# Configuration de `postfix`, `imap`

Ce TP offre différentes politiques de sécurité ; vous pourrez en privilégier une pour la rédaction du compte-rendu final. Il offre également différentes options d'accès selon que vous serez positionné côté LAN ou WAN du routeur `pfSense` pour l'accès aux services. Votre politique de sécurité pourra aussi évoluer en fonction des prochains TP.

## 1 Postfix 2

Sur `lxle`, installez `postfix` en choisissant l'option "Site Internet" puis `mailutils`. Les 2 fichiers utiles pour configurer `postfix` sont (il n'y a que des modifications mineures à y faire) :

- `/etc/postfix/main.cf`
- `/etc/postfix/master.cf`

Vérifiez tout d'abord que `main.cf` instancie correctement les variables `mydomain`, `myhostname`, `mynetworks` ainsi que `alias_maps` et `alias_database` (qui doivent être positionnés).

```
myhostname = lxle.cs.sr
mydomain = cs.sr
mynetworks = 127.0.0.0/8
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
```

Pour faciliter la correction des erreurs, on passe le démon en mode debug dans le fichier `master.cf` :

```
# ==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
# ==========================================================================
smtp      inet  n       -       n       -       -       smtpd -v
```

Normalement `postfix` peut être (re)démarré avec succès pour la distribution locale. Essayez d'envoyer du mail par une connexion `telnet 25` (voir l'Annexe pour la syntaxe). Vérifiez ensuite par une commande `mail` (installée avec `mailutils`) sur un utilisateur, par exemple Alice (créée par un `adduser`). Vérifiez dans `/var/log/mail.log` que le mail a bien été traité.
Il faut ensuite accepter les connexions depuis le réseau local en ajoutant à `main.cf`,

```
inet_interfaces = all
mynetworks = 127.0.0.0/8 <add your virtual network address/mask>
mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
```

Pour activer le mail sur votre interface réseau, il faut faire un `stop/start` du service `postfix`. Essayez d'envoyer du mail (en local sur `cs.sr`). Essayez successivement comme destinataires un utilisateur légitime (p.e. Alice) et un utilisateur indéfini. Vérifiez l'existence de ces utilisateurs par la commande `VRFY` en `telnet 25`. Créez ensuite un alias d'Alice pour que les mails adressés `Alice.Elle` soient délivrés à Alice. Pensez bien à mettre à jour `/etc/aliases` suivi de `postalias /etc/aliases`), sans oublier d'autoriser la connexion au service sur le firewall.

### 1.1 Mail sous kali

Testez d'abord l'envoi de mail sur la `lxle` en `telnet 25`. Vous configurerez `kali` pour qu'elle utilise `lxle` comme relai : installez et configurez `ssmtp` en éditant le fichier `/etc/ssmtp/ssmtp.conf` et en mettant juste à jour la variable `mailhub` (il n'y a aucun service à redémarrer).

### 1.2 Optionnel

En vous inspirant du tutoriel, vous pourrez ajouter ultérieurement une authentification de l'utilisateur pour l'envoi de mails.

## 2 Ajout d'un serveur `imap(s)`

On ajoute `imap(s)` sur `lxle` pour avoir un accès distant à la boîte aux lettres. Vous pourrez vous inspirer du tutoriel. `Postfix` a été configuré pour utiliser `mbox` comme type de boîte à lettres. `Dovecot` est automatiquement configuré pour fonctionner en `imaps`. Pensez à mettre à jour votre firewall.

(1) installez `dovecot-core` et `dovecot-imapd` si ce n'est pas déjà fait ;

(2) Générez un certificat auto-signé et une clé en conformité avec le contenu du fichier `10-ssl.conf`.

(3) Configurez le client mail `thunderbird` sur la `lxle` ou la `kali`.

Notez que le port actuel du service `imap(s)` est le 143 (sécurisé par encapsulation dans une trame `SSL`).

## 3 Accès depuis le WAN

Modifiez la configuration de `pfSense` afin d'accéder au serveur `smtp` et/ou `imap` depuis le côté WAN du routeur en ajoutant des règles au NAT (comme pour l'accès au serveur apache) et vérifier que le firewall de `pfSense` accepte bien les connexions. Vous pourrez également réfléchir s'il est prudent de laisser un accès au serveur `smtp`.
En supposant qu'Alice se connecte depuis une machine distante, votre machine physique (côté WAN), sauriez-vous retrouver son mot de passe si elle se connecte par `imap` ou par `imaps` ?

## 4 Sécuriser `imap` par transfert de port

L'énoncé proposé ici est restreint à l'accès de la `lxle` du côté LAN de `pfSense`. Il peut cependant être adapté à un accès côté WAN au prix de quelques efforts.

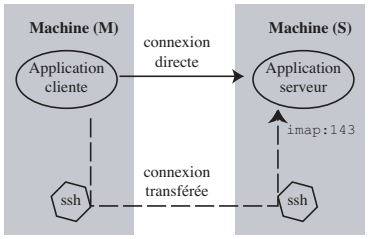


FIGURE 1 – Transfert de port par ssh

Nous supposons ici que vous contactez le service `imap` de (S) depuis (M) et que vous avez un accès `ssh` sur (S). Notre but est de sécuriser l'accès à `imap` au moyen de `ssh` en « tunnelisant » la connexion `imap`.

(1) Après avoir installé et activé `openssh-server` sur `lxle` (S), configurez le client de (M) pour contacter `imap`.

(2) Effectuez un transfert de port par `ssh`. Le service `imap` qui s'exécute sur (S) attend les connexions sur le port 143. Pour « tunneliser » cette connexion, il faut choisir un numéro de port local sur (M) –entre 1024 et 65535– et le transférer sur la socket distante (S,143). Par exemple, avec le port 2023, on crée le tunnel de (M) vers (S) par : `(M): ssh -L2023:localhost:143 S.` L'option `-L` indique un transfert de port local dans lequel le client TCP est sur la machine locale avec le client `ssh`. On spécifie ensuite la valeur du port local `2023`, le numéro du port distant `143` et le nom de la machine distante `S` ou son adresse IP. La commande précédente vous connecte sur (S) en transférant le port `localhost.2023` en `S.143`. Vérifiez par `ssh localhost -p 2023`.

(3) Configurez le client de (M) pour l'accès à `imap` de (S) après transfert de port.

(4) Vérifiez la validité de ce travail au moyen d'`ettercap`, en adaptant le cas échéant le firewall.

# 5 Utilisation de GnuPG (facultatif)

Pour toutes les questions suivantes, reportez-vous à la documentation de `gnupg`. Pensez à utiliser l'option `--armor` de `gpg` pour éviter les problèmes de codage de caractères.

(1) Créez une paire de clés ainsi qu'un cetificat de révocation pour Alice puis pour Bob en prenant soin de les rendre transmissibles.

(2) Alice transmet sa clé publique à Bob par courriel.

(3) Bob importe la clé reçue d'Alice.

(4) Bob vérifie son trousseau.

(5) Bob, qui connait la clé d'Alice, rédige un message, le chiffre puis lui transmet.

(6) Alice déchiffre le message qu'elle a reçu.

(7) Alice lui répond par un message signé mais non chiffré (pour plus de lisibilité, utilisez l'option `--clearsign` pour éviter la compression du message).

(8) Bob vérifie l'intégrité de ce message et l'authenticité de la signature. Cependant, comme la signature d'Alice n'est pas certifiée, il obtient un avertissement.

(9) Pour mieux valider la signature d'Alice, Bob décide de la signer pour la certifier (option `--edit-key`).

(10) Il vérifie à nouveau l'intégrité et l'authenticité de ce message et il ne devrait plus avoir d'erreur.

(11) Il renvoie à Alice sa clé publique certifiée pour qu'elle puisse la transmettre à d'autres utilisateurs.

(12) Alice envoie ensuite à Bob un message chiffré et signé.

(13) Bob le lit et le vérifie.

(14) Comment Charles, un autre utilisateur, peut entrer en lice et communiquer avec Alice et Bob ? Quelle confiance accorde-t-il à la clé d'Alice ? de Bob ?

Evidemment, ailleurs que dans le domaine `cs.sr`, il existe des serveurs de clé qui contiennent les clés publiques des utilisateurs, p.e. `http://wwwkeys.eu.pgp.net/`. On peut à la fois y enregistrer des clés et l'interroger pour retrouver une clé.

# A Le protocole SMTP

On rappelle ci-dessous les principales commandes du protocole `SMTP` :

**HELLO (EHLO site émetteur)**
Initie une session par une identification des 2 parties ; le récepteur s'identifie dans sa réponse.
**Mail (MAIL FROM : expéditeur)**
Débute une nouvelle transaction. Spécifie l'expéditeur pour un éventuel message d'erreur ;
**Recipient (RCPT TO : destinataire)**
Spécifie un destinataire. Cette procédure peut être répétée autant que nécessaire.
Si le destinataire n'est pas valide, un code d'erreur est renvoyé.
**Data (DATA)**
Envoie le message (en-tête et corps) terminé par une ligne ne contenant qu'un point.
**Reset (RSET)**
Annule la transaction en cours et réinitialise le logiciel. Un nouveau message peut être renvoyé.
**Verify (VRFY destinataire)**
Demande au site récepteur de vérifier la validité de l'adresse du destinataire.
**EXPAND (EXPN liste)**
Demande au site récepteur de vérifier la validité de l'adresse de la liste de diffusion
et de fournir les adresses de la liste.
**Quit (QUIT)**
Termine la session SMTP.