R3.03 – Services réseaux avancés

Antoine Pernot – antoine.pernot@iut-dijon.u-bourgogne.fr

20 – 22 novembre 2024

Relecteurs

Arnaud Bitterlin
Ingénieur Réseaux et Télécommunications
IUT R&T Colmar
Université de Technologie de Troyes

Loïc Defief
Ingénieur Réseaux et Télécommunications
IUT R&T Dijon-Auxerre
Université de Technologie de Troyes

Introduction et objectifs

Les objectifs de ce cours est d'appréhender les concepts théoriques et mettre en œuvre différents services réseaux, à savoir :

— Le serveur DNS

— Le serveur SMTP

— Le serveur POP3 et IMAP

Les aspects juridiques, ainsi que ceux liés à la sécurité seront également abordés pour la partie SMTP.

Les supports de cours, TD et TP, ainsi que des ressources complémentaires sont disponibles sur https ://rtaux.antoinepernot.fr

Les comptes-rendus de TP doivent être rendus, par courriel, avant le 24 novembre 2024 à 23 heures. L'heure de réception du courriel faisant foi. Tout livrable reçu en retard sera pénalisé à raison d'un point en moins par heure de retard.

L'évaluation de ce module est répartie comme suit :

— Travaux pratiques : 70%

— Examen final : 30%

Une question, remarque ou suggestion ? Vous pouvez me contacter par courriel à l'adresse : antoine.pernot@iut-dijon.u-bourgogne.fr

Table des matières

Cours

1 Le système DNS

Afin de communiquer sur Internet, il est nécessaire de contacter un serveur avec une adresse IP. Or, cela pose des inconvénients :

— Cela est difficile à retenir et peu pratique

— Si le serveur change d'hébergeur, l'adresse IP peut changer

Pour remédier à cela, nous utilisons des noms de domaine. Ces derniers sont associés aux adresses IP.

Le DNS (*Domain Name System*) est un mécanisme **distribué** permettant de traduire les **noms de domaine** (tels que google.fr, fr.wikipedia.org, etc.) en **adresse IP**. Cependant, d'autres informations peuvent également y être renseignées, nous les verrons plus loin. Le DNS s'apparente au carnet d'adresses en associant un nom de contact à un numéro de téléphone.

Il remplace le fichier `hosts` (RFC[1] 608), encore présent aujourd'hui sur les systèmes d'exploitation. Ce fichier contient l'association adresse IP $\leftrightarrow$ nom de domaine. Il fut créé pour ARPANET et remplissait cette tâche à une époque où les équipements sur le réseau étaient peu nombreux.

```
127.0.0.1               localhost
127.0.1.1               antoine-laptop
216.58.213.131          google.fr

# The following lines are desirable for IPv6 capable hosts
::1                     ip6-localhost ip6-loopback
fe00::0                 ip6-localnet
ff00::0                 ip6-mcastprefix
ff02::1                 ip6-allnodes
ff02::2                 ip6-allrouters
2620:0:862:ed1a::1      fr.wikipedia.org
```

FIGURE 1 – Exemple de fichier `hosts`

La très forte augmentation du nombre d'équipements sur le réseau rendit sa mise à jour et sa diffusion compliquées (transmis par FTP). Un système **décentralisé** et **hiérarchique** devenait nécessaire et c'est dans ce cadre que le système DNS est né.

Le service DNS écoute sur le port **UDP 53**.

1. *Request For Comments* : documents de spécifications techniques d'Internet

1.1 Composition d'un nom de domaine

Le système DNS fonctionne de manière **hiérarchique**. Voici la position du nom de domaine dans une URL [2] Web :

`https://`	`fr`	`.`	`wiktionary`	`.`	`org`	`/wiki/DNS`
Protocole	3e niveau	Séparateur	2e niveau	Séparateur	TLD	Ressource

ATTENTION : Le protocole et les ressources ne font pas partie du domaine.

Le domaine dans une adresse de messagerie se trouve **après le symbole arobase (@)** :

`alain@`	`example`	`.`	`com`
Identifiant	2e niveau	Séparateur	TLD

Le nom de domaine se décompose en **partant de la fin**.

1.2 L'arborescence du DNS

Le système DNS est une **hiérarchie** au sommet de laquelle on retrouve **la racine** représentée par le symbole **point (.)**. Il est souvent implicite lors de l'écriture des noms de domaine, mais il doit apparaître lors de la désignation d'un nom de domaine absolu, notamment dans la configuration des serveurs DNS.

Chaque sous-domaine est rattaché à son domaine parent. Ainsi, les TLD ont des sous-domaines rattachés directement à la racine. Chaque niveau de domaine est séparé du précédent par un point (.). Voici une représentation graphique de l'arborescence DNS *(fig. 2)*.

2. *Uniform Resource Locator* : Adresse unique permettant d'accéder à une ressource sur le Web.

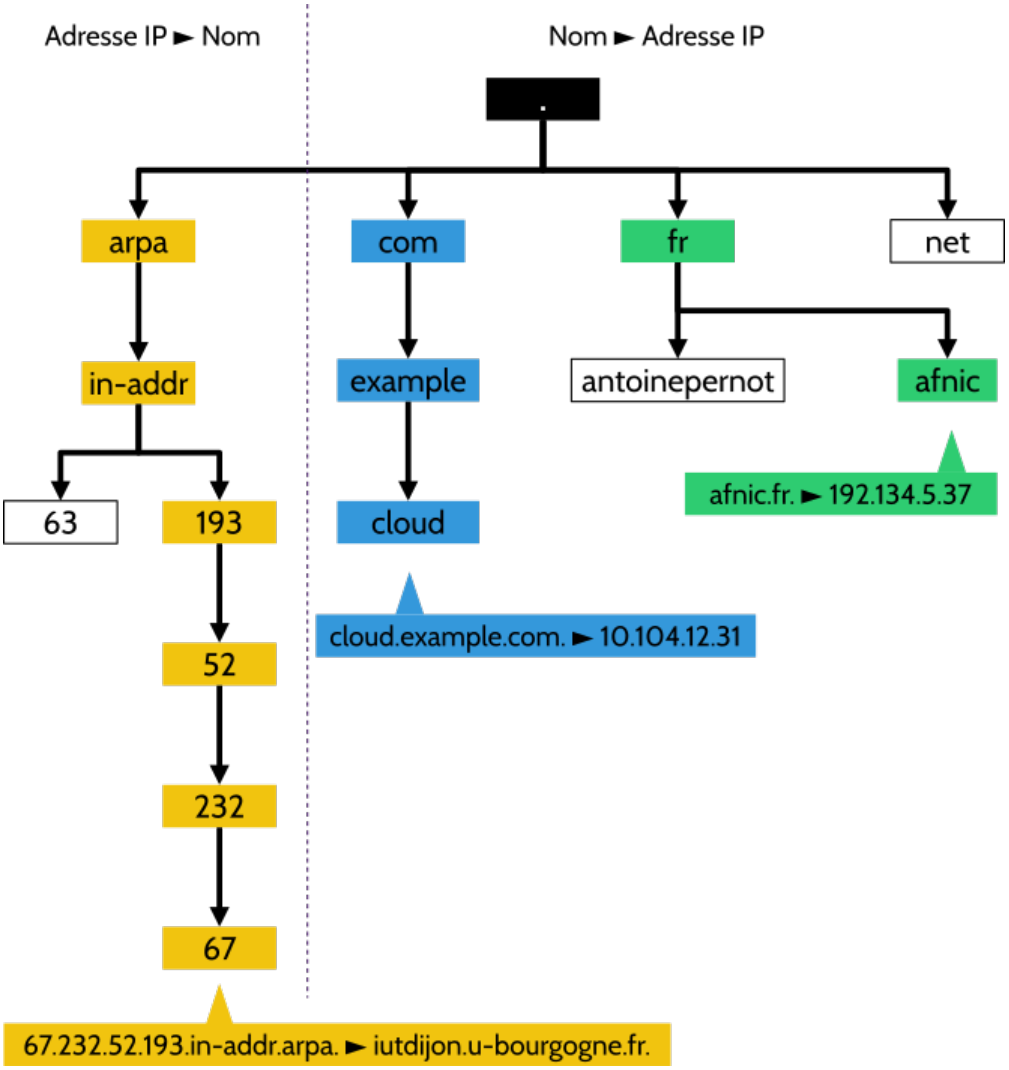


FIGURE 2 – Représentation graphique de l'arborescence DNS

Chaque nœud possède **une base de données** qui détaille les nœuds enfants (sous-domaines, machines, etc.). Ce qui permet une gestion **décentralisée** de l'ensemble du système.

1.3 Les *Top Level Domains* (TLD)

Le **domaine de premier niveau** est situé le plus à **droite** du nom de domaine. Les TLD sont gérés par l'ICAAN. On y retrouve deux types de domaines :

Les domaines nationaux : Ce sont des domaines dédiés à un pays ou territoire (`.fr` pour la France, `.be` pour la Belgique, `.re` pour La Réunion, etc.).

Les domaines génériques : Ils visent à regrouper les domaines partageant une caractéristique autre que géographique (`.com` initialement prévu pour les organismes à but lucratif, `.museum` pour les musées, etc.)

Chaque TLD est géré par un **registre**, qui peut être une association, un état ou tout autre organisme (Le `.fr` est géré par l'AFNIC, une association loi 1901). Chaque registre peut à son tour déléguer la vente des domaines à des sociétés (OVH est habilité par l'AFNIC pour la vente de domaines en `.fr`).

Chaque TLD possède des **conditions et des tarifs propres** afin d'attribuer un domaine (pour le `.fr`, l'AFNIC impose que la personne morale ou physique demandant un nom de domaine soit résident de l'UE, de la Suisse, de la Norvège, de l'Islande ou du Liechtenstein).

1.4 Les sous-domaines

Le DNS fonctionnant de manière hiérarchique, chaque niveau est un sous-domaine du niveau précédant. Ainsi, les TLD sont des sous-domaines de la racine. Il est possible d'avoir plusieurs niveaux de sous-domaines. Voici un exemple :

`fr`	`.www`	`.example`	`.com`	`.`
4e niveau	3e niveau	2e niveau	TLD (1e niveau)	Racine

1.5 Fonctionnement de la résolution de nom de domaine

La résolution du nom de domaine commence par le **TLD** et remonte jusqu'au **sous-domaine**. Le schéma ci-dessous décrit l'ordre des requêtes et des réponses entre le client, le serveur DNS récursif et les serveurs DNS des différentes composantes du nom de domaine demandé *(fig. 3)*.

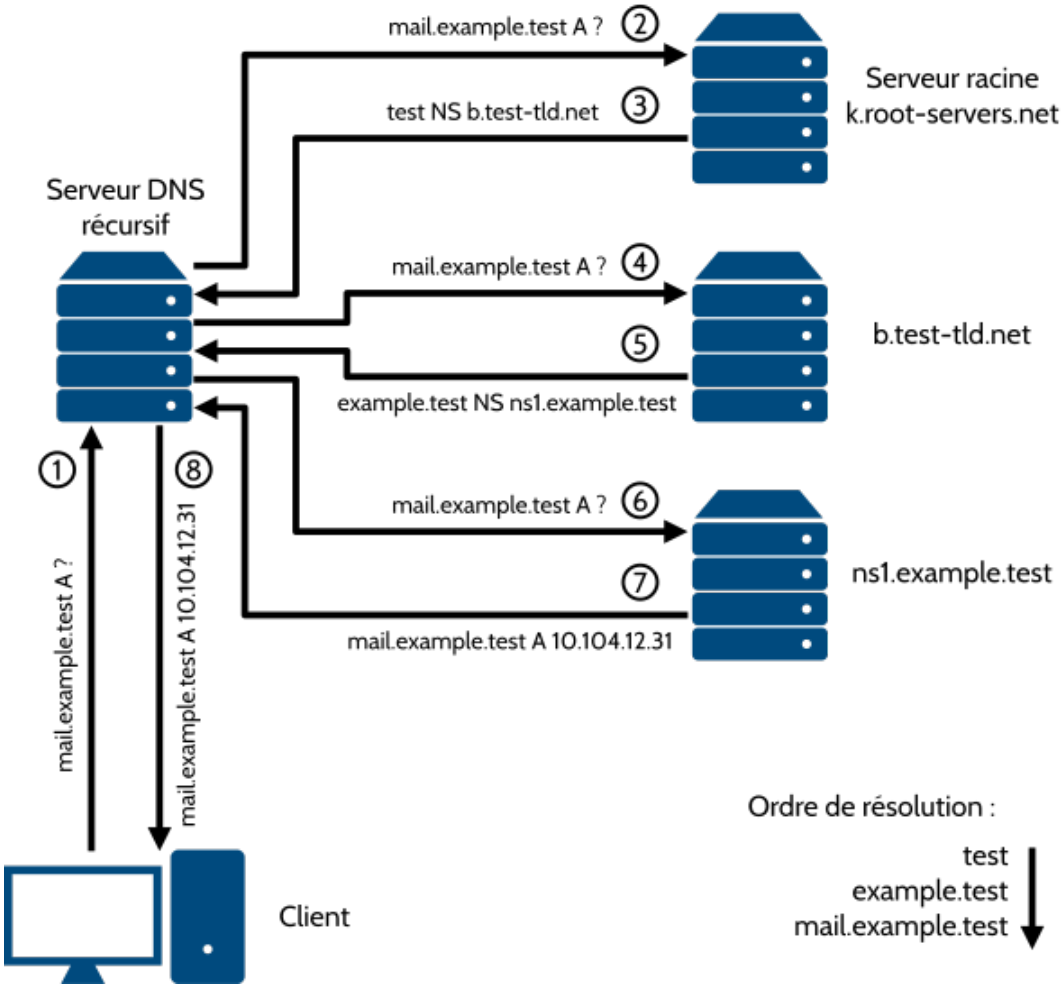


FIGURE 3 – Étapes d'une résolution de nom par un serveur DNS

Exercice 1

Souligner le nom de domaine dans les adresses suivantes :

a. `https://cloud.example.com/`
b. `http://www.cloud/example.com`
c. `example@org.example.com`
d. `video.dns@https.example.net`
e. `https://www.example.com.cloud/`
f. `example.com@test.local`

1.6 Résolution inverse

Ce mécanisme permet de trouver le nom de domaine associé à une adresse IP. Cette information est retournée par un enregistrement **PTR**. Comme la partie désignant le plus grand ensemble d'une adresse IP se situe à gauche (premier octet d'une IPv4), et non à droite pour le nom de domaine (TLD), on cherche a résoudre le domaine composé de l'IP inversée concaténée avec `in-addr.arpa`. Ainsi, pour l'IPv4 10.104.12.31, on résout le domaine 31.12.104.10.in-addr.arpa.

Un enregistrement PTR correct pour les domaines est indispensable pour les services exposés derrière des IP publiques. Par exemple, un courriel provenant d'un serveur SMTP n'ayant pas de PTR valide a de grandes chances d'être considéré comme indésirable et être bloqué.

1.7 Types courants d'enregistrements DNS

Voici les types les plus courants d'enregistrements DNS *(table 1)*.

Enregistrement	Description
A	Associe un nom de domaine à une IPv4
AAAA	Associe un nom de domaine à une IPv6
NS	Identifie le serveur DNS associé à chaque zone
SOA	Fournit des informations sur la zone de domaine (serveur principal, courriel de contact, TTL, etc.)
CNAME	Définit un alias pour un autre domaine
MX	Définit les serveurs SMTP pour le domaine
PTR	Associe une adresse IP à un nom de domaine pour le *reverse* DNS
TXT	Permet de renseigner un texte dans un enregistrement DNS, cela est utilisé par exemple pour la spécification SPF de la messagerie

TABLE 1 – Enregistrements courants DNS

Dans un enregistrement DNS, un **nom de domaine pleinement qualifié** (en anglais *FQDN* pour *Fully Qualified Domain Name*), c'est-à-dire défini de manière **absolue**, se termine par un point (.) qui représente la racine. Sinon, il est relatif au domaine.

Il est possible de faire de l'équilibrage de charge par le DNS avec le mécanisme **DNS round-robin** en inscrivant pour un même domaine, plusieurs enregistrements différents. La réponse à la requête comportera les enregistrements dans un ordre aléatoire. Cela permet de répartir la charge sur les serveurs. Un exemple est donné avec des enregistrements A ci-dessous.

Les exemples suivants sont donnés pour le domaine `example.com`.

Enregistrement A

Voici la syntaxe d'un enregistrement A pour le sous-domaine `cloud`. Les deux enregistrements possèdent les mêmes informations :

```
cloud                   IN A   10.104.12.31
cloud.example.com.      IN A   10.104.12.31
```

Le premier enregistrement est relatif car il ne possède pas de point à la fin du domaine. Le second est absolu.

Voici ci-dessous un exemple de répartition de charge par le DNS :

```
cloud                   IN A   10.104.12.31
cloud                   IN A   10.104.12.32
cloud                   IN A   172.16.85.204
cloud                   IN A   192.168.14.52
```

Enregistrement AAAA

De la même manière que pour un enregistrement A, voici la syntaxe pour un enregistrement AAAA relatif et absolu :

```
cloud                   IN AAAA   2001:412f:c1::1
cloud.example.com.      IN AAAA   2001:412f:c1::1
```

Enregistrement NS

L'enregistrement NS désigne quel(s) serveur(s) DNS est en charge du sous-domaine demandé :

```
example.com.            IN NS  ns0.example.com.
example.com.            IN NS  ns1.example.com.
```

Enregistrement SOA

Un enregistrement SOA permet de renseigner les informations officielles sur la zone DNS :

— Le serveur DNS principal

— Une adresse courriel de contact technique dont le @ est remplacé par un point.

— Un numéro de série de la configuration de la zone. Il doit être incrémenté à chaque version. Par convention, le numéro de série est la date au format `YYYYMMDDNN` soit l'année, le mois, le jour de la modification et le numéro de révision de la journée.

— Le temps de rafraîchissement entre le serveur principal et les serveurs secondaires exprimé en secondes.

— Le temps d'attente, après un essai infructueux de rafraîchissement depuis les serveurs secondaires vers le serveur principal exprimé en secondes.

— Le temps d'expiration, si les serveurs secondaires ne peuvent pas joindre le serveur principal, exprimé en secondes, durée au-delà de laquelle la zone est considérée comme invalide.

— Le TTL [3] des enregistrements dans les caches DNS, en secondes.

La syntaxe est la suivante :

```
example.com. IN SOA ns0.example.com. root.example.com. 2020120401
   43200 7200 259200 3600
```

Enregistrement CNAME

L'enregistrement CNAME permet de faire un alias vers un autre domaine :

```
web                 IN CNAME cloud.example.com.
fr.example.com.     IN CNAME web
```

Enregistrement MX

Cet enregistrement désigne le serveur SMTP auquel il faut envoyer un courriel destiné à une adresse du domaine. Un serveur SMTP va tenter de contacter le serveur dont le numéro de préférence (situé après `MX`) est le plus petit. Un équilibrage de charge est possible si le numéro de préférence est identique à plusieurs enregistrements :

```
example.com.        IN MX 10 mail1.example.com.
example.com.        IN MX 20 mail2.example.com.
example.com.        IN MX 20 mail3.example.com.
```

Enregistrement PTR

Cet enregistrement, également appelé **Reverse DNS Record**, associe une adresse IP à un domaine.

L'adresse IPv4 `10.104.12.31` doit être au format `31.12.104.10.in-addr.arpa`. L'adresse IPv6 `2001:412f:c1::1` doit être saisie (bonne chance !) au format `1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.c.0.0.f.2.1.4.1.0.0.2.ip6.arpa` :

```
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.c.0.0.f.2.1.4.1.0.0.2.ip6.
   arpa. IN PTR cloud.example.com.
31.12.104.10.in-addr.arpa.      IN PTR cloud.example.com.
```

3. *Time To Live* : durée de vie pendant laquelle l'information est conservée en cache

Enregistrement TXT

Cet enregistrement permet de renseigner du texte dans un enregistrement DNS. Nous verrons dans le chapitre SMTP certains de ses usages :

```
example.com.            IN TXT    "Lorem ipsum dolor sit amet"
```

Exercice 2

Fournir l'enregistrement SOA avec les paramètres suivants :

- Le temps d'attente pour un rafraîchissement infructueux est de 10 minutes.
- Contact technique : `pierre.denis@example.fr`
- La configuration est la sixième version éditée aujourd'hui.
- Domaine : example.fr
- Le TTL est de 5 minutes.
- Le rafraîchissement entre le serveur principal et les serveurs secondaires est de 5 minutes.
- L'expiration des serveurs secondaires est de 30 minutes.
- Serveur DNS principal : `dns.example.fr`

Exercice 3

Donner le domaine PTR pour les adresses IP suivantes pour le domaine `example.com` :

Adresse IP	**Réponse**
`192.168.133.196`	
`fe80:fbe:184e:dbd2::a80`	
`172.25.114.40`	
`172.30.177.5`	
`192.168.22.121`	
`10.207.192.196`	

1.8 Interroger manuellement un serveur DNS

GNU/Linux

La commande `dig` affiche les informations renvoyées par le serveur DNS :

```
$ dig iutdijon.u-bourgogne.fr
; <<>> DiG 9.16.1-Ubuntu <<>> iutdijon.u-bourgogne.fr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3728
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;iutdijon.u-bourgogne.fr.  IN A

;; ANSWER SECTION:
iutdijon.u-bourgogne.fr. 360  IN A  193.52.232.67

;; Query time: 68 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: mer. nov. 04 22:46:50 CET 2020
;; MSG SIZE  rcvd: 68
```

Windows

La commande `nslookup` dans l'invité de commandes permet d'effectuer une requête DNS :

```
> nslookup fr.wikipedia.org
Serveur :   Unknown
Address:  192.168.1.254

Réponse ne faisant pas autorité :
Nom :    dyna.wikimedia.org
Addresses:  2620:0:862:ed1a::1
          91.198.174.192
Aliases:  fr.wikipedia.org
```

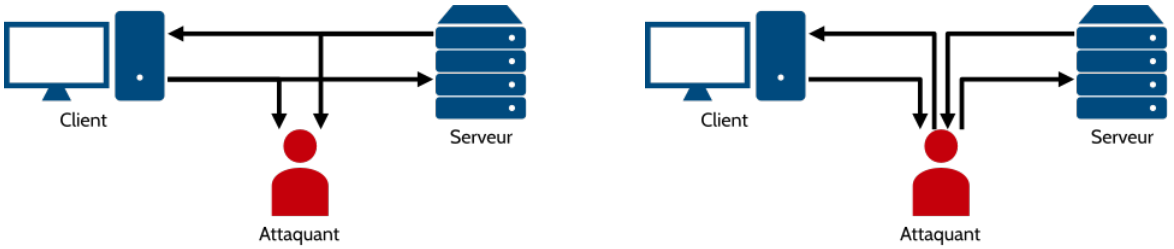
1.9 Sécurité du DNS

Au-delà de l'attaque par déni de service, le service DNS est vulnérable à plusieurs types d'attaques :

Interception et attaque *man in the middle*

L'absence de chiffrement des requêtes et réponses DNS permet **l'interception** de la communication entre le client et le serveur, voire **l'altération** de la requête ou de la réponse par l'attaquant. Le

but est d'obtenir des informations sur les domaines requêtes et de renvoyer de fausses informations (par exemple, renvoyer l'IP d'un serveur malveillant au lieu de l'IP du serveur de votre banque).



(a) Interception (b) Modification

FIGURE 4 – Interception et modifications d'une requête et/ou d'une réponse DNS

Empoisonnement du cache DNS

Cette technique consiste à fournir une mauvaise information à un serveur DNS afin que celui-ci l'insère dans son cache. Cette information erronée sera renvoyée à tout utilisateur du serveurs DNS demandant le même domaine.

DNSSEC

Le DNSSEC permet de résoudre ces problèmes : une **signature cryptographique** est associée à chaque enregistrement d'un serveur doté de ce protocole. Ce qui permet au client de vérifier avec la **clef publique** du serveur que l'enregistrement retourné est bien valide et n'a pas été altéré.

Ce protocole permet également de déléguer les signatures. Un domaine peut déclarer qu'un sous-domaine est signé et ainsi établir une chaîne de confiance jusqu'à la racine.

1.10 Noms de domaine internationalisés

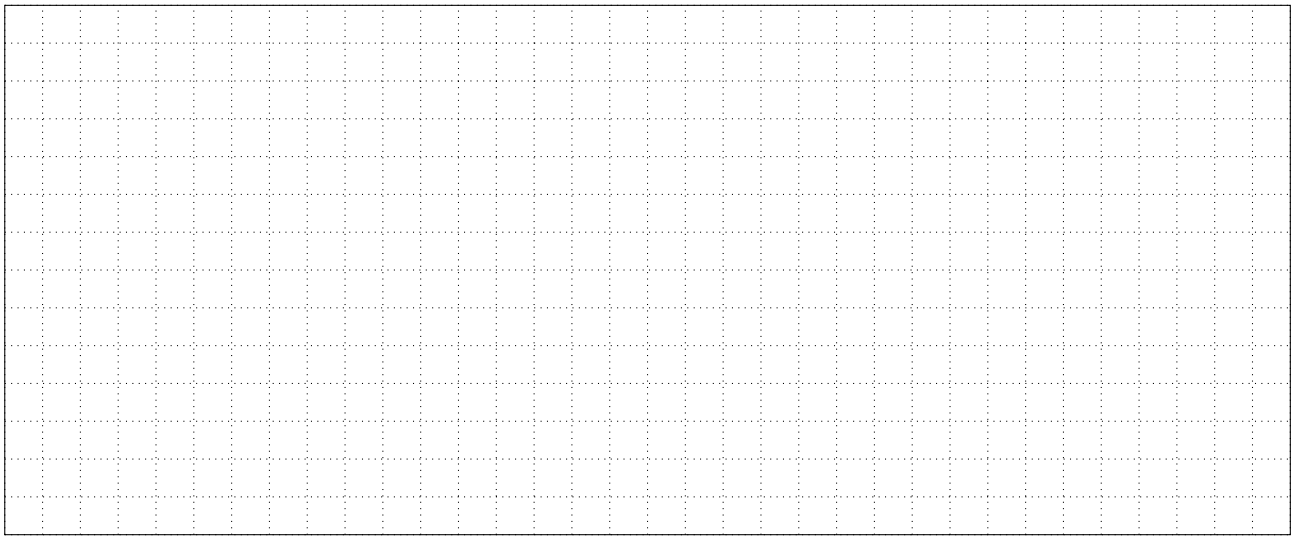
Initialement, les noms de domaine ne pouvaient être composés uniquement des caractères alpha-numériques (a-z et 0-9) sans casse (les majuscules et les minuscules ne sont pas différenciées), ainsi que du trait d'union (-).

La syntaxe *punycode* a été introduite par les RFC 3490, 3491 et 3492 et permet d'encoder des caractères Unicode en chaînes de caractères ASCII afin d'être compatible avec le système DNS.

Exercice 4

Fournir les enregistrements DNS sans PTR ni SOA pour la configuration suivante :

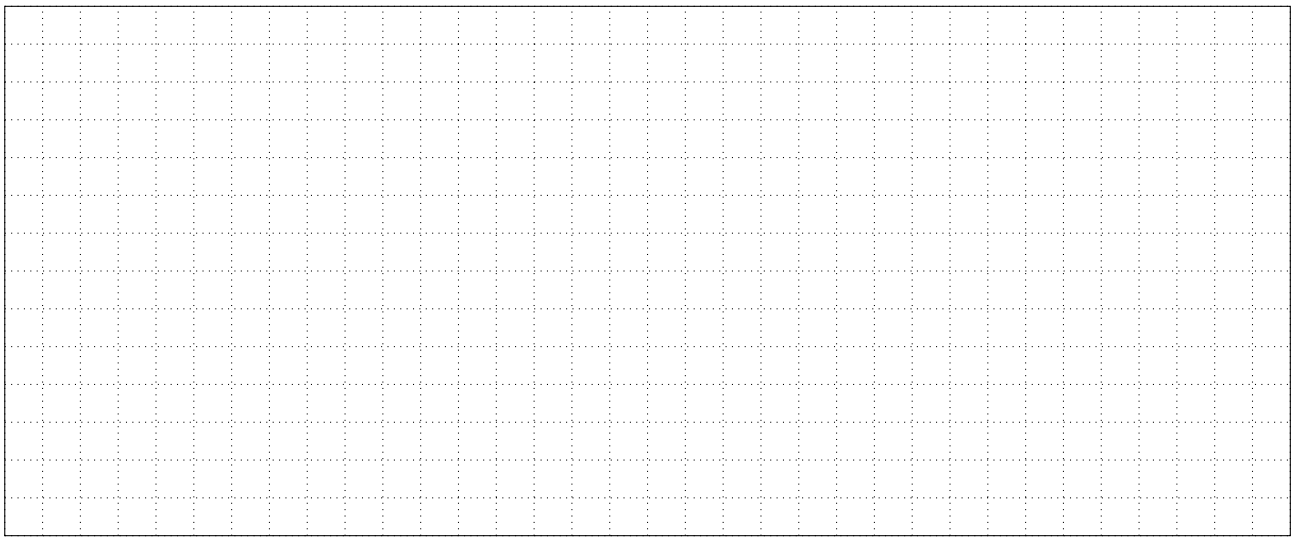
— Serveur DNS du domaine `ns0.example.cloud` : `10.77.154.194`
— Serveur SMTP du domaine `mail.example.cloud` : `fe80::783e:b60e:be0d:32ca`
— Serveur IMAP `imap.example.cloud` : `fe80::11e:f3f2:da9:5732`
— Serveur `suivi.example.cloud` : `10.77.154.194`
— Serveur `tickets.example.cloud` : `10.77.154.194`
— Serveur `factures.example.cloud` : `fe80::11e:f3f2:da9:5732`
— Serveur `printer.example.cloud` : `fe80::783e:b60e:be0d:32ca`



Exercice 5

Fournir les enregistrements DNS PTR sans SOA pour la configuration suivante :

— Serveur `mail.example.fr` : `172.20.84.10`
— Serveur `ftp.example.fr` : `172.20.84.8` et `fe80::9d40:11ee:ac9b:585a`
— Serveur `ldap.example.fr` : `172.20.84.12`
— Serveur `imap.example.fr` : `172.20.84.10`



2 Serveurs de messagerie

La messagerie électronique est l'une des applications les plus anciennes des réseaux informatiques. Sur le réseau ARPANET, ancêtre d'Internet, la messagerie électronique est apparue en 1969.

2.1 Fonctionnement d'un courriel

Un courriel se présente sous la forme d'un **fichier texte**. Voici un exemple du contenu d'un courriel simple : *(fig. 5)*.

```
Received: from iutdijon.u-bourgogne.fr (iut-dijon.u-bourgogne.fr
    [193.52.232.3])
    by example.com (Postfix) with ESMTPS id 1AE55DFE90
    for <jbonbeur@example.com>; Fri,  4 Dec 2020 13:37:18 +0100 (CET)
Received: by example.com (Postfix, from userid 127)
    id 4C1FDDFE19; Fri,  4 Dec 2020 13:37:42 +0100 (CET)
From: Antoine Pernot <antoine.pernot@iut-dijon.u-bourgogne.fr>
To: Jean Bonbeur <jbonbeur@example.com>
Subject: Bonjour !
Date: Fri,  4 Dec 2020 13:37:42 +0100
MIME-Version: 1.0
Content-Transfer-Encoding: 8bit
Content-Type: text/plain; charset=utf-8
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101
 Thunderbird/68.10.0
Message-ID: <754c82a5-9ac6-5d90-b7ec-d897946b7212@iut-dijon.u-
    bourgogne.fr>


Bonjour le monde !
Ceci est un message d'exemple.
Antoine
```

FIGURE 5 – Exemple de courriel avec ses en-têtes

Chaque boîte de réception est un répertoire sur le serveur de messagerie ou un fichier texte dont tous les messages se suivent. Cela est défini dans la configuration du serveur SMTP.

Lors de l'envoi d'un courriel, l'acheminement se fait comme suit :

1. Le courriel est rédigé, puis envoyé en **SMTP** depuis le **Mail User Agent (MUA)** (plus simplement client de messagerie), vers le serveur SMTP également nommé **Mail Transfer Agent (MTA)**.
2. Le premier MTA envoie le courriel vers le MTA du domaine destinataire.
3. Le MTA destinataire transfère le message au **Mail Delivery Agent (MDA)** qui héberge la boîte de réception du destinataire.

4. Le client de messagerie du destinataire (MUA) interroge le MDA afin de récupérer les nouveaux messages en IMAP 4 ou POP 3.
5. Le MDA répond à la requête en fournissant le message envoyé.

Il est courant d'héberger le MTA et le MDA sur le même serveur.

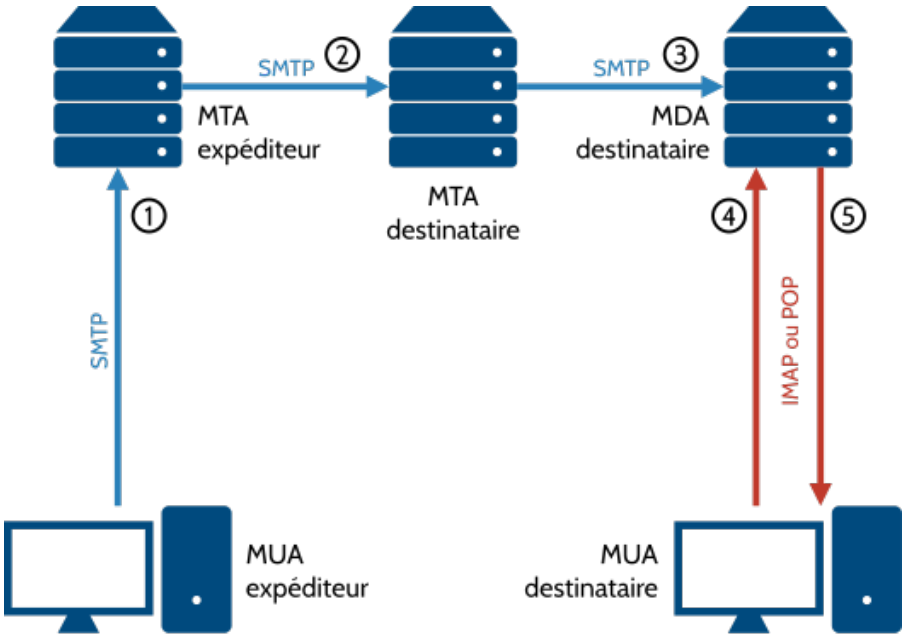


FIGURE 6 – Acheminement d'un courriel

2.2 Les champs destinataires

Une adresse courriel se compose comme suit :

`alain`	`@`	`example.com`
Identifiant utilisateur	Séparateur symbolisant "chez"	Domaine

Le système de courriel admet trois champs permettant de renseigner :

— Les destinataires principaux.

— Les destinataires en copie dont l'adresse est visible par les autres destinataires : **Copie carbone "Cc"**.

— Les destinataires en copie dont l'adresse n'est pas visible par les autres destinataires : **Copie carbone invisible "Cci"**.

Dans les faits, même si un message est destiné à plusieurs destinataires, un seul courriel est acheminé au MTA qui se chargera de le dupliquer pour chaque destinataire.

2.3 Le serveur d'envoi : SMTP

Le protocole utilisé pour l'envoi de courriels est le **Simple Mail Transfer Protocol (SMTP)**. Il écoute par défaut sur les ports suivants (selon la configuration) :

— **TCP 25** (sans chiffrement)

— **TCP 465** (chiffrement implicite)

— **TCP 587** (chiffrement explicite)

Il s'agit d'un protocole simple permettant de se connecter à un serveur et d'envoyer un message en saisissant quelques commandes. Voici ci-dessous un exemple d'envoi de message en saisissant manuellement les commandes SMTP *(fig. 7)*. Ces opérations sont réalisées par le MUA émetteur lors de l'envoi de message.

```
$ telnet example.com 25
Trying 93.184.216.34...
Connected to example.com.
Escape character is '^]'.
220 example.com ESMTP Postfix (Debian/GNU)
HELO antoine
250 example.com
MAIL FROM: <antoine@example.com>
250 2.1.0 Ok
RCPT TO: <alain@example.com>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Subject: Bonjour !

Bonjour le monde !
Ceci est un message d'exemple.
Antoine


.
250 2.0.0 Ok: queued as DE0C7DF3EB
QUIT
221 2.0.0 Bye
```

FIGURE 7 – Envoi de courriel en saisissant des commandes SMTP sans chiffrement

Multipurpose Internet Mail Extensions (MIME)

Initialement, le protocole SMTP ne supporte que l'envoi de textes encodés en ASCII. Le support des autres encodages de caractères (Unicode), ainsi que des fichiers multimédia (images, sons, vidéos, documents bureautiques) est apporté par le format **Multipurpose Internet Mail Extensions (MIME)**.

Des en-têtes supplémentaires spécifiques à MIME sont ajoutées à celles par défaut afin que le client de messagerie destinataire soit informé du format, de l'encodage ou de la présentation du courriel pour l'interpréter correctement.

Il permet ainsi d'étendre les fonctionnalités de la messagerie sans avoir à remplacer les serveurs existants.

Voici quelques en-têtes courantes pour un courriel :

`MIME-Version` : Indique que le message utilise MIME et fournit la version de MIME utilisée (en général 1.0).

`Content-Type` : Définit le format du message. Les plus courants sont :

- `text/plain` pour du texte simple.
- `text/html` pour du contenu formaté en HTML.
- `multipart/alternative` pour envoyer un message contenant plusieurs formats à la fois (HTML et texte simple). Le client de messagerie destinataire choisira le format désiré.
- `multipart/encrypted` pour les messages chiffrés.
- `multipart/signed` pour les messages signés cryptographiquement.
- `multipart/mixed` pour l'envoi des pièces jointes. Chaque élément du message (pièces jointes et texte) est désigné alors par son propre en-tête `Content-Type` et est séparé des autres par un délimiteur spécifié précédé d'un double trait d'union *(fig. 8)*.

```
Content-Type: multipart/mixed; boundary="delimiteur_multipart"
MIME-Version: 1.0

Ce courriel contient une pièce jointe
--delimiteur_multipart
Content-type: text/html; charset=utf-8

Ce courriel contient une <strong>pièce jointe</strong>

--delimiteur_multipart
Content-type: image/gif
Content-transfer-encoding: base64

R0lGODlhQAAQAKEAADRJXv///zRJXjRJXiH5BAEKAAIALAAAAABAABAAAAJmhI+py+0
    Po5y02ruC
3uFw7mxYpCGl2UHi+JyG+6ZhB9e0uzJ2Aus3CgTgZD5Pr6cYGoPCX+
    tDjD2XUmosWqwyfR9rE4XN
EI9h7/esLSGT4+jafBqmlDNeu7GOi9Rylv8PGCg4yFIAADs=

--delimiteur_multipart
```

FIGURE 8 – Exemple de courriel type `multipart/mixed`

Pour les données de type `text`, une spécification sur l'encodage peut y être ajoutée : `Content-type: text/html; charset=utf-8`

`Content-Transfer-Encoding` : Cette en-tête spécifie la méthode utilisée pour convertir les données sous la forme d'une chaîne de caractères ASCII :

- `7bit` indique qu'aucune transformation n'a été appliquée
- `quoted-printable` indique que les caractères non pris en charge dans l'encodage ASCII ont été remplacés par la représentation hexadécimale du ou des octets du caractère, précédé par "=". Par exemple, le caractère "é" de l'encodage UTF-8 est remplacé par `=C3=A9`. Cela permet d'encoder les caractères non ASCII tout en gardant la lisibilité du message.
- `base64` indique que le contenu a été encodé en base64. Chaque groupe de 6 bits a été représenté par un caractère ASCII. Cette méthode permet d'encoder des données binaires, mais cela augmente d'un tiers la taille de la donnée et la rend humainement illisible.

Relais SMTP

Comme son nom l'indique, un relais SMTP permet de transférer les messages émis par un serveur SMTP vers le serveur destinataire. Du point de vue du serveur destinataire, le message provient du serveur relais et non du serveur initial. Cette opération est utile lorsque le serveur SMTP émetteur ne remplit pas les critères nécessaires pour être reconnu comme légitime (adresse IP publique dans une plage d'adresse IP fixe, *reverse* DNS, etc.).

2.4 Le serveur de consultation : IMAP et POP

La consultation des mails sur le MDA peut se faire avec les protocoles **Internet Message Access Protocol (IMAP) 4** ou **Post Office Protocol (POP) 3**.

Le protocole le plus utilisé actuellement est le protocole IMAP car, dans sa configuration par défaut, **il laisse les messages sur le serveur**. Quant à lui, le POP, par défaut, **supprime les messages du serveur après les avoir rapatriés sur le client de messagerie**.

De plus, le protocole IMAP permet de modifier l'état de chaque message sur le serveur (lu/-non lu, répondu, transféré, etc.). Ainsi, le protocole IMAP est plus adapté pour une configuration dans laquelle plusieurs clients de messagerie différents consultent une même boîte de messagerie (ordinateur, téléphone, tablette, etc.).

Le POP non sécurisé écoute sur le port **TCP 110** et la version sécurisée sur le port **TCP 995**. L'IMAP non sécurisé écoute sur le port **TCP 143** et la version sécurisée écoute sur le port **TCP 993**

Exercice 1

Sélectionner quel(s) protocole(s) est/sont en charge des opérations suivantes :

1. Transmettre un courriel du serveur expéditeur au serveur destinataire :

   ☐ POP ☐ IMAP ☐ SMTP

2. Récupérer un courriel du serveur destinataire vers le client courriel destinataire :

   ☐ POP ☐ IMAP ☐ SMTP

3. Envoyer un courriel depuis le client courriel expéditeur vers le serveur expéditeur :

   ☐ POP ☐ IMAP ☐ SMTP

Exercice 2

Quel protocole permet de reçevoir les courriels sur un client tout en les conservant sur le serveur ?

☐ POP ☐ IMAP ☐ SMTP

2.5 Signature et chiffrement cryptographiques

Les procédés de signature et de chiffrement cryptographiques apportent des garanties supplémentaires pour le destinataire du message et ajoutent de la sécurité à la communication.

Les standards de chiffrement et de signature les plus courants sont S/MIME, PGP/Inline et PGP/MIME. Ils se basent sur la **cryptographie asymétrique**.

Contrairement au chiffrement symétrique qui utilise la même clef pour chiffrer et déchiffrer, le chiffrement asymétrique utilise deux clefs :

— **La clef publique** pouvant être diffusée auprès des tiers avec lesquels on communique.

— **La clef privée** devant être gardée secrète.

Ainsi, il est nécessaire à ce que l'expéditeur possède **la clef publique** du destinataire afin de chiffrer ou signer un message lui étant destiné *(fig. 9)*.

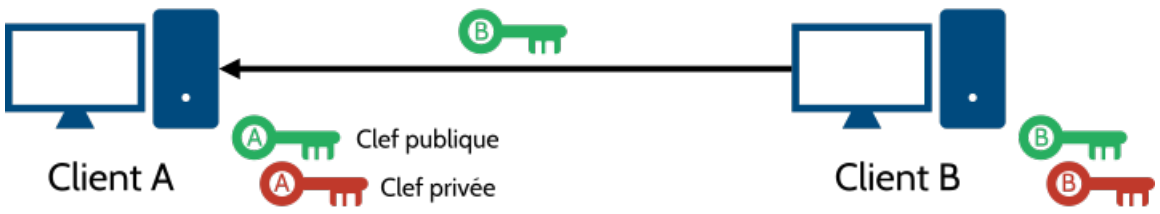


FIGURE 9 – Échange de clef publique de B

Voici la terminologie à employer :

chiffrer : **Transformer un message en clair en un message inintelligible pour assurer le secret de sa transmission.**

déchiffrer : **Traduire en clair un message chiffré à l'aide de la clef de déchiffrement.**

décrypter : **Traduire en clair un message chiffré en ignorant la clef de déchiffrement.**

ATTENTION : Le terme "crypter" reviendrait à chiffrer sans connaître la clef de chiffrement, ce qui n'a aucun sens. L'usage de ce terme est à bannir.

Le chiffrement

Le chiffrement permet de rendre illisible le contenu du message à toute personne ne possédant pas la clef de déchiffrement (**clef privée** dans le cas d'un chiffrement asymétrique).

L'envoi d'un message chiffré de A vers B se fait comme suit *(fig. 10)* :

1. Le message en clair est **chiffré** à l'aide de **la clef publique de B**.
2. Le message chiffré est envoyé à B.
3. Le message est **déchiffré** par la **clef privée** de B.

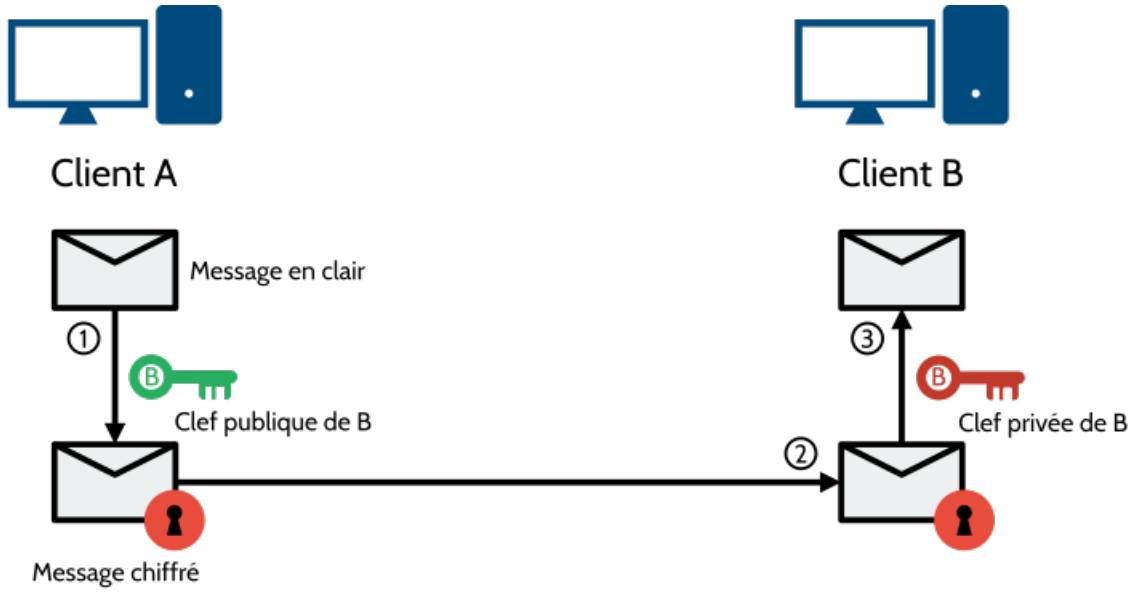


FIGURE 10 – Envoi d'un message chiffré

La signature numérique

La signature numérique permet de garantir l'origine du message et son intégrité. Elle possède les avantages suivants :

— Elle est **authentique et infalsifiable** : le signataire est identifié et son identité ne peut être usurpée.

— La signature est **à usage unique** : on ne peut pas réutiliser la signature d'un message pour en signer un second.

— Le document est **inaltérable** : s'il est modifié, la signature devient invalide.

— La signature est **irrévocable** : le signataire ne peut pas nier la signature vu qu'il est seul a posséder la clef privée.

Ainsi, la signature numérique propose de nombreux avantages par rapport à la signature manuscrite.

L'envoi d'un courriel signé entre A et B s'effectue comme suit *(fig. 11)* :

1. Une **empreinte** du message est calculée à l'aide d'une **fonction de hachage**. Une fonction de hachage produit une **empreinte unique** pour un contenu. Il s'agit d'une **fonction à sens unique** qui ne permet pas à partir de l'empreinte de retrouver le contenu originel. Quelques fonctions de hachages courantes : MD5, SHA1, SHA256, SHA512. Certaines sont a éviter telles que le MD5 et SHA1 car il y a risque de collision (le fait que deux contenus différents aient la même empreinte).
2. L'empreinte est **chiffrée** avec la **clef privée de l'émetteur**. On obtient la **signature** du message.
3. La signature est ajoutée au message en clair.
4. Le message signé est envoyé.
5. La signature est séparée du message.
6. La signature est **déchiffrée** avec la **clef publique de l'émetteur**.
7. **L'empreinte** du message est calculée à l'aide de **la même fonction de hachage** que lors de la création de la signature.
8. L'empreinte obtenue en hachant le message en clair et l'empreinte obtenue en déchiffrant la signature sont comparées. Si elles sont identiques, **le message est authentique**.

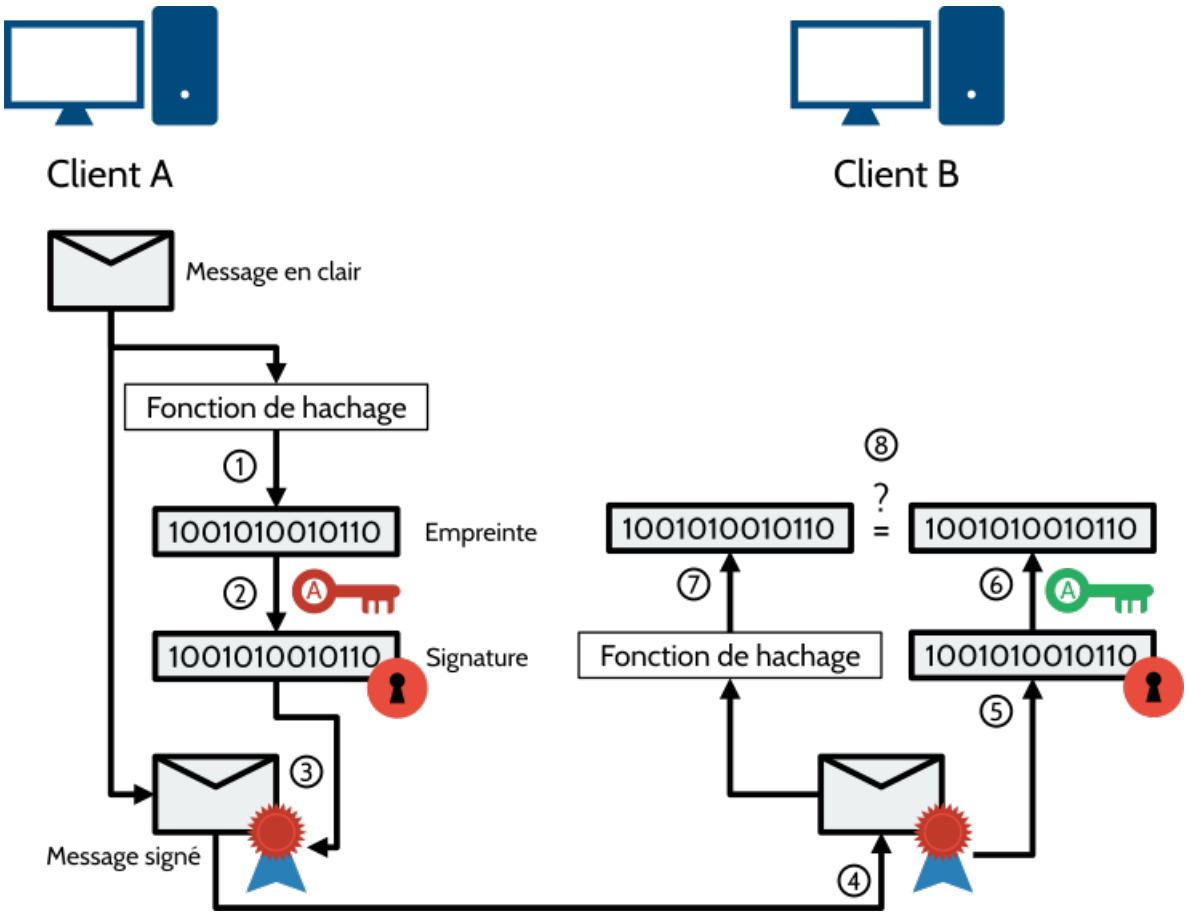


FIGURE 11 – Envoi d'un message signé

Exercice 3

Je souhaite envoyer un message à `alain@example.com`, `bernard@example.com` et `chantal@example.com` avec `martine@example.com` en copie et `pascal@example.com` en copie cachée. Comment doivent-être renseignés les champs destinataires ?

Exercice 4

Quelle clef est utilisée pour les opérations suivantes dans le cadre de l'envoi par Lucie d'un message à Simon :

1. Lucie chiffre son message avant de l'envoyer à Simon :
   ☐ Clef publique de L ☐ Clef privée de L ☐ Clef publique de S ☐ Clef privée de S
2. Lucie signe son message avant de l'envoyer à Simon :
   ☐ Clef publique de L ☐ Clef privée de L ☐ Clef publique de S ☐ Clef privée de S
3. Simon déchiffre un message chiffré reçu de Lucie :
   ☐ Clef publique de L ☐ Clef privée de L ☐ Clef publique de S ☐ Clef privée de S
4. Simon vérifie la signature d'un message signé par Lucie :
   ☐ Clef publique de L ☐ Clef privée de L ☐ Clef publique de S ☐ Clef privée de S

2.6 La norme *DomainKeys Identified Mail* (DKIM)

Le DKIM permet de signer numériquement les messages émis par un domaine. Son fonctionnement est celui d'une **signature cryptographique à clef asymétriques**. Le DKIM permet de s'assurer que le message n'a pas été altéré durant le transport entre les serveurs SMTP. Il est décrit dans la RFC 6376.

La clef publique permettant à un MTA destinataire de vérifier la signature est renseignée dans **un enregistrement DNS TXT** du sous-domaine `_domainkey`. Voici un exemple *(fig. 12)*.

```
mail._domainkey.example.com. IN TXT "v=DKIM1; k=rsa; t=y; h=sha256;
   p=<clef publique encodée en base64>"
```

FIGURE 12 – Enregistrement DNS de la clef publique DKIM

Les informations sont fournies sous la forme "clef=valeur" séparées par un point-virgule *(table 2)*

Clef	Libellé
`v=`	Version de DKIM
`h=`	Liste des fonctions de hachages supportées
`k=`	Type de clef publique
`n=`	Notes humainement lisibles, non interprétées
`p=`	Clef publique encodée en base64
`s=`	Types de services pour lesquels la clef est destinée. Par défaut, tous
`t=`	Drapeaux. `y` : le domaine vérifie l'enregistrement DKIM. `s` : le domaine de la balise "i=" ne doit pas être un sous-domaine de la balise "d="

TABLE 2 – Valeurs possibles dans un enregistrement DNS de DKIM

Chaque courriel émis par le domaine comporte un en-tête supplémentaire de ce type *(fig. 13)*.

```
DKIM-Signature: v=1; a=rsa-sha256; c=simple/simple; d=example.com;
    s=mail; t=1607069100;
    bh=xMoSG8LI7XmdNbRh0iQburgk/ydKxfjnV3uNf9tSjHY=;
    h=To:From:Subject:Date:From;
    b=<signature encodée en base64>
```

FIGURE 13 – Exemple d'en-tête DKIM

Les champs de l'en-tête DKIM du message sont également au format "clef=valeur" et sont séparés par un point virgule *(table 3)*

Clef	Libellé
v=	Version de DKIM
a=	Fonction de hachage utilisée
b=	Signature du message encodée en base64
bh=	Empreinte du corps du message mis en forme canonique
c=	Algorithme de canonicalisation utilisé
d=	Identifiant de domaine responsable de la signature (SDID)
h=	Liste des champs d'en-têtes concernés par la signature
i=	Courriel du responsable du domaine SDID
l=	Taille du corps de message après canonicalisation
q=	Méthodes de requêtes utilisées afin de récupérer la clef publique de signature, séparées par une virgule
s=	Le sélecteur de sous-domaine du SDID
t=	Date et heure de la signature au format timestamp UNIX
x=	Date et heure de l'expiration de la signature au format timestamp UNIX
z=	Copie des noms et des valeurs des champs d'en-tête présents lors de la signature du message.

TABLE 3 – Valeurs possibles dans une en-tête DKIM d'un message

2.7 Le *Sender Policy Framework* (SPF)

Le *Sender Policy Framework* (SPF) permet de **restreindre les serveurs autorisés** à envoyer des messages dont l'adresse courriel expéditeur appartient à un domaine donné. Il est défini dans la RFC 7208.

En effet, il est possible d'envoyer un message provenant d'un domaine depuis un serveur SMTP d'un autre domaine. L'enregistrement SPF permet alors au MTA destinataire de vérifier si le MTA expéditeur est autorisé par le domaine expéditeur.

Cela a pour but de limiter les courriers indésirables et les usurpations d'identités depuis le domaine avec un enregistrement SPF.

Les informations sont saisies dans un enregistrement TXT sur le DNS du domaine *(fig. 14)*.

```
example.com. IN TXT "v=spf1 a mx -all"
example.fr. IN TXT "v=spf1 ip4:10.207.72.0/24 ip4:10.207.73.12 -all"
```

FIGURE 14 – Exemples d'enregistrements SPF

Un enregistrement SPF doit commencer par `v=spf1`. Voici les principaux mécanismes possibles (non exhaustif) *(table 4)* :

Clef	Libellé
`all`	Toutes les IP sont valides
`a`	Si l'IP fait partie des enregistrements A ou AAAA du domaine
`ip4`	Si l'IPv4 fait partie de la plage spécifiée
`ip6`	Si l'IPv6 fait partie de la plage spécifiée
`mx`	Si l'IP correspond à une IP de l'enregistrement MX du domaine
`exists`	Vérifie si le domaine peut être résolu

TABLE 4 – Principaux mécanismes d'un enregistrement SPF

Les modificateurs principaux sont à mettre avant le mécanisme. Le symbole + fait que si le mécanisme est validé, le courriel est approuvé. Il s'agit du comportement par défaut et est donc facultatif. Le symbole - rejette le mail correspondant au mécanisme : par exemple, le paramètre `-ip4:10.205.0.0/16` rejettera tous les courriels émis par les IP dans la plage 10.205.0.0/16.

Ainsi, le paramètre `-all` **rejettera tous les messages** ne correspondant pas aux règles précédentes.

2.8 Bonnes pratiques

Afin de limiter les risques de pertes de données et réduire l'impact environnemental et énergétique, voici quelques recommandations. N'hésitez pas à les partager auprès de vos utilisateurs :

— Laissez une copie des messages sur le serveur en cas de dysfonctionnement du stockage du client de messagerie.

— Supprimez les messages inutiles afin de réduire l'impact sur le stockage serveur et client, et ainsi limiter les besoins en dispositifs de stockage pour réduire l'impact environnemental et énergétique.

— Si une pièce jointe doit être envoyée à plusieurs destinataires, déposez le document sur un service de partage de fichiers et envoyer le lien pour y accéder. Auquel cas, la pièce jointe sera copiée autant de fois qu'il y a de destinataires.

— Limitez les signatures au strict nécessaire afin de réduire le poids de chaque message.

— Si vous répondez ou transférez un message, supprimez les échanges précédents si cela n'est pas nécessaire.

— N'imprimez un message que si cela est nécessaire.

— Lors de la création d'un compte sur un service en ligne, désactivez l'envoi de messages des partenaires commerciaux.

— Signez cryptographiquement les mails si cela est possible.

— N'ouvrez ni transférez les courriels vérolés (indésirables, frauduleux). Prenez garde à l'origine des messages.

— Sauvegardez les courriels de votre serveur, si possible sur un site de sauvegarde distant.

— Inscrivez-vous aux publications des préconisations d'une liste noire afin de limiter la réception de messages indésirables (Spamhaus, SORBS, etc.).

— Vérifiez régulièrement que votre serveur de messagerie ne soit pas émetteur de messages indésirables.

— Ouvrez une boîte de messagerie du type `abuse@domaine.tld` afin que puissent être signalés les envois de messages indésirables depuis vos serveurs.

2.9 Aspects juridiques

Divers textes de loi encadrent les communications électroniques.

Secret des correspondances

L'article 226-15 du Code Pénal punit **toute violation du secret des correspondances**, y compris des courriels :

> "Le fait, commis de mauvaise foi, d'ouvrir, de supprimer, de retarder ou de détourner des correspondances arrivées ou non à destination et adressées à des tiers, ou d'en prendre frauduleusement connaissance, est puni d'un an d'emprisonnement et de 45 000 euros d'amende.
> Est puni des mêmes peines le fait, commis de mauvaise foi, d'intercepter, de détourner, d'utiliser ou de divulguer des correspondances émises, transmises ou reçues par la voie électronique ou de procéder à l'installation d'appareils de nature à permettre la réalisation de telles interceptions.
> Lorsqu'ils sont commis par le conjoint ou le concubin de la victime ou le partenaire lié à la victime par un pacte civil de solidarité, ces faits sont punis d'une peine de deux ans d'emprisonnement et de 60 000 euros d'amende."
>
> *Source : https://www.legifrance.gouv.fr/codes/article_lc/LEGIARTI000042193573/2020-11-11*

Dans le milieu professionnel, le pourvoi n°99-42.942 du 2 octobre 2001 de la Cour de Cassation prévoit que si un employé **identifie clairement** un répertoire, un message ou des pièces jointes comme étant des correspondances **privées**, il ne peut pas violer le secret des correspondances, même si l'employeur interdit l'usage des équipements professionnels à cette fin.

En l'absence d'indications claires sur le caractère privé, l'employeur peut consulter les correspondances professionnelles.

Messages commerciaux

L'envoi de message commerciaux (publicités, prospection directe, *newsletters*, etc.) par courriel est encadré par l'article L34-5 du Code des Postes et des Communications Électroniques :

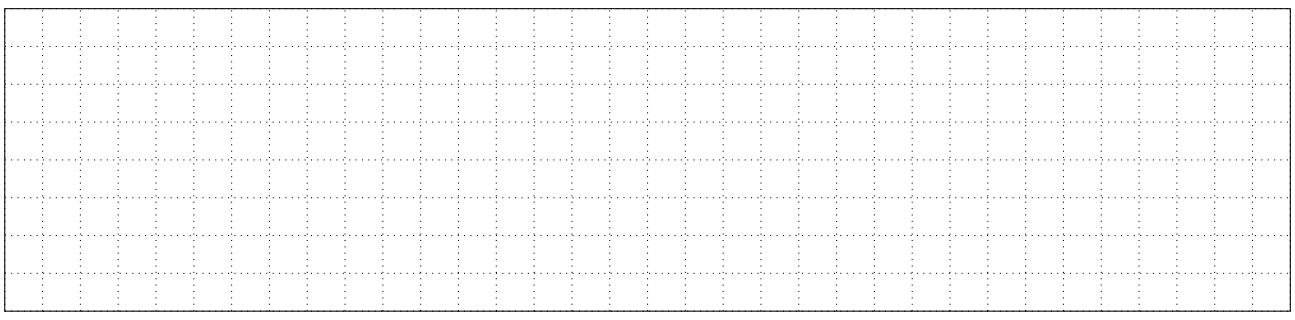
> "Est interdite la prospection directe au moyen de système automatisé de communications électroniques au sens du 6° de l'article L. 32, d'un télécopieur ou de courriers électroniques utilisant les coordonnées d'une personne physique, abonné ou utilisateur, qui n'a pas exprimé préalablement son consentement à recevoir des prospections directes par ce moyen.
> [...]
> Dans tous les cas, il est interdit d'émettre, à des fins de prospection directe, des messages au moyen de système automatisé de communications électroniques au sens du 6° de l'article L. 32, télécopieurs et courriers électroniques, sans indiquer de coordonnées valables auxquelles le destinataire puisse utilement transmettre une demande tendant à obtenir que ces communications cessent sans frais autres que ceux liés à la transmission de celle-ci. Il est également interdit de dissimuler l'identité de la personne pour le compte de laquelle la communication est émise et de mentionner un objet sans rapport avec la prestation ou le service proposé.
> [...]"
>
> *Source : https://www.legifrance.gouv.fr/codes/article_lc/LEGIARTI000042155961/2020-07-26*

Ne peuvent être envoyés des messages commerciaux **uniquement aux personnes l'ayant demandé** et elles **peuvent à tout moment résilier** leur abonnement.

Exercice 5

Écrire l'enregistrement SPF restreignant l'envoi de courriels du domaine `example.local` aux seuls serveurs suivants :

- Les enregistrements MX du domaine.
- La plage d'IP `172.25.32.0/19`
- Le serveur `10.56.217.118`
- Le serveur `fe80::dec7:47e8:e9b8:51c9`



Travaux pratiques

Les travaux pratiques, présentés ici, sont des mises en situation analogues aux demandes qui seront susceptibles de vous être proposées lors de votre vie professionnelle.

Vous serez progressivement mis en autonomie afin de rechercher par vous-mêmes les éléments nécessaires à la réalisation de vos demandes.

Le compte-rendu doit mentionner les diverses étapes qui vous ont menés au résultat et doit permettre à une tierce personne de mettre en œuvre et exploiter votre solution en se basant uniquement sur votre compte-rendu (tel une recette de cuisine). Il doit être aussi complet et clair que possible, il s'agit d'une documentation destinée à l'entreprise pour laquelle vous travaillez. Elle doit être réutilisable et permettre de comprendre les étapes que vous avez suivies.

Lorsque le choix d'une solution technologique vous est demandé, justifiez ces choix que vous avez faits. Ceux-ci doivent être étayés par des arguments factuels afin de répondre au besoin et assurer la maintenabilité de l'infrastructure que vous déployez.

Des documentations vous permettant de réaliser ces travaux pratiques sont disponibles en annexes. Lorsque vous utilisez des documentations provenant de sources externes, mentionnez ces sources. Justifiez votre choix.

Les travaux pratiques sont à réaliser sur la plateforme `https://rtaux.antoinepernot.fr`

Le barème se répartit comme suit :

— 10 points sur le bon fonctionnement de la solution proposée. Cette note est individuelle, c'est à dire que chacun doit réaliser les opérations sur la machine virtuelle fournie. Elle est déterminée en fonction du résultat des tests automatiques fournis.

— 10 points sur la qualité du rapport rendu. Cette note est commune au binôme de TP et sanctionne la clarté des explications, la pertinence des choix techniques et le niveau de détails proposés. La note se compose ainsi :

  — 5 points sur le bon fonctionnement de la solution documentée. Celle-ci doit être cohérente avec l'installation effectuée sur les machines virtuelles.

  — 2 points pour sanctionner la pertinence des choix techniques effectués et leur justification.

  — 2 points sur la clarté du rapport, sa lisibilité, le soin apporté à sa rédaction.

  — 1 point sur la présence de sources et la description du contexte dans le compte-rendu.

Sujet 1 : Le système DNS

La société Dubois import souhaite ouvrir une filiale d'exportation de tissus nommée "Lin de Normandie". Pour l'occasion, la société a acquis le nom de domaine `linnormandie.fr` et a dédié une partie de son parc informatique pour cette nouvelle entité. Les nouveaux serveurs seront les suivants :

Nom de domaine	Adresse IP	Usage
`linnormandie.fr`	`10.204.52.1`	Serveur Web
`linnormandie.fr`	`fe80::b46d:feaa:5cfe:36fd`	Serveur Web (IPv6)
`www.linnormandie.fr`	`10.204.52.1`	Sous-domaine du serveur Web
`dns.linnormandie.fr`	*Votre adresse IP*	Serveur DNS
`mail.linnormandie.fr`	`10.204.52.1`	Webmail pour les collaborateurs
`smtp.linnormandie.fr`	`10.204.52.2`	Serveur de messagerie (envoi)
`imap.linnormandie.fr`	`10.204.52.2`	Serveur de messagerie (consultation)
`support.linnormandie.fr`	`10.204.52.3`	Portail de support client

Le serveur DNS sera le seul serveur du domaine. Le TTL des enregistrements sera de 30 minutes. Le référent technique est `laurent.martin@linnormandie.fr`. Un enregistrement TXT pour le domaine sera à ajouter avec comme contenu :

```
v=spf1 a mx -all
```

Sujet 2 : Serveurs de messagerie

La société Constructions dijonnaises souhaite, pour des raisons budgétaires et de confidentialité, rapatrier ses serveurs de messagerie sur son infrastructure. Le partenaire, hébergeant jusqu'alors la messagerie, utilise les logiciels Postfix et Dovecot. Ce sont ces logiciels qui ont été retenus pour la réalisation de cette nouvelle infrastructure. Des protections contre les logiciels malveillants et les messages indésirables (reçus et émis) devront être mises en place.

Il a également été décidé, afin de faciliter l'utilisation de la messagerie, d'ajouter les deux fonctionnalités suivantes :

— L'authentification des utilisateurs se fera depuis le serveur LDAP de l'entreprise. Un compte en lecture seule devra être utilisé pour interroger le serveur LDAP.

— Afin d'empêcher la saturation du serveur par un utilisateur, un quota de 5 Go est imposé à toutes les boîtes de messagerie.

Le domaine est `constructions-dijonnaises.com`. Le serveur LDAP est hébergé sur `10.50.255.254`, le compte technique à utiliser est `cn=mail-ro,dc=constructions-dijonnaises,dc=com`. Le mot de passe de ce compte est `tprezo`.

La société utilise Mozilla Thunderbird comme client de messagerie. Déployez une solution permettant de faciliter la configuration de ce logiciel.

Le quota devra être géré par le plugin de quota de Dovecot. Le serveur IMAP devra écouter sur le port 993 et le serveur SMTP sur le port 465. Le chiffrement SSL/TLS est choisi. La question de la migration des messages hébergés par le partenaire vers la nouvelle infrastructure est à résoudre.

Les anciens courriels sont à télécharger sur `http://10.50.255.254/export-mails.tar.gz`

Annexes

Serveur DNS Bind9

Sources de la documentation :

— *https://blog.foulquier.info/tutoriels/systeme/installation-et-parametrage-d-un-resolveur-dns-avec-bind-9-sur-debian-7*

Procédure validée sur Debian 12.

Documentation disponible en ligne : https://antoinepernot.fr/articles/installation-dns

Présentation du projet

Nous souhaitons installer un serveur DNS pour la zone réseau `example.com`. Dans cette documentation, le nom de domaine du serveur est `ns.example.com` et a pour IP 10.42.0.1. Nous le paramétrerons également afin d'être un serveur DNS récursif pour des clients sur le réseau. Nous utiliserons le logiciel Bind9.

Pré-requis

Modifier le nom d'hôte dans le fichier **/etc/hosts** :

```
127.0.0.1       localhost
127.0.1.1       ns.example.com    ns
```

Modifier également le nom d'hôte dans le fichier **/etc/hostname** :

```
ns.example.com
```

Nous allons installer les paquets Bind9 :

```
apt -y install bind9 dnsutils nftables
```

Autoriser le trafic vers le serveur DNS et en loopback en modifiant le fichier **/etc/nftables.conf**. Adapter selon les autres services déjà installés :

Pensez à créer une règle pour SSH si vous utilisez ce service pour configurer votre serveur. Ne vous enfermez pas dehors !

```
#!/usr/sbin/nft -f

flush ruleset

table inet tableinet {
   chain input {
      type filter hook input priority filter; policy drop;
```

```
        iifname lo accept
        udp dport 53 accept
        ct state {established, related} accept
    }
    chain forward {
        type filter hook forward priority filter;
    }
    chain output {
        type filter hook output priority filter;
    }
}
```

Activer et redémarrer nftables :

```
systemctl enable nftables.service
systemctl restart nftables.service
```

Modifier le résolveur dans le fichier **/etc/resolv.conf** :

```
domain example.com
search example.com
nameserver 10.42.0.1
```

Création de la zone de résolution

Créer le fichier **/etc/bind/db.example.com** suivant cet exemple :

```
;
; BIND data file for local loopback interface
;
$TTL  604800
@  IN SOA   ns.example.com. root.example.com. (2020120401 604800
   86400 2419200 604800)
;
@  IN NS ns.example.com.
@  IN A  10.42.0.1
ns IN A  10.42.0.1
www   IN A  10.42.0.2
www   IN AAAA  fe80::74cf:64ad:6cfa:9604
mail  IN CNAME www
```

Création de la zone de résolution inverse

Si vous possédez des enregistrements IPv6, il est nécessaire de faire deux zones de résolutions inverses distinctes IPv4 et IPv6.

Créer le fichier **/etc/bind/db.example.com.inv** suivant cet exemple :

```
;
; BIND reverse data file for local loopback interface
;
$TTL  604800
@  IN SOA   ns.example.com. root.example.com. (2020120401 604800
   86400 2419200 604800)
;
@  IN NS ns.example.com.
1  IN PTR   ns.example.com.
2  IN PTR   www.example.com.
```

Créer le fichier pour la zone IPv6 **/etc/bind/db.example.com.inv6** suivant cet exemple :

```
;
; BIND reverse data file for local loopback interface
;
$TTL  604800
@  IN SOA   ns.example.com. root.example.com. (2020120401 604800
   86400 2419200 604800)
;
@  IN NS    ns.example.com.
4.0.6.9.a.f.c.f.d.a.4.6.f.c.4.7.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.
   arpa.  IN PTR   www.example.com.
```

Paramétrage des zones

Modifier le fichier **/etc/bind/named.conf.local** afin de paramétrer les zones :

```
zone "example.com" {
        type master;
        file "/etc/bind/db.example.com";
        forwarders{};
};

zone "0.42.10.in-addr.arpa" {
        type master;
        file "/etc/bind/db.example.com.inv";
        forwarders{};
};

zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa" {
        type master;
        file "/etc/bind/db.example.com.inv6";
        forwarders{};
};
```

Activer la résolution récursive

Modifier le fichier **/etc/bind/named.conf.options** :

```
options {
        directory "/var/cache/bind";
        // forwarders {
        //      0.0.0.0;
        // };
        dnssec-validation auto;
        auth-nxdomain no;    # conform to RFC1035
        listen-on-v6 { any; };
        allow-recursion { any; };
};
```

Redémarrer le service :

```
service bind9 restart
```

Tester la résolution :

```
dig @127.0.0.1 www.example.com
dig @127.0.0.1 antoinepernot.fr
```

Serveur mail Postfix et Dovecot

Sources de la documentation :

- *https://vorkbaard.nl/installing-a-mailserver-on-debian-8-part-1-introduction/*
- *https://www.tecmint.com/setup-postfix-mail-server-and-dovecot-with-mariadb-in-centos/*
- *https://www.badsender.com/2014/01/13/delivrabilite-spf-dkim-dmarc/*
- *https://easyengine.io/tutorials/mail/dkim-postfix-ubuntu*
- *https://blog.debugo.fr/serveur-messagerie-complet-postfix-dovecot-ldap-rspamd/*
- *https://www.laintimes.com/configurer-postfix-en-relay-avec-ovh/*

Procédure validée sur Debian 12.

Documentation disponible en ligne : https://antoinepernot.fr/articles/installation-mail

Présentation du projet

Nous souhaitons installer un serveur SMTP Postfix et un serveur IMAP Dovecot. Nous utiliserons SpamAssassin pour lutter contre les courriers indésirables. Nous configurerons également notre serveur afin d'être validé et reconnu comme émetteur de mails valides. Les comptes utilisateurs de notre serveur mail seront consignés soit dans une base MariaDB, soit dans un serveur OpenLDAP afin de permettre une plus grande souplesse de configuration.

Cette documentation est destinée à la fois aux personnes souhaitant installer leur serveur en réseau local ou pour un site Internet. Si vous êtes dans le premier cas, ignorez les étapes concernant l'installation d'un serveur public.

Pré-requis

Nous allons utiliser une base LAMP pour notre serveur. Pour cela, nous installons les paquets suivants :

```
apt -y install apache2
```

Nous allons configurer Apache. Pour cela, éditez le fichier **/etc/apache2/sites-available/000-default.conf** et éditez la ligne suivante :

```
#ServerName www.example.com
```

Dé-commentez la et replacez le nom de domaine par celui sur lequel vous installez votre serveur mail :

```
ServerName example.com
```

Créez le fichier **/etc/apache2/sites-available/thunderbird-autoconfig.conf** comme suit :

```
<VirtualHost *:80>
ServerName autoconfig.example.com
DocumentRoot /var/www/html/autoconfig/
<Directory /var/www/html/autoconfig/>
        Options -Indexes +FollowSymLinks +MultiViews -Includes -
           ExecCGI
        AllowOverride None
        Order allow,deny
        allow from all
</Directory>
</VirtualHost>
```

Activez le site et rechargez la configuration Apache :

```
a2ensite thunderbird-autoconfig
service apache2 reload
```

Configurez les entrées suivantes sur le DNS de votre nom de domaine en remplaçant le domaine et l'IP :

```
example.com. IN A 1.2.3.4
example.com. IN MX 10 example.com.
autoconfig.example.com IN CNAME example.com.
example.com. IN TXT "v=spf1 a mx -all"
```

Pour un serveur public

Si votre serveur mail est destiné à être utilisé sur Internet, vous devez disposer d'un nom de domaine et d'une IP fixe. Assurez-vous que les ports TCP 25, 80, 443, 465 et 993 sont ouverts et que vous pouvez y accéder depuis Internet.

Nous allons installer Let's Encrypt afin d'obtenir un certificat SSL reconnu. Pour cela, exécutez les commandes suivantes et suivez l'assistant en choisissant le mode `Secure` qui redirige tout trafic HTTP sur HTTPS :

```
apt -y install python-certbot-apache
certbot --authenticator webroot --installer apache --webroot-path /
   var/www/html -d example.com
```

Éditez la table cron de l'utilisateur root et renseignez la ligne suivante afin de demander automatiquement le renouvellement du certificat tous les dimanches :

```
0 3 * * 0 certbot renew && systemctl restart dovecot.service
```

Configurez l'entrée suivante sur le DNS de votre FAI en remplaçant le domaine et l'IP :

```
4.3.2.1.in-addr.arpa. IN PTR example.com
```

Installation des paquets

Nous allons ici installer tout ce qui est nécessaire à notre serveur mail :

Postfix Serveur SMTP en charge du transfert des mails.

Dovecot Serveur IMAP en charge de fournir les mails aux clients de messagerie.

SpamAssassin Filtre anti-spam

ClamAV Antivirus

OpenDKIM Outil de génération de la clef DKIM

```
apt -y install postfix dovecot-core dovecot-imapd dovecot-lmtpd
   dovecot-managesieved dovecot-sieve spamassassin spamc clamav-
   daemon clamav clamsmtp opendkim opendkim-tools rsyslog mailutils
   nftables
```

Lors de l'installation, Postfix vous demande les questions suivantes :

Configuration type du serveur de messagerie Site Internet

Domaine *Votre domaine*

Autoriser le trafic vers le serveur mail et en loopback en modifiant le fichier **/etc/nftables.conf**. Adapter selon les autres services déjà installés :

Pensez à créer une règle pour SSH si vous utilisez ce service pour configurer votre serveur. Ne vous enfermez pas dehors !

```
#!/usr/sbin/nft -f

flush ruleset

table inet tableinet {
   chain input {
      type filter hook input priority filter; policy drop;
      iifname lo accept
      tcp dport 25 accept
      tcp dport 993 accept
      tcp dport 465 accept
      tcp dport 80 accept
      ct state {established,related} accept
   }
   chain forward {
      type filter hook forward priority filter;
   }
   chain output {
      type filter hook output priority filter;
   }
}
```

Activer et redémarrer nftables :

```
systemctl enable nftables.service
systemctl restart nftables.service
```

Initialisation de la base de données / de l'annuaire

Configuration avec MariaDB

Le serveur Postfix gère par défaut les utilisateurs en provenance du système d'authentification Unix. Nous souhaitons utiliser à la place des utilisateurs virtuels stockés dans la base de données. Pour cela, nous allons créer la base de données stockant les comptes utilisateurs et les alias.

Installez les paquets supplémentaires :

```
apt -y install postfix-mysql dovecot-mysql mariadb-server
```

Connectez-vous à la base de données MariaDB :

```
mysql -u root -p
```

Saisissez les commandes suivantes afin de créer la base de données :

```
CREATE DATABASE postfix;
USE postfix;

CREATE TABLE addresses (
    email VARCHAR(50) NOT NULL PRIMARY KEY,
    active TINYINT(1) NOT NULL DEFAULT 1,
    passwd VARCHAR(106) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE aliases (
    source VARCHAR(50) NOT NULL PRIMARY KEY,
    target VARCHAR(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE USER 'postfix'@'localhost' IDENTIFIED BY '
    MonMotDePasseBaseDeDonnees';
GRANT SELECT ON postfix.addresses TO 'postfix'@'localhost';
GRANT SELECT ON postfix.aliases TO 'postfix'@'localhost';
```

Nous allons créer un premier utilisateur comme suit :

```
INSERT INTO postfix.addresses (email, active, passwd) VALUES ("
    antoine@example.com", 1, ENCRYPT('MotDePasseEmail', CONCAT('$6$',
     SUBSTRING(SHA(RAND()), -16))));
```

Nous allons créer un premier alias comme suit :

```
INSERT INTO postfix.aliases (source, target) VALUES ("
    apernot@example.com", "antoine@example.com");
```

Configuration avec OpenLDAP

Pour la création d'un annuaire OpenLDAP, reportez-vous à la documentation spécifique : `https://www.antoinepernot.fr/articles/installation-openldap-nfs`.

Installez les paquets supplémentaires :

```
apt -y install postfix-ldap dovecot-ldap
```

Pour la suite de cette documentation, les utilisateurs seront stockés dans l'OU `ou=Utilisateurs,dc=example,dc=com` avec la structure suivante :

```
dn: cn=pdubois,ou=Utilisateurs,dc=example,dc=com
objectClass: top
objectClass: inetOrgPerson
objectClass: posixAccount
cn: Paul Dubois
cn: pdubois
userPassword:: <hash du mot de passe>
sn: pdubois
gidNumber: 1100
homeDirectory: /home/pdubois
uid: pdubois
mail: pdubois@example.com
uidNumber: 1200
```

Configuration de Postfix

Inscrivez votre domaine dans le fichier **/etc/mailname**.

Créez l'utilisateur gérant les utilisateurs virtuels et le répertoire stockant les mails :

```
useradd -d /var/mail -U -u 5000 vmail
mkdir -p /var/mail/vmail/example.com
chown -R vmail:vmail /var/mail/vmail
```

Éditez le fichier **/etc/postfix/main.cf**. Nous effectuerons les modifications suivantes :

Saisissez votre nom de domaine dans le champ `myhostname` :

```
myhostname = example.com
```

Retirez votre domaine du champ `mydestination`. Il est en effet impossible qu'un domaine soit affecté à des utilisateurs réels et virtuels à la fois.

Éditez la configuration propre à la gestion des domaines virtuels et la sécurité (modifier le chemin du certificat SSL si nécessaire) :

```
smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff = no
append_dot_mydomain = no
readme_directory = no
compatibility_level = 3.6

smtpd_tls_cert_file=/etc/letsencrypt/live/example.com/fullchain.pem
smtpd_tls_key_file=/etc/letsencrypt/live/example.com/privkey.pem
smtpd_tls_security_level=may

smtp_tls_CApath=/etc/ssl/certs
smtp_tls_security_level=may
smtp_tls_session_cache_database = btree:${data_directory}/
   smtp_scache
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/
   smtpd_scache

smtpd_relay_restrictions = permit_mynetworks
   permit_sasl_authenticated defer_unauth_destination
myhostname = example.com
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = localhost
relayhost =
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all

virtual_mailbox_domains = example.com
virtual_mailbox_base = /var/mail/vmail
virtual_gid_maps = static:5000
virtual_uid_maps = static:5000
virtual_minimum_uid = 5000
virtual_mailbox_maps = mysql:/etc/postfix/mysql_virtual_mailbox_maps
   .cf
virtual_alias_maps = mysql:/etc/postfix/mysql_virtual_aliases.cf
virtual_transport = lmtp:unix:private/dovecot-lmtp
smtpd_sasl_auth_enable = yes
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_tls_auth_only = yes

# Signature OpenDKIM
milter_default_action = accept
milter_protocol = 2
```

```
smtpd_milters = inet:localhost:8892
non_smtpd_milters = inet:localhost:8892

# ClamAV
strict_rfc821_envelopes = yes
disable_vrfy_command = yes
smtpd_helo_required = yes
smtpd_client_restrictions =
smtpd_helo_restrictions =
smtpd_sender_restrictions =

content_filter = scan:localhost:10026
```

Pour une configuration utilisant LDAP, modifiez les lignes suivantes :

```
virtual_mailbox_maps = ldap:/etc/postfix/ldap_virtual_mailbox_maps.
    cf
virtual_alias_maps = ldap:/etc/postfix/ldap_virtual_aliases.cf
```

Modifiez les lignes suivantes dans le fichier **/etc/postfix/master.cf** en prenant garde à ce que les options (commençant par -o) doivent être précédées par au moins un espace :

```
smtp      inet  n       -       y       -       -       smtpd
  -o content_filter=spamassassin
submission inet n       -       y       -       -       smtpd
  -o syslog_name=postfix/submission
  -o smtpd_tls_security_level=encrypt
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
  -o smtpd_relay_restrictions=permit_sasl_authenticated,reject
smtps     inet  n       -       y       -       -       smtpd
  -o syslog_name=postfix/smtps
  -o smtpd_tls_wrappermode=yes
  -o smtpd_sasl_auth_enable=yes
pickup    unix  n       -       y       60      1       pickup
cleanup   unix  n       -       y       -       0       cleanup
qmgr      unix  n       -       n       300     1       qmgr
tlsmgr    unix  -       -       y       1000?   1       tlsmgr
rewrite   unix  -       -       y       -       -       trivial-
    rewrite
bounce    unix  -       -       y       -       0       bounce
defer     unix  -       -       y       -       0       bounce
trace     unix  -       -       y       -       0       bounce
verify    unix  -       -       y       -       1       verify
flush     unix  n       -       y       1000?   0       flush
proxymap  unix  -       -       n       -       -       proxymap
proxywrite unix -       -       n       -       1       proxymap
smtp      unix  -       -       y       -       -       smtp
relay     unix  -       -       y       -       -       smtp
        -o syslog_name=postfix/$service_name
```

```
showq     unix  n       -       y       -       -       showq
error     unix  -       -       y       -       -       error
retry     unix  -       -       y       -       -       error
discard   unix  -       -       y       -       -       discard
local     unix  -       n       n       -       -       local
virtual   unix  -       n       n       -       -       virtual
lmtp      unix  -       -       y       -       -       lmtp
anvil     unix  -       -       y       -       1       anvil
scache    unix  -       -       y       -       1       scache
postlog   unix-dgram n  -       n       -       1       postlogd

maildrop  unix  -       n       n       -       -       pipe
  flags=DRXhu user=vmail argv=/usr/bin/maildrop -d ${recipient}
uucp      unix  -       n       n       -       -       pipe
  flags=Fqhu user=uucp argv=uux -r -n -z -a$sender - $nexthop!rmail
     ($recipient)
ifmail    unix  -       n       n       -       -       pipe
  flags=F user=ftn argv=/usr/lib/ifmail/ifmail -r $nexthop (
     $recipient)
bsmtp     unix  -       n       n       -       -       pipe
  flags=Fq. user=bsmtp argv=/usr/lib/bsmtp/bsmtp -t$nexthop -
     f$sender $recipient
scalemail-backend unix -       n       n       -       2       pipe
  flags=R user=scalemail argv=/usr/lib/scalemail/bin/scalemail-store
      ${nexthop} ${user} ${extension}
mailman   unix  -       n       n       -       -       pipe
  flags=FRX user=list argv=/usr/lib/mailman/bin/postfix-to-mailman.
     py ${nexthop} ${user}
spamassassin unix -     n       n       -       -       pipe
  user=debian-spamd argv=/usr/bin/spamc -f -e /usr/sbin/sendmail -oi
     -f ${sender} ${recipient}

# Antivirus
scan      unix  -       -       n       -       16      smtp
   -o smtp_send_xforward_command=yes
   -o disable_dns_lookups=yes
   -o smtp_data_done_timeout=1200

# For injecting mail back into postfix from the filter
localhost:10025 inet  n -       n       -       16      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_style=host
   -o smtpd_authorized_xforward_hosts=127.0.0.0/8
```

Configuration afin d'utiliser un relais SMTP

Un relais SMTP permet de faire transiter les messages émis sur un serveur relais afin que ces messages soient vus par le serveur destinataire comme émis par le relais. Cela est utile lorsque votre serveur ne peut être reconnu comme serveur de confiance (adresse IP publique dynamique, reverse DNS incorrect, etc.). Un compte valide sur un service mail permettant le relais SMTP sera nécessaire.

Modifiez le fichier **/etc/postfix/main.cf** comme suit :

Supprimez le domaine du champ `mydestination` autre que localhost et ajoutez les lignes suivantes :

```
relayhost=[mail.servicemail.net]:587
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl/sasl_passwd
smtp_sasl_security_options = noanonymous
```

Configurez le compte mail du relais dans le fichier **/etc/postfix/sasl/sasl_passwd** :

```
[mail.servicemail.net]:587 relais-smtp@servicemail.net:
    MotDePasseCompteRelaisSMTP
```

Générez le hash SASL :

```
postmap /etc/postfix/sasl/sasl_passwd
```

Configuration avec MariaDB

Créez le fichier **/etc/postfix/mysql_virtual_mailbox_maps.cf** et saisissez la configuration :

```
user = postfix
password = MonMotDePasseBaseDeDonnees
hosts = 127.0.0.1
dbname = postfix
query = SELECT 1 FROM addresses WHERE email = '%s'
```

Créez le fichier **/etc/postfix/mysql_virtual_aliases.cf** et saisissez la configuration :

```
user = postfix
password = MonMotDePasseBaseDeDonnees
hosts = 127.0.0.1
dbname = postfix
query = SELECT 1 FROM aliases WHERE source = '%s'
```

Testez en saisissant la commande suivante :

```
postmap -q antoine@example.com mysql:/etc/postfix/
    mysql_virtual_mailbox_maps.cf
```

Si vous obtenez 1, la configuration est correcte.

Configuration avec OpenLDAP

Créez le fichier **/etc/postfix/ldap_virtual_mailbox_maps.cf** et saisissez la configuration :

```
server_host = ldap://127.0.0.1
version = 3
bind = yes
bind_dn = cn=lecteur,dc=example,dc=com
bind_pw = MotDePasseLectureSeule
search_base = ou=Utilisateurs,dc=example,dc=com
scope = sub
query_filter = (&(mail=%s)(objectClass=inetOrgPerson))
result_attribute = mail
```

Créez le fichier `/etc/postfix/ldap_virtual_aliases.cf` et saisissez la configuration :

```
server_host = ldap://127.0.0.1
version = 3
bind = yes
bind_dn = cn=lecteur,dc=example,dc=com
bind_pw = MotDePasseLectureSeule
search_base = ou=Utilisateurs,dc=example,dc=com
scope = sub
query_filter = (&(mail=%s)(objectClass=alias))
result_attribute = mail
```

Testez en saisissant les commandes suivantes :

```
service clamav-daemon restart
service clamsmtp restart
service postfix restart
postmap -q pdubois@example.com ldap:/etc/postfix/
   ldap_virtual_mailbox_maps.cf
```

Si vous obtenez l'adresse mail demandée, la configuration est correcte.

Configuration d'OpenDKIM

Modifiez le fichier **/etc/opendkim.conf** :

```
Domain                  example.com
KeyFile                 /etc/postfix/dkim/mail.private
Selector                mail
Socket                  inet:8892@localhost
```

Dans le fichier **/etc/default/opendkim**, modifiez la ligne suivante :

```
SOCKET="inet:8892@localhost"
```

Créez et allez dans le dossier accueillant la clef, puis générez la :

```
mkdir /etc/postfix/dkim/
cd /etc/postfix/dkim/
opendkim-genkey -t -s mail -d example.com
```

Configurez une entrée TXT sur une seule ligne (ou DKIM) dans votre DNS avec le contenu du fichier **/etc/postfix/dkim/mail.txt**.

Configuration de Dovecot

Configurez le serveur Dovecot. Pour cela, nous allons modifier les lignes suivantes dans le fichier **/etc/dovecot/conf.d/10-mail.conf** :

```
mail_location = maildir:/var/mail/vmail/%d/%n
mail_privileged_group = vmail
mail_uid = 5000
mail_gid = 5000
```

Modifiez le fichier **/etc/dovecot/conf.d/10-master.conf** :

```
service imap-login {
   inet_listener imap {
   }
   inet_listener imaps {
      port = 993
      ssl = yes
   }
}

service lmtp {
    unix_listener /var/spool/postfix/private/dovecot-lmtp {
        mode = 0600
        user = postfix
        group = postfix
    }
}

service auth {
    unix_listener /var/spool/postfix/private/auth {
        mode = 0666
        user = postfix
        group = postfix
    }

    #unix_listener auth-userdb {
     #mode = 0666
     #user =
     #group =
   #}
}
```

Modifiez les chemins vers les certificats SSL dans le fichier **/etc/dovecot/conf.d/10-ssl.conf** (ajuster les chemins vers les certificats SSL si nécessaire) :

```
ssl_cert = </etc/letsencrypt/live/example.com/fullchain.pem
ssl_key = </etc/letsencrypt/live/example.com/privkey.pem
```

Modifiez la configuration du fichier **/etc/dovecot/conf.d/15-lda.conf** :

```
protocol lda {
  mail_plugins = $mail_plugins sieve
}
```

Modifiez la configuration du fichier **/etc/dovecot/conf.d/15-mailboxes.conf** :

```
mailbox Drafts {
  special_use = \Drafts
  auto = subscribe
}
mailbox Junk {
  special_use = \Junk
  auto = subscribe
}
mailbox Trash {
  special_use = \Trash
  auto = subscribe
}
mailbox Sent {
  special_use = \Sent
  auto = subscribe
}
```

Modifiez la configuration du fichier **/etc/dovecot/conf.d/20-lmtp.conf** :

```
protocol lmtp {
  mail_plugins = "sieve"
  postmaster_address = pdubois@example.com
}
```

Ajoutez la ligne suivante au fichier **/etc/dovecot/conf.d/90-sieve.conf** :

```
sieve_after = /etc/dovecot/sieve/spamfilter.sieve
```

Générez les paramètres DH :

```
openssl dhparam 2048 > /etc/dovecot/dh.pem
```

Ajoutez la ligne suivante au fichier **/etc/dovecot/conf.d/10-ssl.conf** :

```
ssl_dh=</etc/dovecot/dh.pem
```

Configuration avec MariaDB

Le réglage des méthodes d'authentification se fait en modifiant le fichier **/etc/dovecot/conf.d/10-auth.conf** :

```
auth_mechanisms = plain login
#!include auth-system.conf.ext
!include auth-sql.conf.ext
```

Renseignez les requêtes utilisées par Dovecot pour rechercher les utilisateurs virtuels dans la base de données en éditant le fichier **/etc/dovecot/dovecot-sql.conf.ext** :

```
driver = mysql
connect = host=127.0.0.1 dbname=postfix user=postfix password=
   MonMotDePasseBaseDeDonnees
default_pass_scheme = SHA512-CRYPT
password_query = \
   SELECT email as username, passwd AS password FROM addresses WHERE
       email = '%u'
user_query = \
   SELECT 5000 AS uid, 5000 as gid, email, '/var/mail/vmail/%d/%n'
      AS home \
   FROM addresses WHERE email = '%u'
iterate_query = SELECT email AS user FROM addresses
```

Configuration avec OpenLDAP

Le réglage des méthodes d'authentification se fait en modifiant le fichier **/etc/dovecot/conf.d/10-auth.conf** :

```
auth_mechanisms = plain login
#!include auth-system.conf.ext
!include auth-ldap.conf.ext
```

Renseignez les requêtes utilisées par Dovecot pour rechercher les utilisateurs virtuels dans la base de données en éditant le fichier **/etc/dovecot/dovecot-ldap.conf.ext** :

```
uris = ldap://127.0.0.1
dn = cn=lecteur,dc=example,dc=com
dnpass = MotDePasseLectureSeule
debug_level = 0
auth_bind = no
ldap_version = 3
base = ou=Utilisateurs,dc=example,dc=com
scope = subtree

user_attrs = homeDirectory=home
user_filter = (&(|(uid=%u)(mail=%u))(objectClass=inetOrgPerson))

pass_attrs = mail=user,userPassword=password
pass_filter = (&(|(uid=%u)(mail=%u))(objectClass=inetOrgPerson))
```

Redémarrez Dovecot et testez la connexion avec Dovecot :

```
service dovecot restart
doveadm auth test -x service=imap pdubois@example.com
```

Configuration de SpamAssassin

Si, au lieu de mettre les spam en pièce-jointe, vous souhaitez simplement les étiqueter comme tel, modifiez la ligne suivante dans le fichier **/etc/spamassassin/local.cf** :

```
report_safe 0
```

Activez le service au démarrage :

```
systemctl enable spamd.service
```

Configuration du système d'auto-configuration Thunderbird

Créez le fichier **/var/www/html/autoconfig/mail/config-v1.1.xml** :

```
<?xml version="1.0" encoding="UTF-8"?>
<clientConfig version="1.1">
  <emailProvider id="example.com">
    <domain>example.com</domain>
    <displayName>Mon domaine</displayName>
    <displayShortName>Mon domaine</displayShortName>
    <incomingServer type="imap">
      <hostname>example.com</hostname>
      <port>993</port>
      <socketType>SSL</socketType>
      <authentication>password-cleartext</authentication>
      <username>%EMAILADDRESS%</username>
    </incomingServer>
    <outgoingServer type="smtp">
      <hostname>example.com</hostname>
      <port>465</port>
      <socketType>SSL</socketType>
      <authentication>password-cleartext</authentication>
      <username>%EMAILADDRESS%</username>
    </outgoingServer>
  </emailProvider>
</clientConfig>
```

Démarrage et tests du service

Redémarrez les services :

```
service apache2 restart
service opendkim restart
service postfix restart
```

```
service dovecot restart
service spamd restart
service clamsmtp restart
service clamav-daemon restart
```

Nous allons tester l'envoi d'un mail :

```
echo "Ceci est mon premier mail" | mailx -s "Hello world"
    pdubois@example.com
```

Si un fichier mail correspondant au message envoyé est présent dans le répertoire **/var/mail/vmail/example.com/pdubois/new**, la configuration est correcte.

Memento Shell

Auteur : Antoine Pernot

• **Gestion des fichiers/répertoires**

Créer un répertoire (make directory) :

`-p` : Créer le répertoire parent si besoin.

```
mkdir [-p] rep
```

Créer un fichier :

```
touch fich
```

Changer de répertoire (change dir) :

```
cd rep
cd ..      # repertoire parent
cd         # repertoire personnel
cd ~alice # repertoire personnel de alice
```

Afficher répertoire courant (print working dir) :

```
pwd
```

Copier un fichier/répertoire vers un autre :

`-r` : Copier le répertoire de manière récursive.

```
cp [-r] orig dest
```

Créer un lien :

`-s` : Créer un lien symbolique.

```
ln [-s] orig lien
```

Déplacer/renommer un fichier/répertoire vers un autre (move) :

```
mv orig dest
```

Supprimer un fichier (remove) :

`-r` : Supprime un répertoire de manière récursive.

```
rm [-r] fich
```

Supprimer un répertoire vide (remove directory) :

```
rmdir rep
```

Lister le contenu d'un répertoire :

`-l` : Liste détaillée.
`-a` : Liste tous les fichiers (inclut les fichiers cachés).
`-s` : Tri par taille.
`-t` : Tri par date.
`-r` : Inverse l'ordre de tri.

```
ls [-latsr] rep
```

• **Gestion du contenu des fichiers/des flux**

Affichage brut de fichiers (non-interactif) :

```
cat [fich1 [fich2 ...]]
```

Affichage interactif de fichiers :

```
more [fich1 [fich2 ...]] # sens unique
less [fich1 [fich2 ...]] # 2 sens
```

Afficher le début d'un fichier :

`-n` : Affiche n lignes.

```
head [-n=10] [fich]
```

Afficher la fin d'un fichier :

`-n` : Affiche n lignes.

```
tail [-n=10] [fich]
```

Rechercher dans un fichier :

`-i` : Insensible à la casse.
`-v` : Affiche les lignes ne correspondant PAS à l'expression.

```
grep [-iv] expression [fich]
```

Trie les lignes d'un fichier :

`-r` : Inverse l'ordre.
`-R` : Trie dans un ordre aléatoire.

```
sort [-rR] [fich]
```

Compte les éléments d'un fichier :

`-l` : Compte le nombre de lignes.
`-w` : Compte le nombre de mots.
`-c` : Compte le nombre de caractères.

```
wc [-clw] [fich]
```

Découper un flux de texte :

`-d` : Délimiteur de découpe.
`-f` : Sélectionne les champs à renvoyer.

```
cut [-df] [fich]
```

• **Droits d'accès aux fichiers**

Changer les droits d'accès aux fichiers :

`-R` : Change les droits de manière récursive.

```
chmod [-R] {ugoa}{+-}{rwx} fich
```

Changer le propriétaire du fichier :

`-R` : Change le propriétaire de manière récursive.

```
chown [-R] nvuser[:nvgrp] fich
```

Changer le groupe du fichier :

`-R` : Change le groupe de manière récursive.

```
chgrp [-R] nvgrp fich
```

• **Recherche de fichiers**

Rechercher un fichier :

`-exec` : Exécute une commande en replaçant {} par le chemin de chacun des fichiers trouvés.

```
find rep_rech -name regex [-exec cmd {} ';']
```

• **Gestion des flux de texte**

Rediriger la sortie d'une commande vers l'entrée d'une autre :

```
cat villes.txt | grep Dijon
```

Écrire la sortie d'une commande dans un fichier (écrase le contenu) :

```
grep Dijon villes.txt > info_Dijon.txt
```

Ajoute la sortie d'une commande à un fichier :

```
grep Dijon villes.txt >> info_Dijon.txt
```

Ajoute la sortie d'erreurs d'une commande à un fichier :

```
grep Dijon villes.txt 2>> erreurs_villes.txt
```

• **Contrôle de tâches**

Affiche les processus en cours d'exécution :

```
ps aux
top     # Mode interactif
```

Envoie un signal (d'arrêt) au processus :

`-9` : Envoie un signal d'arrêt SIGKILL

```
kill [-9] pid
```

Envoie un signal (d'arrêt) aux processus appelés "nom" :

```
killall nom
```

• **Aide système**

Lister les pages de manuel contenant une chaîne de caractère :

```
apropos chaine
```

Afficher la page de manuel d'une commande :

```
man cmd
```

Memento scripts

Auteur : Antoine Pernot

- **Structure de base**

```
#!/bin/bash
# Version du script
echo "Bonjour"
exit 0
```

- **Les variables**

Affecter une variable :

```
message="Coucou !"
read rep # Stocke la reponse utilisateur
```

Appeler une variable :

```
echo $message
```

Appeler un fragment de chaîne de caractères :

```
echo ${message:offset:nbchars}
```

Variables spéciales :

$*	Contient tous les arguments passés à la fonction.
$#	Contient le nombre d'argument.
$?	Contient le code de retour de la dernière opération.
$0	Contient le nom du script.
$n	Contient l'argument n.
$!	Contient le PID de la dernière commande lancée.

- **Les tableaux**

Affectation (1ère méthode) :

```
tab=(valeur1 valeur2 ...)
```

Affectation (2ème méthode) :

```
tab[0]=John Smith
tab[1]=Jane Doe
```

Compter le nombre d'éléments du tableau :

```
len=${#tab[*]}
```

Afficher un élément :

```
echo ${tab[1]}
```

Afficher tous les éléments :

```
echo ${tab[@]}
```

- **Les structures de contrôle**

Syntaxe :

```
[ -f fichier ]
```

Opérateurs de test :

-e fichier	Contrôle si fichier existe.
-d fichier	Contrôle si fichier existe et est un répertoire.
-f fichier	Contrôle si fichier existe et est un fichier 'normal'.
-w fichier	Contrôle si fichier existe et est en écriture.
-x fichier	Contrôle si fichier existe et est exécutable.
f1 -nt f2	Contrôle si f1 est plus récent que f2.
f1 -ot f2	Contrôle si f1 est plus vieux que f2.

Opérateurs de comparaison numériques :

$n1 -eq $n2	Vérifie si les nombres sont égaux. Utiliser = pour les chaînes de caractères.
$n1 -ne $n2	Vérifie si les nombres sont différents. Utiliser != pour les chaînes de caractères.
$n1 -lt $n2	Vérifie si n1 est inférieur à n2.
$n1 -le $n2	Vérifie si n1 est inférieur ou égal à n2.
$n1 -gt $n2	Vérifie si n1 est supérieur à n2.
$n1 -ge $n2	Vérifie si n1 est supérieur ou égal à n2.

Les opérateurs logiques :

expr1 -a expr2 ou expr1 && expr2	Opérateur ET.
expr1 -o expr2 ou expr1 || expr2	Opérateur OU.
! expr	Opérateur NON.

- **Les conditions**

```
if [ condition1 ]
then
    Condition 1 vraie
elif [ condition2 ]
then
    Condition 2 vraie
else
    Aucune condition vraie
fi
```

- **Les boucles**

"Tant que ..." :

```
while [ condition ]
do
     Tant que la condition est vraie
done
```

"Pour ..." :

```
for variable in valeurs
do
    instructions
done
```

- **Les aiguillages**

```
case "$var" in
    val1 | val2 )  Valeur 1 ou 2;;
    val3 )         Valeur 3;;
    * )            Autres valeurs;;
esac
```

- **Les fonctions**

Déclaration de la fonction :

```
nom_fonction(){
        instructions
}
```

Appel de la fonction :

```
nom_fonction
```

- **La couleur**

Syntaxe :

```
echo -e '\033[A;B;Cm Bonjour ! \033[0m'
```

Valeurs d'effets (A) :

0	Normal
1	Gras
21	Non-gras
4	Souligné
24	Non souligné
5	Clignotant
25	Non-clignotant
7	Inversé
27	Non-inversé

Valeurs de couleur (B et C) :

Couleur	Couleur texte (B)	Couleur fond (C)
Noir	30	40
Rouge	31	41
Vert	32	42
Jaune	33	43
Bleu	34	44
Magenta	35	45
Cyan	36	46
Blanc	37	47