Leçon 12 : Le service mail

12.1 Introduction

Le service mail existe depuis les débuts d'internet. C'est probablement un des premiers services qui a vu le jour. Bien qu'il fût, par moment et encore régulièrement, décrié, il est toujours indispensable aujourd'hui.

Il est indispensable aux entreprises, premier moyen de communication, il souffre également de son plus grand démon : le spam.

Bien que nous allons expliquer le fonctionnement et la configuration du service mail ci-après, il faut bien être conscient qu'expliquer toutes les techniques possibles et déployées pour combattre le spam pourrait faire l'objet d'un cours séparé complet. J'ai donc choisi une présentation plus simple et plus claire, en listant à la fin de celle-ci les techniques indispensables qu'il faut déployer pour protéger son serveur mail.

12.2 Le mail interne vs externe

Beaucoup d'entreprises séparent le mail interne du mail externe. Le mail interne est davantage utilisé pour le fonctionnement de l'entreprise, il doit être entouré d'outils comme les agendas, les carnets d'adresse partagés, … alors que le mail externe est plutôt utilisé pour la communication avec les clients et les fournisseurs.

Ces deux objectifs séparés ont très souvent conduit à séparer les serveurs mails utilisés. Ainsi, pour le mail interne, on trouve davantage des *serveurs collaboratifs* comme *Microsoft Exchange Server* (dans le monde Microsoft), *Zimbra* (dans le monde Unix), ou bien d'autres plateformes (OBM, SOgo, Zarafa, …) intégrant bon nombre d'outils à côté du courrier électronique. A l'inverse, le mail externe est presque exclusivement dévolu à des serveurs Unix et des logiciels comme *sendmail*, *postfix*, *qmail*, *exim*, …

Certains programmes comme *qmail* sont utilisés par de grands noms comme Yahoo pour le service Yahoo Mail. D'autres comme *sendmail* sont présents sur tous les systèmes UNIX. C'est ce programme serveur que nous allons découvrir dans cette leçon car il est un des plus anciens et sa présence le rend, probablement, incontournable.

12.3 Rappel sur le fonctionnement du mail

Pour rappel, le service mail utilise le protocole SMTP. Ce protocole est associé au port TCP 25 et il sert à l'envoi d'un mail. La réception d'un mail est réalisée par le protocole POP3 (port TCP 110) ou IMAP (port TCP 143).

Pour des raisons de sécurité, seul le serveur configuré pour votre réseau peut être utilisé pour envoyer votre mail[37]. Ainsi, si vous êtes client chez *Proximus,* vous devez utiliser leurs serveurs pour envoyer vos mails.

Cet élément est très important pour éviter les serveurs mails configurés en *open-relay*, c'est-à-dire des serveurs qui relaient des mails de n'importe qui. Ils peuvent alors servir massivement à transmettre du SPAM.

12.4 Configuration de sendmail

La configuration de *sendmail* est localisée à deux endroits distincts :

1. Le fichier `/etc/aliases` qui reprend *les alias* pour les mails
2. Le dossier `/etc/mail/` qui reprend la configuration du serveur mail

Nous allons aborder ces deux éléments.

12.4.1 Les alias

Un *alias* est un raccourci pour une boîte mail. On peut configurer des *alias* pour des usages divers : changer le nom d'une boîte mail, ajouter nom supplémentaire à la boîte, créer des adresses de groupes qui transmettent le mail à plusieurs destinataires, …

Ces *alias* sont concentrés dans le fichier texte `/etc/aliases`. Chaque ligne définit un *alias* et est structurée simplement : on commence par indiquer *le nom de l'alias (son identifiant, ce que l'utilisateur devra entrer)*, on continue avec « : » et à droite, on indique le (ou les) destinataires qui recevront ce courrier.

Ainsi, nous pourrions définir les *alias* suivants :

```
maitre.swinnen: p010544
prof-b3: p010544, d.bayers@helmo.be, v.reip@helmo.be, c.mathy@helmo.be
```

La modification du fichier `/etc/aliases` doit être suivie du lancement de la commande :

```
$ newaliases
```

Cette commande permet la prise en compte des *alias* présents.

Comme nous pouvons le voir, il est possible pour un *alias*, de mélanger *des comptes locaux* (comme *p010544*) et des adresses mails. Ainsi l'envoi d'un mail à destination de `prof-b3@mondomaine.be` (pour autant qu'une entrée `MX` dans le DNS mentionne que les mails de ce domaine sont gérés par le serveur) sera transmis à 4 personnes différentes.

L'envoi d'un mail à `maitre.swinnen@mondomaine.be` placera le mail dans la boite mail de l'utilisateur local *p010544*.

[37] Cette remarque est *partiellement vraie* seulement puisque grâce aux extensions d'authentification et de cryptage, il est possible d'envoyer des mails depuis des réseaux "étrangers". Ainsi, le serveur mail de HELMo autorise l'envoi de mail depuis n'importe où si le client s'authentifie au préalable. Par contre, sans authentification, seuls les ordinateurs connectés au réseau de HELMo peuvent utiliser le serveur pour envoyer leur mail.

12.4.2 Configuration du serveur

La configuration du serveur mail est concentrée dans le dossier `/etc/mail`. Nous allons, dans la suite, détailler les fichiers de configuration et les options qu'il faut configurer.

Notons au passage que les boîtes aux lettres des utilisateurs sont, par défaut, stockée dans le dossier `/var/spool/mail/<login>`

Etape 1 : le fichier access

Le premier fichier sur lequel nous allons nous concentrer est le fichier `/etc/mail/access`. Ce fichier détermine quelles machines peuvent utiliser ce serveur comme serveur d'envoi. Ce fichier peut également être utilisé pour interdire certaines adresses mails connues de spammeurs.

Par défaut, il contient les entrées suivantes :

```
Connect:localhost.localdomain          RELAY
Connect:localhost                      RELAY
Connect:127.0.0.1                      RELAY
```

Ces entrées mentionnent que seul l'ordinateur local (*localhost* ou l'IP 127.0.0.1) peut utiliser le serveur pour relayer des mails. On peut mentionner également des sous-réseaux comme suit :

```
Connect: 192.168.128                   RELAY
```

Ainsi, nous venons d'autoriser toutes les machines dont l'adresse IP se trouve dans ce sous-réseau (*192.168.128.x*) à utiliser ce serveur pour transmettre des mails vers l'extérieur.

Nous pouvons également mentionner des adresses mails que nous souhaitons proscrire :

```
l.swinnen@helmo.be                     REJECT
```

Ca peut-être une mesure simple pour combattre les mails publicitaires.

Etape 2 : Le fichier local-host-names

Le fichier `/etc/mail/local-host-names` est un autre fichier très important dans la configuration du service mail. En effet, grâce à ce fichier, le serveur mail peut déterminer **si le mail <u>est arrivé</u> à destination**.

Comment le serveur peut-il déterminer si le mail est arrivé à destination ? En fait, il doit analyser **la partie droite de l'adresse mail** (derrière le symbole « @ »). Il extrait donc le nom de domaine qui est mentionné et va vérifier si :

- Il s'agit du nom de la machine (le *hostname*) ;
- Le nom apparait dans le fichier `local-host-names` ;

Si c'est le cas, le mail est considéré comme arrivé à destination par le serveur et il va alors déterminer dans quelle boîte mail il doit placer ce courrier.

Ainsi pour le serveur mail de HELMo, nous trouvons les informations suivantes dans ce fichier :

```
helmo.be
helmo.eu
student.helmo.be
mx.helmo.be
smtp.helmo.be
mail.helmo.be
relay.helmo.be
```

```
salto.helmo.be
hemes.be
isell.be
```

Ainsi, lorsqu'un mail arrive avec, comme nom DNS derrière le symbole « @ » l'un de ces domaines, le serveur mail sait qu'il doit délivrer ce courrier localement. Notons au passage que l'on mentionne dans ce fichier aussi bien des noms de domaine (comme `helmo.be`, `helmo.eu`, `student.helmo.be` par exemple) que des noms de machines (comme `smtp.helmo.be` ou `relay.helmo.be` qui sont tous deux des noms différents pour le serveur mail).

Etape 3 : Le fichier relay-domains

Le fichier `/etc/mail/relay-domains` est nécessaire **sur le serveur mail *secondaire* (ou backup)**. Comme annoncé précédemment, le service mail étant assez critique, il est courant de proposer plusieurs serveurs pour traiter le courrier d'un domaine donné. Cependant, comme les boîtes mails sont stockées sur un seul serveur[38], nous devons configurer le serveur *secondaire* comme pouvant relayer certains mails (et donc *mettre en file d'attente*) vers le serveur principal.

Cette méthode est intéressante. Ainsi, lorsque le serveur principal est déconnecté, le serveur mail *secondaire* prend le relai et stocke, temporairement, les mails qui arrivent. Dès que le serveur principal est à nouveau en ligne, le serveur secondaire lui transmet les mails qu'il a reçus. Ces mails peuvent alors être placés dans les boîtes aux lettres des utilisateurs concernés.

Et donc l'objet du fichier `/etc/mail/relay-domains` est de lister les domaines qui peuvent être acceptés par le serveur mail *secondaire*. Ainsi, le serveur mail *secondaire* de HELMo, `smtp2.helmo.be`, mentionne les informations suivantes dans ce fichier :

```
helmo.be
helmo.eu
isell.be
hemes.be
```

Lorsqu'on mentionne `helmo.be`, on autorise le relai pour tous les domaines et sous-domaines de HELMo (comme `student.helmo.be` ou encore `salto.helmo.be`).

Etape 4 : Le fichier virtusertable

Le fichier `/etc/mail/virtusertable` est également un fichier intéressant. En effet, sans ce fichier, c'est en fonction du **login de l'utilisateur** que les mails sont délivrés. Or, il n'est pas souhaitable de publier les logins des utilisateurs à l'extérieur. De plus, pour avoir une grande latitude, ce fichier est très intéressant.

Grâce à ce fichier, nous pouvons faire correspondre une adresse mail à un compte utilisateur (ou une autre adresse mail). **Attention !** Il n'est pas possible, dans ce fichier, de faire correspondre à une adresse mail plusieurs destinataires.

Exemple de contenu pour le fichier `/etc/mail/virtusertable` :

[38] Ceci n'est pas nécessairement vrai. Il est très courant de stocker les boîtes mails sur un stockage réseau en haute disponibilité (comme un SAN) et celui-ci est alors accessible sur plusieurs serveurs. C'est d'autant plus vrai si l'entreprise doit gérer beaucoup de mails et donc met en place plusieurs serveurs mails fonctionnant en parallèles. Cependant, dans notre leçon, nous ne considérerons que le cas d'un serveur *principal* avec les boîtes aux lettres des utilisateurs et un serveur secondaire pouvant relayer les mails vers le serveur principal.

```
l.swinnen@helmo.be          p010544
l.swinnen@helmo.eu          p010544
godswila@helmo.be           l.swinnen@helmo.be
```

Attention ! Le séparateur est bien une (ou plusieurs) tabulation(s).

Comme nous pouvons le voir, les entrées sont **des adresses mails complètes**. Il est donc possible d'inclure des domaines comme helmo.be et helmo.eu par exemple. Il est également possible de mentionner une adresse mail (qui ne doit pas être nécessairement dans les domaines gérés par ce serveur).

Etape 5 : Le fichier sendmail.mc

La dernière étape consiste à modifier ou adapter la configuration du serveur mail via le fichier `/etc/mail/sendmail.mc`. Ce fichier est un peu particulier car il contient des *commandes de configuration*.

Ces *commandes de configuration* seront ensuite utilisées pour générer le fichier `/etc/mail/sendmail.cf` qui décrit exactement le comportement attendu du service *sendmail*. Cependant, la compréhension et l'analyse du fichier `sendmail.cf` est particulièrement éprouvante. C'est pourquoi les *commandes de configuration* sont bien plus simples lorsqu'il s'agit de configurer *sendmail*.

Dans le fichier `sendmail.mc`, les **lignes commençant** par `dnl` sont des lignes de commentaires.

Voici un certain nombre d'options importantes :

Option	Explication
`define(`confSMTP_LOGIN_MSG', `$j Sendmail; $b')dnl`	Cette option détermine les informations affichées par *sendmail* lors de la connexion. Afin de ne pas annoncer la version, il est intéressant de modifier cette option.
`define(`SMART_HOST', `smtp.your.provider')dnl`	Cette option permet de transmettre les mails sortant (ceux qui ne sont pas arrivés à destination) à un autre serveur mail qui se chargera de l'expédition. Cette option est nécessaire parfois à cause des mesures de sécurité de certains ISP (comme le blocage des ports 25 sur les réseaux DSL).
`define(`confAUTH_OPTIONS', `A p')dnl` `TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl`  `define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl`	L'option `confAUTH_OPTIONS` contrôle si l'authentification est activée sur le serveur SMTP. Par défaut, aucune authentification n'est nécessaire. Les options `TRUST_AUTH_MECH` et `confAUTH_MECHANISMS` sont nécessaires pour indiquer au serveur mail comment cette authentification peut avoir lieu.

`define(`confCACERT_PATH', `/etc/pki/tls/certs')dnl`
 	
`define(`confCACERT', `/etc/pki/tls/certs/ca-`
	Ces options actives SSL au niveau du serveur mail. L'activation de SSL est nécessaire si l'authentification utilise le login et le mot de passe de l'utilisateur. Ainsi ces informations sont

`bundle.crt')dnl` `define(`confSERVER_CERT',` `/etc/pki/tls/certs/sendmail.pem')dnl`  `define(`confSERVER_KEY',` `/etc/pki/tls/certs/sendmail.pem')dnl`	transmises de manière chiffrée au serveur. Il faut indiquer l'emplacement des certificats (le chemin ou le fichier contenant les autorités racines, et le chemin vers le certificat serveur) et la clé privée.
`DAEMON_OPTIONS(`Port=smtp,` ~~`Addr=127.0.0.1`~~`, Name=MTA')dnl`	Cette option est activée et la mention de l'adresse *Addr=127.0.0.1* empêche le serveur mail d'être disponible sur le réseau. En effet, par défaut, seul *localhost* peut atteindre celui-ci. Pour que le serveur puisse être utilisé, il est nécessaire de retirer la mention de l'adresse.

Etape 6 : Activation de la configuration

Comme nous l'avant découvert à l'étape précédence, le fichier sendmail.mc contient *des commandes de configuration.* Une fois ces commandes correctement configurées, il faut activer la configuration en construisant le fichier sendmail.cf

Pour ce faire, il suffit de lancer, à l'invite de commande :

```
$ make -C /etc/mail
```

Cette instruction `make` va construire le fichier `sendmail.cf` conforme aux *commandes de configuration*.

Pour redémarrer le service mail, il faut passer par la commande `systemctl` :

```
$ systemctl restart sendmail
```

ATTENTION ! CETTE CONFIGURATION DE BASE N'INCLUT AUCUNE PROTECTION CONTRE LES SPAM. IL EST IMPORTANT, DANS UN ENVIRONNEMENT DE PRODUCTION, DE PRENDRE DES DISPOSITIONS POUR ERADIQUER LES COURRIERS INDESIRABLES.

12.5 Les techniques anti-spam

Les techniques anti-spam sont nombreuses et très différentes. Ainsi, bon nombre d'entreprises ont fait de ces techniques des solutions vendues et, parfois, très chères. Ainsi, *IronPort* est une solution complète pour éradiquer le spam. Sans débourser un euro, il est possible d'adjoindre des mécanismes anti-spam à son serveur mail *sendmail*. Nous allons dans ce paragraphe introduire brièvement certaines techniques :

Outils	Description
MailScanner	MailScanner est un ensemble logiciel complet utilisé pour gérer les mails et les spams en se basant sur *sendmail*. Il permet de combiner bon nombre de techniques connues comme des antivirus, Spam Assassin (détecteur probabiliste), des listes RBL (liste de serveurs connus pour transmettre des spams), des outils comme *Pyzor* ou *DCC* (qui utilisent des réseaux *peer-to-peer* pour s'échanger des empreintes de SPAM). Les techniques ici sont donc nombreuses et pas toujours évidentes à appréhender. Cependant, la stabilité et l'intégration de cet outil est

	vraiment un élément important. A titre d'exemple, le serveur mail de HELMo utilise MailScanner.
greylist-milter	Ajout à *sendmail* pour implémenter une *liste grise*. Alors que les *listes blanches* (toujours autorisé) et les *listes noires* (tout est rejeté) sont bien connues, l'instauration de la *liste grise est un peu particulière*. En fait, la norme SMTP prévoit que la boîte aux lettres d'un utilisateur *peut être inaccessible* momentanément. Dans ce cas, le serveur mail est tenu de renvoyer le mail plus tard, jusqu'à ce qu'il soit accepté. Or, les outils anti-spam n'ont bien souvent pas la capacité de retransmettre le spam si le serveur répond qu'il faut *revenir plus tard.* De plus, le *délai* peut être déterminé en fonction du serveur qui transmet la demande (on peut, pour les pays réputés *plus spammeurs*, allonger le délai d'attente). C'est donc une technique assez efficace qui, ajoutée aux autres, permettent de diminuer significativement les spams.
Liste RBL	Il est possible d'indiquer des listes spécifiques, à interroger, lorsqu'un mail est reçu d'un serveur. L'intérêt est de déterminer si le serveur mail est fiable et s'il est autorisé à envoyer un mail. Par exemple, toutes les connexions utilisant des adresses IP dynamiques sont automatiquement répertoriées comme n'étant pas autorisée à envoyer un mail. L'utilisation de ces listes permet ainsi de diminuer les spams également. Les listes intéressantes sont : • `dul.dnsbl.sorbs.net` • `sbl.spamhaus.org`
DCC / Pyzor	DCC et Pyzor sont deux outils plutôt surprenant car ils échangent, en *peer-to-peer*, des empreintes de mail spam. Ainsi, un mail transmis massivement peut ainsi se faire repérer grâce aux échanges réalisés par ces outils.

12.6 POP3 et IMAP

Comme mentionné en introduction, SMTP est utilisé pour l'envoi du mail alors que pour réceptionner les mails, il faut utiliser POP3 ou IMAP. Le choix entre les deux services est souvent une question de politique d'entreprise. IMAP stocke les mails sur les serveurs de l'entreprise alors que POP3 télécharge les mails sur les postes clients.

A l'instar du protocole SMTP, il existe de nombreux programmes permettant de gérer la réception des mails par POP3 ou IMAP. Citons, parmi les programmes serveurs courant *courrier, cyrus,* ou encore *dovecot*. C'est ce dernier que nous allons aborder dans cette leçon.

Le programme *dovecot* implémente aussi bien les protocoles POP3 qu'IMAP. On peut même décider d'activer les deux protocoles en même temps (ce qui ne pose aucun problème puisque des ports TCP différents sont utilisés pour chaque service). Nous allons, dans la suite de ce paragraphe, analyser les options de configuration de *dovecot.*

La configuration de *dovecot* est concentrée en 2 endroits : le fichier `/etc/dovecot/dovecot.conf` et le dossier `/etc/dovecot/conf.d`.

12.6.1 Le fichier dovecot.conf

Dans le fichier `/etc/dovecot/dovecot.conf`, il y a des éléments de configuration qu'il est intéressant de pointer :

Option	Explication
protocols	Permet d'activer les protocoles souhaités. Les options valides sont `imap`, `pop3` et `lmtp`. Par défaut, les trois protocoles sont actifs.
listen	Permet de déterminer sur quelle adresse IP le serveur écoute. Par défaut, il écoute sur toutes les adresses IP configurées sur la machine.
login_greeting	Permet de changer l'information retournée par le serveur lors de la connexion. Cela permet de brouiller un peu les pistes en n'annonçant pas que dovecot est utilisé. `login_greeting = oui allo ?`

12.6.2 Le dossier conf.d

Dans le dossier `/etc/dovecot/conf.d`, tous les fichiers `.conf` sont pris en compte par le serveur comme des ajouts de configuration. Nous n'allons pas passer en revue tous les fichiers mais juste certaines options contenues dans certains d'entre-eux.

Fichier	Option	Explication
10-auth.conf	`disable_plaintext_auth`	Désactive l'authentification en clair à moins que SSL soit utilisé. Cette option doit être placée à `no` si l'on souhaite autoriser l'authentification sans certificat.
10-auth.conf	`auth_username_format`	Cette option permet d'indiquer si le nom d'utilisateur doit être transformé avant de tenter une authentification. Ainsi, mettre l'option `%Lu` assure que le login sera transformé en minuscule au préalable.
10-mail.conf	`mail_location`	Indique l'emplacement de la boite mail de l'utilisateur. `mail_location = mbox:~/mail:INBOX=/var/spool/mail/%u` Cette valeur mentionne que la boite aux lettre se trouve dans le dossier personnel de l'utilisateur et dans le dossier: `/var/spoo/mail/login`
10-ssl.conf	`toutes`	Ce fichier contient la configuration pour l'activation de SSL sur IMAP et POP3. Ces options sont intéressantes pour sécuriser l'authentification.

		Cette sécurité **est activée par défaut**. Cependant, sans certificat valide, la configuration des postes clients risque de poser quelques soucis.

Voici un récapitulatif des ports utilisés suivants les protocoles activés :

Protocole	SSL / Certificat requis ?	Port TCP
SMTP (envoi de courrier) sendmail	Sans SSL	25
SMTPS (envoi de courrier) sendmail	Avec SSL+Certificat	465
POP3 (téléchargement mail) dovecot	Sans SSL	110
POP3S (téléchargement mail) dovecot	Avec SSL+Certificat	995
IMAP (consultation mail depuis le serveur) dovecot	Sans SSL	143
IMAP (consultation mail depuis le serveur) dovecot	Avec SSL+Certificat	993

12.6.3 Démarrer le service dovecot

Le démarrage du service *dovecot* se fait grâce à la commande `systemctl` :

```
$ systemctl start dovecot
```

12.7 Installation d'un Webmail

Il existe de nombreux programmes *Webmail* open-source. Parmi les plus courants, citons *IMP/Horde*[39], *squirrelmail*[40] ou encore *RoundCube*[41] (qui est installé notamment à HELMo). Il en existe bien d'autres, notamment tous les logiciels collaboratifs cités en introduction de cette leçon.

Bon nombre de ces sites sont développés en PHP + MySQL. Pour les utiliser, il faut donc activer le serveur web Apache, s'assurer que PHP est installé et fonctionnel et, finalement, configurer une base de données pour que ces sites puissent l'utiliser.

Dans cette leçon, nous ne verrons pas l'installation étape par étape de ces programmes Webmail. En effet, l'installation est simple et est complètement décrite sur le site du fournisseur. C'est ainsi que l'installation et le programme RoundCube peuvent être téléchargé depuis le site web `http://www.roundcube.net`.

[39] http://www.horde.org/apps/imp/
[40] http://www.squirrelmail.org/
[41] http://www.roundcube.net

12.8 Exercices

On vous demande de :

1. Configurer votre serveur mail (sans SSL, ni authentification, ni techniques antispam) pour le domaine configuré (voir leçon DNS)
2. Installer un serveur POP3/IMAP sur votre machine et configurer celui-ci
3. Configurer Thunderbird installé sur le poste Windows pour interroger votre serveur mail
4. Capturer le trafic sur votre serveur mail pour trouver le mot de passe échangé

Testez votre configuration en envoyant des mails à vos comptes locaux (en effet, la configuration ne permettra pas d'envoyer des mails vers Internet).

5. Installer Roundcube Webmail sur votre serveur pour consulter, par le web, les mails envoyés et reçus.