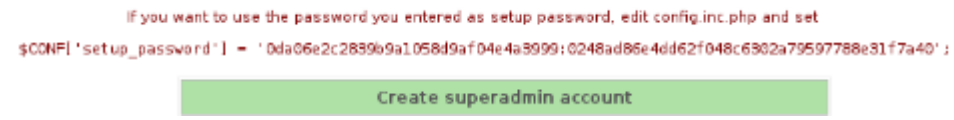
Pour commencer la configuration de Postfixadmin on va sur ce lien (mettez soit votre DNS soit votre adresse IP) :

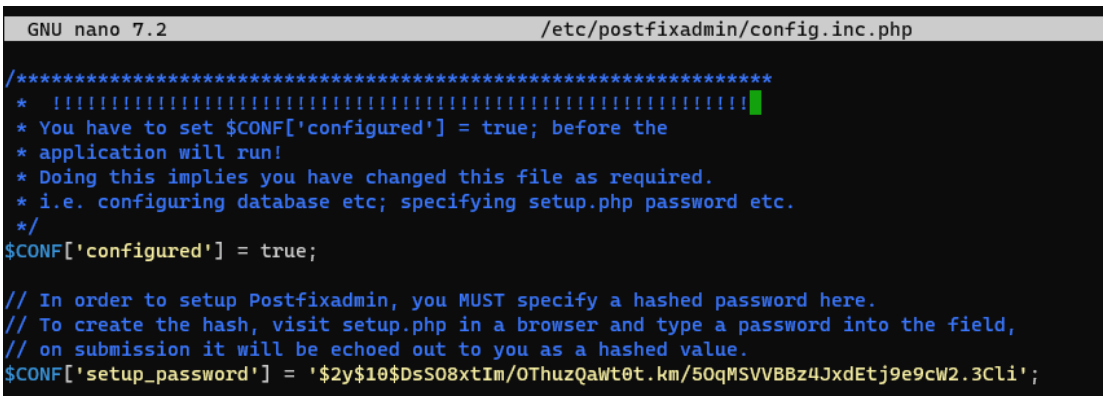
http://lunar.monge/postfixadmin/setup.php

Puis on crée un mot de passe :



Ensuite on ajoute le mot de passe crypter dans le dossier config.inc.php et on crée un compte admin :





Ensuite on se connecte avec le compte admin :

Puis on crée un compte mail (test pour la part) :

---

6. Configuration de Postfix

Pour commencer la configuration de Postfix on configure le fichier mysql-virtual-mailbox-domains :

```
  GNU nano 7.2                          /etc/postfix/mysql-virtual-mailbox-domains.cf
user = root
password = rootroot
hosts = 127.0.0.1
dbname = postfixadmin
query = SELECT 1 FROM domain WHERE domain='%s'
```

Et on exécute ces commandes :

postconf -e virtual_mailbox_domains=mysql:/etc/postfix/mysql-virtual-mailbox-domains.cf

postmap -q example.org mysql:/etc/postfix/mysql-virtual-mailbox-domains.cf

Si le domaine example.org est présent dans vos domaines, la valeur "1" sera affichée, sinon rien.

```
  GNU nano 7.2                          /etc/postfix/mysql-virtual-mailbox-maps.cf
user = root
password = rootroot
hosts = 127.0.0.1
dbname = postfixadmin
query = SELECT 1 FROM mailbox WHERE username='%s'
```

Et on exécute ces commandes :

postconf -e virtual_mailbox_maps=mysql:/etc/postfix/mysql-virtual-mailbox-maps.cf

postmap -q john@example.org mysql:/etc/postfix/mysql-virtual-mailbox-maps.cf

```
  GNU nano 7.2                           /etc/postfix/mysql-virtual-alias-maps.cf
user = root
password = rootroot
hosts = 127.0.0.1
dbname = postfixadmin
query = SELECT goto FROM alias WHERE address='%s'
```

Et on exécute ces commandes :

postconf -e virtual_alias_maps=mysql:/etc/postfix/mysql-virtual-alias-maps.cf

postmap -q abuse@example.org mysql:/etc/postfix/mysql-virtual-alias-maps.cf

chgrp postfix /etc/postfix/mysql-*.cf

Puis on change le port qu'utilise le serveur mail :

```
  GNU nano 7.2                                        /etc/postfix/master.cf
submission inet n       -       -       -       -       smtpd
```

```
 GNU nano 7.2                                    /etc/postfix/main.cf
virtual_mailbox_maps = mysql:/etc/postfix/mysql-virtual-mailbox-maps.cf
virtual_alias_maps = mysql:/etc/postfix/mysql-virtual-alias-maps.cf
virtual_transport = dovecot
dovecot_destination_recipient_limit = 1

smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_auth_enable = yes
smtpd_tls_security_level = may
smtpd_tls_auth_only = yes
smtpd_recipient_restrictions = permit_mynetworks permit_sasl_authenticated reject_unauth_destination
```

---

7. Configuration de Dovecot

Pour commencer la configuration de Dovecot on configure le fichier 10-auth.conf :

```
 GNU nano 7.2                                      10-auth.conf
## Password and user databases
##


#
# Password database is used to verify user's password (and nothing more).
# You can have multiple passdbs and userdbs. This is useful if you want to
# allow both system users (/etc/passwd) and virtual users to login without
# duplicating the system users into virtual database.
#
# <doc/wiki/PasswordDatabase.txt>
#
# User database specifies where mails are located and what user/group IDs
# own them. For single-UID configuration use "static" userdb.
#
# <doc/wiki/UserDatabase.txt>

#!include auth-deny.conf.ext
#!include auth-master.conf.ext

#!include auth-system.conf.ext
!include auth-sql.conf.ext
#!include auth-ldap.conf.ext
#!include auth-passwdfile.conf.ext
#!include auth-checkpassword.conf.ext
#!include auth-static.conf.ext
```

Puis on configure le fichier auth-sql.conf.ext :

```
  GNU nano 7.2                                          auth-sql.conf.ext
}

# "prefetch" user database means that the passdb already provided the
# needed information and there's no need to do a separate userdb lookup.
# <doc/wiki/UserDatabase.Prefetch.txt>
#userdb {
#  driver = prefetch
#}

#userdb {
#  driver = sql
#  args = /etc/dovecot/dovecot-sql.conf.ext
#}

# If you don't have any user-specific settings, you can avoid the user_query
# by using userdb static instead of userdb sql, for example:
# <doc/wiki/UserDatabase.Static.txt>
#userdb {
  #driver = static
  #args = uid=vmail gid=vmail home=/var/vmail/%u
#}
userdb {
  driver = static
  args = uid=vmail gid=vmail home=/var/vmail/%d/%n
}
```

Puis le fichier 10-mail .conf :

```
  GNU nano 7.2                                                  10-mail.conf
#
mail_location = maildir:/var/vmail/%d/%n/Maildir
```

Ensuite le fichier 10-master.conf :

```
  GNU nano 7.2                                                10-master.conf *
  unix_listener /var/spool/postfix/private/auth {
    mode = 0660
    user = postfix
    group = postfix
  }
```

Pour la suite le fichier 14-lda.conf :

```
  GNU nano 7.2                                                  15-lda.conf *

protocol lda {
  # Space separated list of plugins to load (default is global mail_plugins).
  mail_plugins = $mail_plugins sieve
}
```

Enfin le fichier dovecot-sql.conf.ext :

```
  GNU nano 7.2                                      /etc/dovecot/dovecot-sql.conf.ext *
driver = mysql
connect = host=127.0.0.1 dbname=postfixadmin user=root password=rootroot
password_query = SELECT username,domain,password FROM mailbox WHERE username='%u';
```

Et pour finir on exécute ces commandes :

chown root:root /etc/dovecot/dovecot-sql.conf.ext

chmod go= /etc/dovecot/dovecot-sql.conf.ext

chgrp vmail /etc/dovecot/dovecot.conf

chmod g+r /etc/dovecot/dovecot.conf

service dovecot restart

```
dovecot   unix  -       n       n       -       -       pipe
  flags=DRhu user=vmail:vmail argv=/usr/lib/dovecot/dovecot-lda -f ${sender} -d ${recipient}
```

service postfix restart

postfix/master[…]: daemon started -- version 2.9.6, configuration /etc/postfix

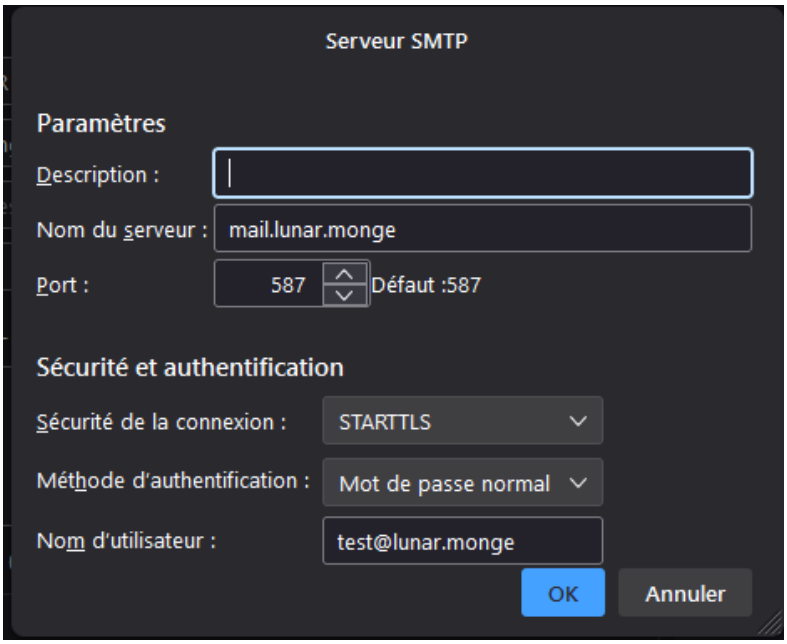
postconf -e virtual_transport=dovecot

postconf -e dovecot_destination_recipient_limit=1

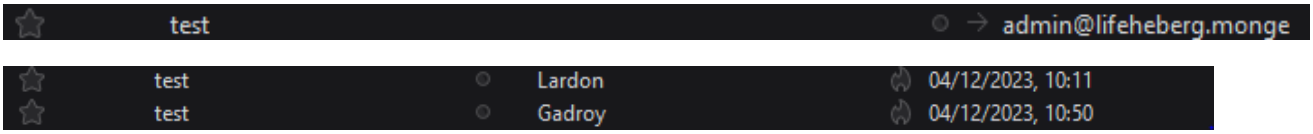
---

8. Teste de la configuration

Pour faire les tests j'utilise thunderbird un logiciel de service mail et on se connecte avec le mail test :



Puis on peut voir que je reçois et je peux envoyer des mails :



---

9. Partie explicative

La commande **apt install php8.2-mysql php8.2-bstring php8.2-imap php8.2-xml php8.2-curl php8.2 -y** permet d'installer les modules PHP nécessaires à la gestion des bases de données MySQL, des chaînes binaires, des comptes de messagerie IMAP, des données XML et des requêtes HTTP sur un système Debian ou Ubuntu. L'option -y évite de répondre manuellement à chaque invite, ce qui accélère l'installation.

j'ai choisie de télécharger phpmyadmin comme ceci car il y a moins de bug ou de problème et je suis sûr que cela fonctionne :

Wget https://files.phpmyadmin.net/phpMyAdmin/5.2.1/phpMyAdmin-5.2.1-all-languages.zip

J'ai fait cette commande car cela me permet de connecter à phpmyadmin en mettant pma sur mon navigateur pour cela soit plus rapide et plus intuitif (http://lunar.monge/pma) : **Cp -r /var/www/html/phpMyAdmin-5.2.1-all-languages /var/www/html/pma**

La commande **apt-get install postfix postfix-mysql libsasl2-modules sasl2-bin** permet d'installer les paquets nécessaires à la configuration et à la sécurisation d'un serveur de messagerie Postfix sur un système Debian ou Ubuntu. Elle permet d'utiliser MySQL comme base de données pour les informations d'utilisateur et d'activer l'authentification SASL pour protéger les connexions de messagerie.

Et j'ai choisi postfix car c'est un des outils pour faire des serveurs mails le plus utilisé est plus simple à mettre en place.

La commande **groupadd vmail** crée un groupe pour les comptes de messagerie virtuels. Le GID du groupe est 5000 par défaut.

La commande **useradd -g vmail -u 5000** crée un utilisateur vmail appartenant au groupe vmail avec le UID 5000.

La commande **openssl req -new -x509 -days 3650 -nodes -newkey rsa:4096 -out /etc/ssl/certs/mailserver.pem -keyout /etc/ssl/private/mailserver.pem** permet de générer un certificat TLS/SSL pour sécuriser un serveur de messagerie. Le certificat est auto-signé et valable 10 ans. La clé privée est générée sans phrase de passe.

```
  GNU nano 7.2                          /etc/postfix/mysql-virtual-mailbox-domains.cf
user = root
password = rootroot
hosts = 127.0.0.1
dbname = postfixadmin
query = SELECT 1 FROM domain WHERE domain='%s'
```

Le fichier **/etc/postfix/mysql-virtual-mailbox-domains.cf** définit les paramètres pour la connexion de Postfix à la base de données MySQL. Dans l'image, les paramètres sont définis comme suit :

user: root

password: rootroot

hosts: 127.0.0.1

dbname: postfixadmin

query: SELECT 1 FROM domain WHERE domain='%s'

La requête **SQL SELECT 1 FROM domain WHERE domain='%s'**recherche le domaine correspondant à l'adresse e-mail spécifiée dans la variable %s. Si le domaine existe, la requête renvoie 1. Si le domaine n'existe pas, la requête renvoie 0.

Cette configuration permet à Postfix de rechercher les informations d'utilisateur dans la base de données MySQL.

La commande **postconf -e virtual_mailbox_domains=mysql:/etc/postfix/mysql-virtual-mailbox-domains.cf** active l'utilisation de la base de données MySQL pour stocker les informations sur les domaines virtuels. Cela permet de centraliser les données et de simplifier la gestion des domaines et des utilisateurs virtuels.

La commande **postmap -q example.org mysql:/etc/postfix/mysql-virtual-mailbox-domains.cf** vérifie l'existence du domaine example.org dans la base de données MySQL utilisée par Postfix pour les domaines virtuels. Si le domaine existe, la commande renvoie 1, sinon 0.

```
  GNU nano 7.2                            /etc/postfix/mysql-virtual-alias-maps.cf
user = root
password = rootroot
hosts = 127.0.0.1
dbname = postfixadmin
query = SELECT goto FROM alias WHERE address='%s'
```

Réponse courte

L'image montre la configuration de Postfix pour utiliser une base de données MySQL pour stocker les informations sur les comptes de messagerie virtuels. Les paramètres de configuration sont les suivants :

Nom d'utilisateur : root

Mot de passe : rootroot

Adresse IP du serveur MySQL : 127.0.0.1

Nom de la base de données MySQL : postfixadmin

Requête SQL : SELECT 1 FROM mailbox WHERE username='%s'

La première commande, **postconf -e virtual_alias_maps=mysql:/etc/postfix/mysql-virtual-alias-maps.cf**, permet à Postfix d'utiliser une base de données MySQL pour stocker les informations relatives aux alias. Le fichier de configuration de la base de données est /etc/postfix/mysql-virtual-alias-maps.cf. Cette configuration centralise les données d'alias, ce qui facilite la gestion des alias et des utilisateurs virtuels.

La deuxième commande, **postmap -q abuse@example.org mysql:/etc/postfix/mysql-virtual-alias-maps.cf**, permet de vérifier l'existence de l'alias abuse@example.org dans la base de données MySQL. Si l'alias existe, la commande renvoie la valeur 1, sinon 0.

La troisième commande, **chgrp postfix /etc/postfix/mysql-*.cf**, permet de modifier le propriétaire des fichiers /etc/postfix/mysql-virtual-alias-maps.cf au groupe postfix. Cela garantit que seul l'utilisateur postfix peut modifier la configuration des alias.

En résumé, ces trois commandes permettent de configurer Postfix pour utiliser une base de données MySQL pour stocker les informations d'alias et de gérer les alias de manière centralisée. Elles assurent également la sécurité du système en limitant le droit de modification des fichiers de configuration aux utilisateurs autorisés.

Ceci permet d'utiliser le port 587 car le port 25 (port par défaut du smtp) est très souvent bloquer par défaut sur les routeurs ou par les FAI :

Commande 1 : **chown root:root /etc/dovecot/dovecot-sql.conf.ext**

Cette commande permet de changer le propriétaire du fichier /etc/dovecot/dovecot-sql.conf.ext à l'utilisateur root. Cela signifie que seul l'utilisateur root pourra modifier ce fichier.

Commande 2 **: chmod go= /etc/dovecot/dovecot-sql.conf.ext**

Cette commande permet de changer les permissions du fichier /etc/dovecot/dovecot-sql.conf.ext afin que seuls les utilisateurs root et other puissent le lire.

Commande 3 : **chgrp vmail /etc/dovecot/dovecot.conf**

Cette commande permet de changer le groupe propriétaire du fichier /etc/dovecot/dovecot.conf au groupe vmail. Cela signifie que tous les utilisateurs appartenant au groupe vmail pourront lire ce fichier.

Commande 4 : **chmod g+r /etc/dovecot/dovecot.conf**

Cette commande permet de donner au groupe vmail le droit de lire le fichier /etc/dovecot/dovecot.conf. Cela permet aux utilisateurs appartenant au groupe vmail de lire la configuration de Dovecot, mais pas de la modifier.

En résumé

Ces quatre commandes permettent de protéger la configuration de Dovecot et de limiter l'accès aux utilisateurs non autorisés.

La première commande : **postfix/master[…]: daemon started -- version 2.9.6, configuration /etc/postfix**

Cette commande indique que le serveur de messagerie Postfix a été démarré et qu'il utilise la configuration définie dans le fichier /etc/postfix/main.cf. La version de Postfix installée est 2.9.6.

La deuxième commande : **postconf -e virtual_transport=dovecot**

Cette commande configure Postfix pour utiliser Dovecot comme transporteur virtuel. Cela signifie que Postfix utilisera Dovecot pour gérer les boîtes aux lettres virtuelles et la messagerie sortante.

La troisième commande : **postconf -e dovecot_destination_recipient_limit=1**

Cette commande limite le nombre de destinataires que Postfix peut transmettre à Dovecot à 1. Cela permet de limiter le nombre de courriels qui peuvent être envoyés à la fois et de protéger le serveur Dovecot contre les attaques par déni de service.

En résumé

Ces trois commandes permettent de configurer Postfix pour utiliser Dovecot comme transporteur virtuel et de limiter le nombre de courriels envoyés à Dovecot à la fois. Cela améliore la sécurité et la performance du serveur de messagerie.

---

10. Conclusion

Pour conclure durant ce TP j’ai appris à utiliser Postfix et Dovecot qui m’a permit de créer un serveur mail (SMTP) Postfix est un service très simple à mettre en place et très efficace.