Sécurisation de la messagerie électronique

Matthieu Herrb
CNRS-LAAS
matthieu.herrb@laas.fr

Octobre 2003

Introduction

La messagerie : le Service Universel

- Service normalisé il y a 21 ans (RFC 821/822).
- Devenu un outil indispensable de communication.

  Mais aussi vecteur pour les utilisateurs abusifs ou malveillants :
  SPAM / vers / virus / etc.
- Évolutions vers des outils encore plus globaux : SMS, PDA, Web services, etc.

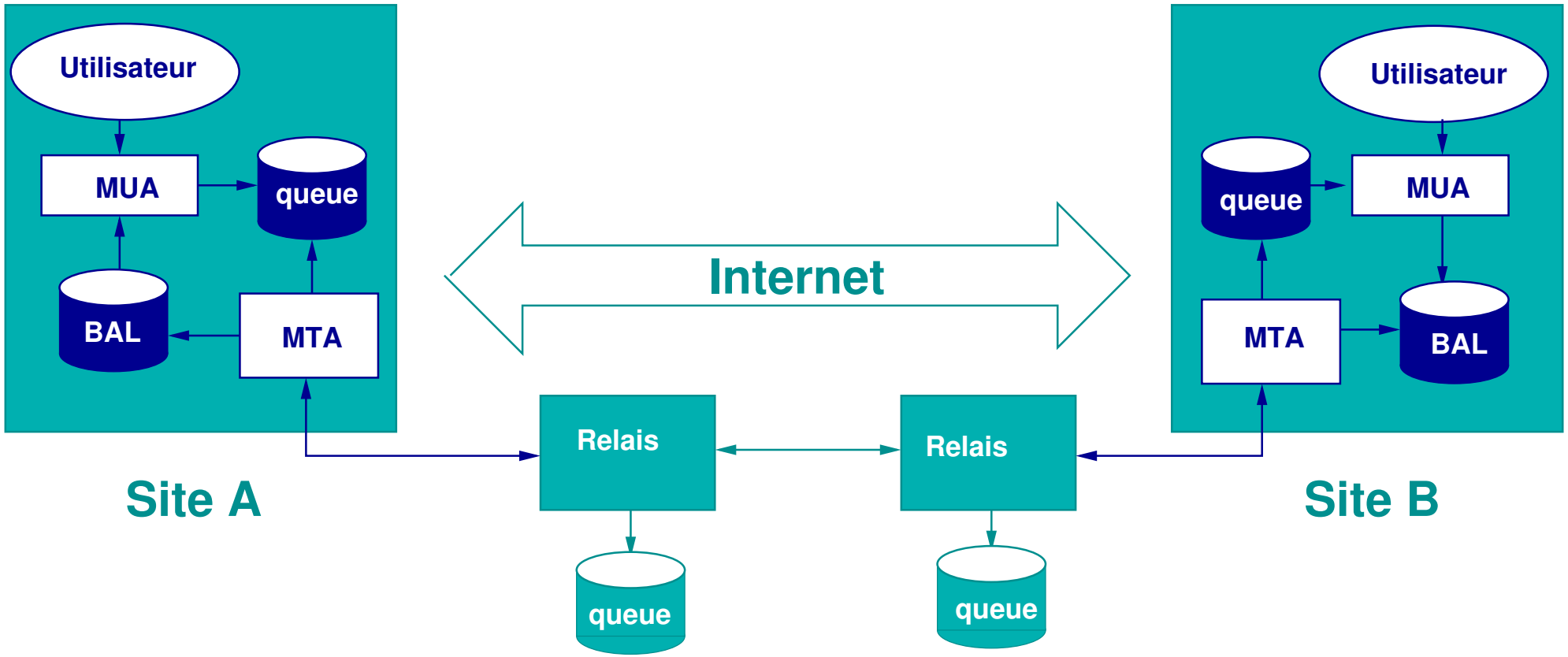
Plan

- Rappels - architecture de la messagerie
  - MUA - MTA
  - les protocoles SMTP, POP3, IMAP
  - MIME
- Vulnérabilités, Menaces et Attaques
- Outils de sécurisation
  - Bonne pratiques d'administration
  - Filtrage : anti-spam et anti-virus
  - Sécurisation du contenu S/MIME et PGP
  - Sécurisation du transport : TLS
  - Solutions pour utilisateurs nomades

Chapitre 1

Architecture(s) de messagerie

Schéma général

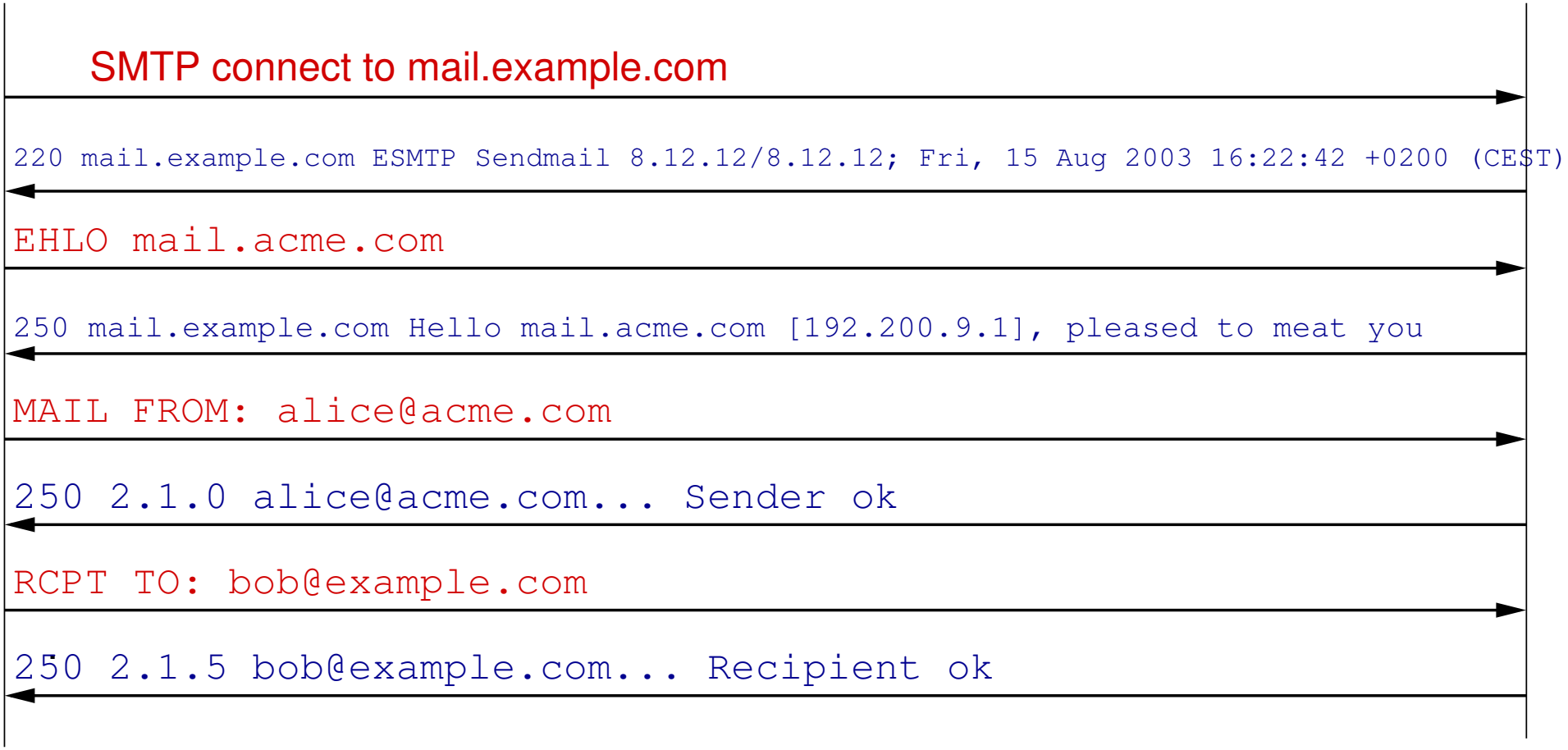


SMTP (1/2)

RFC2821 - Version revue du RFC 821 + ESMTP

Alice
mail to: bob@example.com

Bob



SMTP (2/2)


Alice
mail to: bob@example.com
Bob
DATA
354 Enter mail, end with "." on a line by itself
From: alice@acme.com
To: bob@example.com
Subject: Hello
Date: Sat 26 jul 2003 14:14:14 +0200
Salut, comment
ca va?
.
250 2.0.0 h7FEMdQD001359 Message accepted for delivery
QUIT
221 2.0.0 mail.example.com closing connection
Cloture connexion TCP

Résolution des adresses de messagerie

`utilisateur@exemple.domaine.fr`

Partie gauche

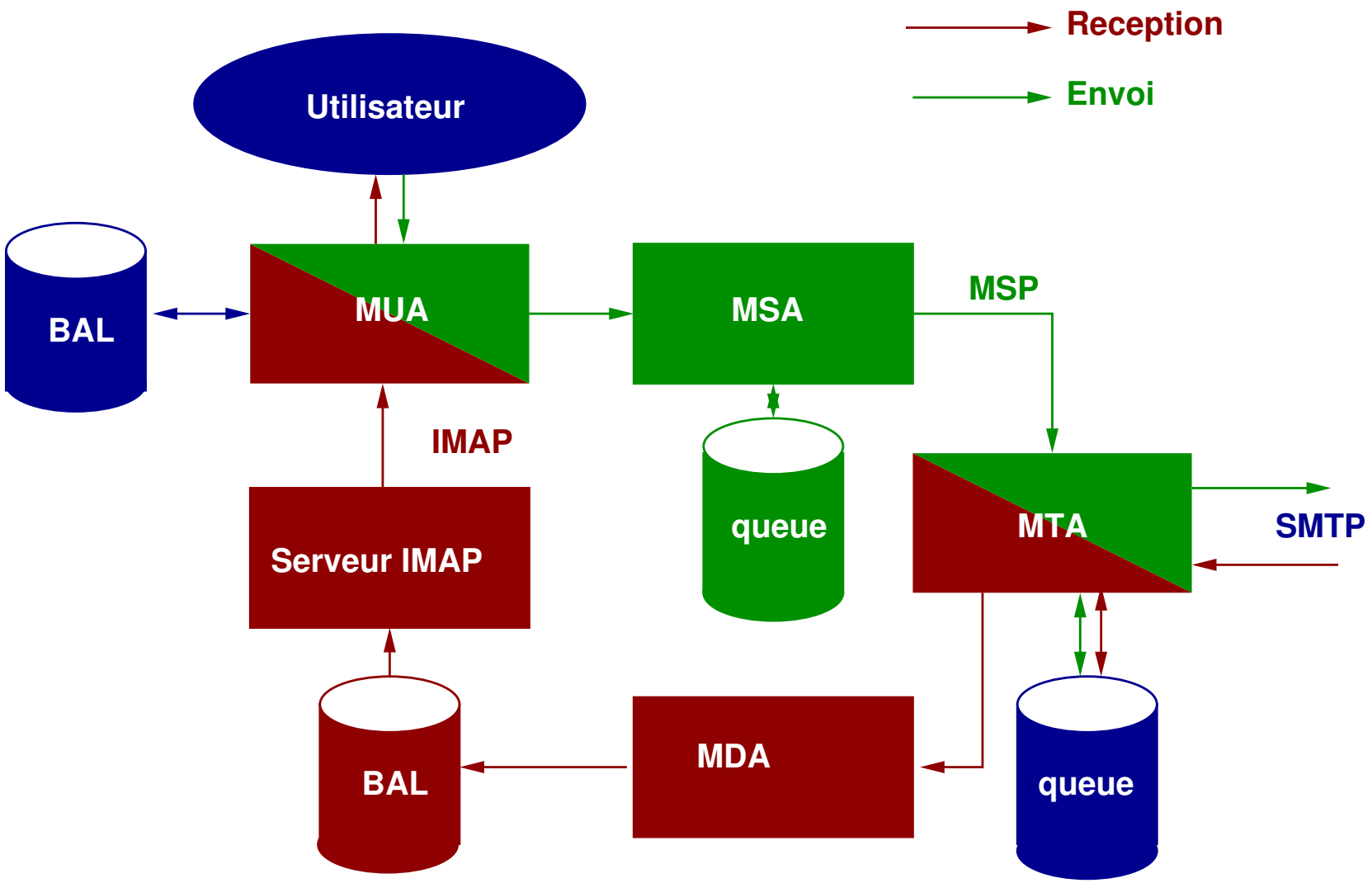
Définie et interprétée localement par le site de destination.

Partie droite

Adresse de messagerie du site. Utilise le DNS. Peut être :
- un enregistrement de type MX (Mail eXanger)
  pointe vers un enregistrement de type A.
- un enregistrement de type A (adresse IP)
- un enregistrement de type CNAME (alias – Attention danger)

Attention : un MX ne peut pas pointer vers un CNAME.

Zoom sur l'architecture locale



Composants d'un serveur de messagerie

MUA *Mail User Agent* logiciel de messagerie de l'utilisateur : Outlook, Evolution, Netscape Messenger, mutt, pine, etc.

MTA *Mail Transport Agent* serveur SMTP pour le transport des messages : sendmail, postfix, qmail, exim, etc.

MSA *Mail Submission Agent* (optionnel) interface utilisée par le MUA pour soumettre les messages sortants au MUA. (sendmail -Ac). Fonctionnalité parfois intégrée au MTA.

MDA *Mail Delivery Agent* outil utilisé pour déposer les messages dans la boîte à lettres locale (`/bin/mail`, procmail, etc.)

Serveur IMAP/POP (optionnel) serveur proposant l'accès aux boîtes aux lettre selon le protocole choisi POP ou IMAP.

Accès aux boîtes à lettres

3 protocoles possibles :

- accès direct par le système de fichiers - utilisé par les anciens MUA Unix (mailx, mailtool, etc.)
- POP3 protocole d'accès à une boîte à lettres à la fois. Très répandu mais limité
- IMAP4 protocole multi boîtes à lettres, plus sophistiqué et plus adapté à la messagerie moderne.

Format des boîtes à lettres

- format Unix traditionnel
  Un fichier messages séparés par `^From␣` Protège les lignes commençant par `From␣` dans le corps du message : `>From`.
- format Unix avec `Content-Length`
  Le même mais l'entête `Content-Length` est utilisé en plus pour donner la longueur de chaque message.
  Plus besoin de protéger les lignes qui commencent par `From␣`
  Mais risque de confusion avec le format traditionnel.
- format MMDF - un fichier par BAL, séparation par `^A^A^A^A`.
- format MH
  Un répertoire par boîte à lettres, un fichier par message, avec un nom numérique.
- format maildir (qmail)
  Un répertoire par boîte à lettres, 3 sous-répertoires `tmp`, `new` et `cur`. Dans chaque répertoire un message a un identificateur unique
- formats propriétaires : Outlook Express, Eudora, etc.

Enveloppe/Message

Enveloppe : données échangées par le protocole SMTP.

En-têtes : données présentes dans le corps du message.

Il n'y a pas de lien entre l'enveloppe et le contenu des messages, sauf quelques entêtes (`Received :`) ajoutées par les MTA.

En particulier aucune garantie sur les entêtes `From :` et `To :` d'un message. C'est l'enveloppe qui permet au message d'arriver dans une boîte à lettres, pas les entêtes.

MIME

RFC 2045..2049 (1996), 2183 (1997).

Multi-purpose **I**nternet **M**ail **E**xtensions

Format historique d'un message texte (ASCII - 7 bits) seul.

MIME permet de transporter des contenus quelconques.

Nouveaux en-tête :
- `MIME-Version : 1.0`
- `Content-Type :`
- `Content-Transfer-Encoding :`

MIME - types de messages

`Content-Type` : définit comment afficher le message à la réception.

Exemples : `text/plain`, `text/html`, `text/enriched`, `image/gif`, `video/mpeg`, `application/pdf`

Messages composites : `multipart` définit un séparateur. Entre chaque séparateur structure MIME avec `Content-Type` : et `Content-Encoding`

- `multipart/mixed` : plusieurs éléments à la suite
- `multipart/alternative` : plusieurs versions du même élément
- `multipart/parallel` : plusieurs éléments à montrer en même temps (video/audio)
- `multipart/signed` : Un élément et sa signature numérique

Possibilité d'emboîter les `multipart`

MIME - Transfert Encoding

Codage du message pour le transport compatible avec le protocole SMTP.

Cinq méthodes standard :

méthode	exemple (cet été)	commentaires
7bit	cet iti	pas bon...
8bit	cet été	
quoted-printable	cet =E9t=E9	
base64	Y2V0IOI06Qo=	
binary	cet été	réservé pour HTML

MIME - exemple

```
Mime-Version: 1.0
Content-Type: multipart/mixed; boundary="3Pql8miugIZX0722"

--3Pql8miugIZX0722
Content-Type: text/plain; charset=iso-8859-1
Content-Disposition: inline
Content-Transfer-Encoding: 8bit

Bonjour,
blah blah, ci-joint un document pdf génial, blah blah

--3Pql8miugIZX0722
Content-Type: application/pdf
Content-Disposition: attachement; filename="figure.pdf"
Content-Transfer-Encoding: base64

JVBERi0xLjMKJcfsj6IKNiAwIG9iago8PC9MZW5ndGggNyAwIFIvRmlsdGVyIC9GbGF0ZURl
Y29kZT4+CnN0cmVhbQp4nN1aS4/cxhG+81fw6BhG+81fw6BhQp9+Po4wYRoAIiLRr5BxM1paCoRzJkvP3

--3Pql8miugIZX0722--
```

Chapitre 2

Vulnérabilités, Menaces, Attaques

Résumé

Vulnérabilités :
- pas d'authentification ni d'intégrité ni de confidentialité,
- relayage SMTP,
- transport de contenus exécutables,
- excès de privilèges,
- failles des logiciels,
- etc.

Menaces :
- diffusion de SPAM
- propagation de virus
- dénis de service
- élévation de privilèges
- atteinte à la vie privée des utilisateurs,
- etc.

Idée reçue : seul Windows est vulnérable...

Absence d'authentification, d'intégrité et de confidentialité

Le protocole SMTP ne fournit aucun de ces trois services de base.

- les messages (y compris les entêtes) sont transportés sans aucune vérification.
- les contrôles sur l'enveloppe sont limités aux règles anti-relayage. L'enveloppe peut contenir n'importe quoi pourvu que ça passe l'anti-relais
- un relais SMTP peut modifier un message sans violer le protocole
- les échanges se font en clair

Ne jamais faire confiance à un e-mail !

Relayage SMTP

Faille dans le protocole :

Le protocole SMTP ne restreint pas les connexions acceptées. Historiquement un relais SMTP acceptait de relayer n'importe quel message.

Utilisé (entre autres) pour diffuser du SPAM en masquant le plus possible l'identité de l'expéditeur : connexion à un relais SMTP aléatoire entre l'expéditeur et le destinataire.

Règle :
Un serveur SMTP accepte de relayer un message seulement si l'expéditeur ou le destinataire est local.

Mise en oeuvre :
Utilise les champs de l'enveloppe (EHLO, MAIL FROM : et RCPT TO :).
Vérifications dans le DNS. Comparaison avec la liste des sites locaux.

Contenus exécutables

(problème pas spécifique à la messagerie)

MIME permet de transporter des fichiers exécutables (code machine, scripts, plug-ins,...)

Pour le confort de l'utilisateur les MUA ouvrent automatiquement les documents attachés ; pour un document exécutable, l'action par défaut est (était) de l'exécuter.

□ « Ingénierie Sociale » convaincre les utilisateurs d'ouvrir (exécuter) les documents attachés malgré les recommandations (exemple : virus Swen, se présente comme mise à jour de sécurité Microsoft).

$\longrightarrow$ vecteur de propagation rêvé pour vers/virus. Ex. I Love You : temps de diffusion dans tout l'internet $<$ 12h.

Dénis de service

Le protocole SMTP suppose une utilisation raisonnable des ressources.

Un serveur de messagerie se met en général complètement hors-ligne lorsque sa charge devient trop importante.

Que se passe-t-il si un utilisateur est déraisonnable (nombre de messages/taille des messages,...) ?

Malgré l'ajout de nombreuses limites plus ou moins pifométriques, de nombreuses configurations sont sujettes à des dénis de service potentiels .

Protection de la vie privée

Problèmes lié à l'absence de confidentialité mais en plus :

- commandes SMTP `expn` et `vrfy` donnent accès à la liste des comptes connus,
- informations contenues dans les entêtes permettent de tracer géographiquement (et temporellement) l'émetteur d'un message
- systèmes de répondeurs automatiques qui donnent souvent trop d'information. Bien si l'interlocuteur est amical, problématique s'il est malfaisant.

Excès de privilèges

MTA et MDA exécutés avec l'identité du super-utilisateur augmentent l'impact des vulnérabilités dans ces logiciels.

Sendmail $<$ 8.12 installé avec bit **setuid** $\rightarrow$ uid 0 même si lançé par un utilisateur λ.

Compte de messagerie $==$ compte Unix avec shell et exécution de code sur le serveur POP/IMAP.

L'outil local de messagerie a l'accès complet à la machine de l'utilisateur, alors qu'il traite des contenus potentiellement dangereux.

Failles de logiciels

Défauts dans l'implémentation des logiciels qui créent des possibilités supplémentaires.

Ex. Vers historique d'internet (1985) exploitant une faille de sendmail.

Failles dans les mesures anti-relais

Configuration par défaut des MUA trop permissive : ouverture automatique des documents attachés (y compris exécutables), javascript, etc.

Failles dans les MUA permettant l'exécution de code à distance malgré la désactivation de l'option d'ouverture automatique des documents attachés.

Exemples :
- MS02-058 Unchecked Buffer in OE S/MIME parsing could enable system compromize,
- CAN-2003-0721, Pine integer overflow in its MIME header parsing

Chapitre 3

Outils de sécurisation

Fonctions de sécurité

Disponibilité	Redondance, sauvegardes, bonne administration
Intégrité	Signature
Authentification	
Non-répudiation	
Confidentialité	Chiffrement

Bonnes pratiques d'administration

Recommandations de base

- Espace disque suffisant
- Sauvegardes /var/mail / RAID
- Ne pas autoriser le relayage
- Surveillance des journaux / provoquer des alertes
- Gérer les erreurs
- Utiliser un MX secondaire
- Annuaire des utilisateurs
- Respect de la vie privée des utilisateurs
- Respecter les RFCs et la netetiquette :
  - partie droite des adresses solubles dans le DNS
  - serveur qui respecte le protocole SMTP
  - alias standard pour contact extérieur : postmaster, abuse, etc.
  - génère des entêtes MIME corrects (codage ISO,...)
  - utilisateurs conscients des formats MIME et de leur bon usage
  - signatures raisonnables

Redondance

- Serveur DNS secondaire hors - site
  rôle : garantir que des agents SMTP pourront toujours trouver votre domaine et les serveurs de messagerie du domaine, même si ces derniers ne sont pas accessibles.
- Serveur MX secondaire
  - sur site : le plus simple à mettre en oeuvre. Duplication de la configuration du serveur principal si fonction relais seul.
  - hors site : sur un serveur d'un domaine existant. Accepter le courrier pour son site, relais vers le serveur principal.
- Problème ouvert : serveur SMTP secondaire pour les utilisateurs internes ? (la plupart des MUA ne connaissent qu'un serveur SMTP).

Sauvegardes

Éléments à sauvegarder :

Configuration	lorsqu'elle change
Données annexes (alias, access, ...)	lorsqu'elles changent
Boîtes aux lettres	au moins une fois par jour
Queues	

Redondance matérielle : système RAID pour BAL / Queue :

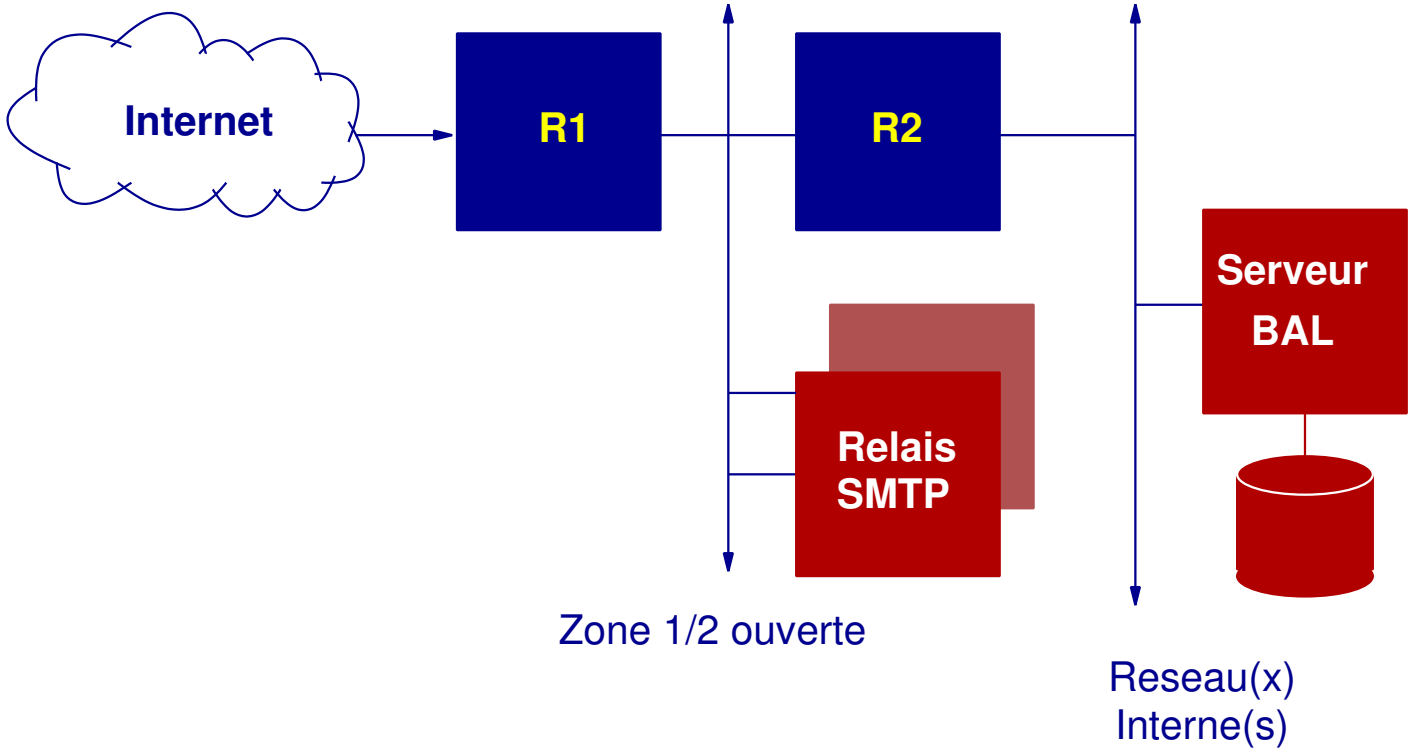
- pas d'arrêt en cas de panne,
- pas de perte de données en cas de panne,
- mais ne peut rien contre les erreurs humaines.

Gestion des utilisateurs

Règles de bonne organisation vis à vis des utilisateurs :

- avoir une charte d'utilisation de la messagerie,
- annuaire : correspondance compte messagerie / personnes physiques,
- fermer les comptes non utilisés,
- réfléchir à la mise en place de quotas sur les boîtes aux lettres,
- avoir des moyens pour informer les utilisateurs,
- informer le CE / Conseil de Laboratoire des mesures de sécurité.

Architecture réseau



R1 : laisse passer le port 25 de l'internet vers le relais SMTP
bloque SMTP vers le réseau interne
bloque SMTP du réseau interne vers l'internet

R2 : Routeur ou commutateur/routeur interne
Peut faire du filtrage entre réseaux internes

Analyse de logs MTA

Points à surveiller :

- erreurs de connexion SMTP
- erreurs de résolution DNS
- connexions trop fréquentes
- taille de la file d'attente
- tentatives de connexions sur des machines autre que le serveur
- tout ce qui peut paraître étrange

Exemples d'outils d'analyse de log :

Logcheck : http://www.astro.uiuc.edu/~r-dass/logcheck/

isoqlog : http://www.enderunix.org/isoqlog/

Filtres anti-virus / anti-spam

À quel niveau (MUA / MTA) ?

Niveau MUA indispensable mais :
- garantir la mise à jour (automatique ?)
- être exhaustif (un seul poste non protégé peut bloquer/contaminer tout un réseau).

Au niveau MTA :
- protection globale
- plus facile de garantir la mise à jour et l'utilisation systématique.
- problèmes de charge serveurs.
- problèmes de politique de sécurité globale.

Anti-virus

Approches :

- **par signature :** ce que font les anti-virus traditionnels
  Problèmes :
  « fenêtre de vulnérabilité » entre l'apparition d'un nouveau virus et la mise à jour des anti-virus.
  fausses alertes de plus en plus nombreuses (statistiquement inévitable).
  coût des solutions les plus performantes

- **plus simple :** bloquer l'exécution des contenus potentiellement dangereux.
  Capable de bloquer de nouveaux vers/virus.
  Pas de faux positifs (par définition).
  Problèmes :
  liste exhaustive des types MIME exécutables ?
  comment échanger des exécutables légitimes ?

Anti-virus au niveau MTA

Plusieurs solutions existent :

- Relais SMTP intégré
  - amavisd-new http://www.ijs.si/software/amavisd/,
  - mail-scanner http://www.sng.ecs.soton.ac.uk/mailscanner/
  - Produits commerciaux : Sophos PureMessage, Symantec AntiVirus Gateway, MacAfee WebShield SMTP,

  Avantages : indépendant du serveur SMTP utilisé, peu de modifications de configuration

  Inconvénients : manque de souplesse de configuration, nécessite une machine supplémentaire
- API Milter pour sendmail :
  - MIMEDefang
  - Colburn Anti-virus

  Avantages : grande richesse de configurations, s'intègre dans une configuration existante sur la même machine

  Inconvénients : configuration un peu lourde (pas incluse en standard en général), spécifique sendmail

Le spam

SPAM = Messages électroniques non sollicités.

Nuisance de plus en plus importante.
(Contenus douteux...)

Diffusés par des relais ouverts à des listes d'adresses collectées sur le Web.

Pourquoi tant de SPAM ?

Envoyer un message ne coûte presque rien.

Un taux de retour très faible permet de faire des bénéfices.

Aucune chance de voir les spammeurs arrêter : activité lucrative et sans risques.
(Source : Wall Street Journal – http://online.wsj.com/article_email/0,,SB10371386792204471 48,00.html)

Exemple récent : spam vendant des outils anti-spam.

Seul point positif : les spammeurs continueront à utiliser les solutions les moins chères possible pour conserver leurs marges bénéficiaires $\longrightarrow$ outils imparfaits et détectables.

SPAM - aspect légaux

Le spam est illégal.

La loi sur l'économie numérique instaure le principe du :
« **consentement préalable** en matière de prospection directe opérée par des systèmes automatisés d'appel, télécopieurs ou courriers électroniques ».

Mais il y a des exceptions...

Cf. Délibération CNIL n° 02-093

LEN : http://www.droit-technologie.org/3_1.asp?legislation_id=138

commentaire : http://www.droit-technologie.org/1_2.asp?actu_id=714

Site général : http://www.spamlaws.com/

[Je ne suis pas juriste]

Lutte contre le SPAM - Recommandations de la CNIL

http://www.cnil.fr/thematic/internet/spam/lacnil_aide1.htm

- Faites toujours preuve de vigilance quand vous communiquez votre adresse électronique.
- Ne rendez pas visible les adresses méls de vos correspondants lorsque vous créez un groupe ou une liste de diffusion.
- Sensibilisez vos enfants sur l'utilisation qu'ils peuvent être amenés à faire de leur adresse électronique
- Ne répondez jamais à un « spam ».
- Ne communiquez pas à des tiers des adresses mél autres que la votre sans le consentement des intéressés.
- Utilisez un filtre de « spam ».
- Ne cliquez pas sur les liens hypertexte insérés dans le corps du « spam ».
- Ne jamais ouvrir un fichier joint figurant dans un « spam ».

En pratique...

Pour l'utilisateur

- Le bouton 'Poubelle' (ou le raccourci clavier) est le moyen le plus rapide et le plus efficace jusqu'à environ 30 messages par jour.
- Au delà : filtrage (traité plus bas)

Pour l'administrateur système

- (In)former ses utilisateurs
  - Cf. recommandations de la CNIL
  - Ne pas devenir spammeurs (organisation de conférences, gestion de listes de mail, etc.)
- Mettre en place un dispositif anti-spam au niveau du serveur de messagerie
- Lutter contre les relais ouverts.
- Dénoncer à la justice les pratiques illégales (pédophilie, « chaînes », etc.)

Politiques de filtrage

Définir avec le conseil de laboratoire / CE une politique de filtrage claire

Basée sur la charte d'utilisation des moyens informatiques

Définir
- ce qui est du trafic légitime,
- ce qui sera bloqué (mis en quarantaine).

Tolérer une utilisation raisonnable à usage extra-professionel.

Respecter les conseils de la CNIL dans les fiches pratiques éditées avec le rapport sur la cyber-surveillance sur les lieux de travail :

http://www.cnil.fr/thematic/docs/entrep/cybersurveillance2.pdf
http://www.cnil.fr/thematic/docs/entrep/cyber_fiches.pdf

Mimedefang - Outil général de filtrage pour sendmail

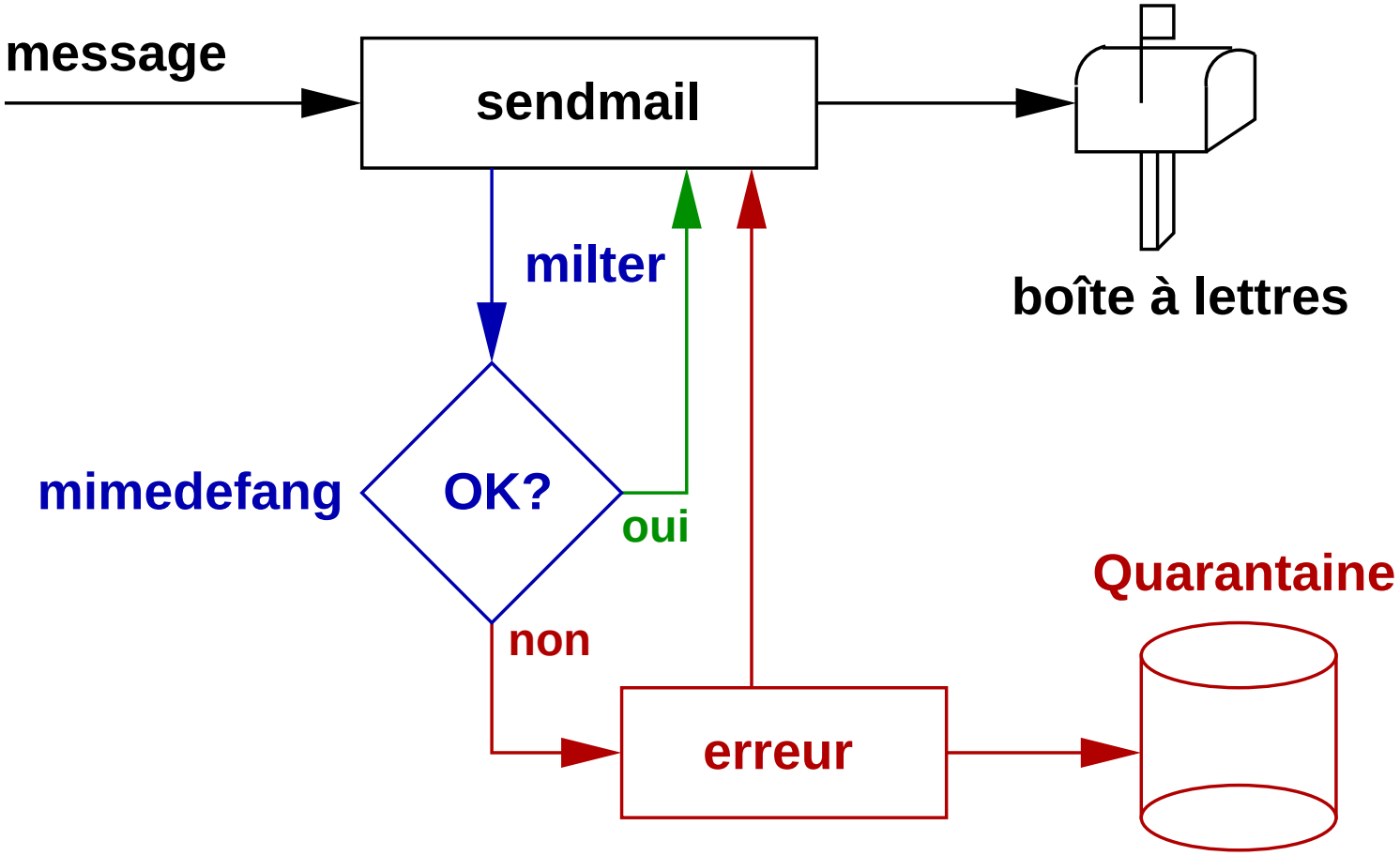
http://www.roaringpenguin.com/mimedefang/
http://www.mimedefang.org/

Programme général de filtrage de mails, utilisant l'API milter de Sendmail.

- destruction, modification ou mise en quarantaine de fichiers attachés « dangereux »
- interface avec Anti-virus existants, avec SpamAssassin
- ajout de notices aux messages
- actions configurables en fonction du domaine, de l'utilisateur, du relais utilisé, etc.
- utilise un « pool » de processus pour les serveurs chargés.
- ...

$\longrightarrow$ Permet de faire anti-virus et anti-spam en même temps.

Interaction sendmail/mimedefang



Listes noires - Principe

Rejeter les connexions SMTP en provenance de relais ouverts connus.

Fonctionnement : Un serveur DNS gère un domaine spécial contenant une base de donnée de relais ouverts.

Exemple : Est-ce que `192.9.200.1` (`mail.example.net`) est un relais ouvert ?

$\longrightarrow$ recherche `1.200.9.192.relays.ordb.org`.

Si réponse, alors c'est un relais ouvert...

Configuration : (sendmail/m4) Dans le fichier `.mc` ajouter :

```
FEATURE(‘dnsbl’,‘relays.ordb.org’,‘"550 rejected - see http://ordb.org/"’)
```

Possibilité d'exceptions dans `accessdb` : `192.9.200.1 OK`

Quelques listes noires :

`rbl.net`	payant
`SpamCop.net`	payant
`relays.ordb.org`	http://ordb.org/

Problèmes des listes noires

- Contenu non maîtrisé
- Plus facile d'y rentrer que d'en sortir
- Souvent trop agressives (bloque toute une classe C pour un relais ouvert)
- problème récent Osirusoft : s'est mis à bloquer tout l'internet.
- etc.

Listes blanches

Complémentaire extrême des listes noires :

N'accepter que les messages de personnes identifiées.

- despam : http://www.laas.fr/~felix/despam.html
- http://impressive.net/people/gerald/2000/12/spam-filtering.html
- n'accepter que des messages signés (S/MIME ou PGP) avec ou non une liste d'autorités de certification de confiance.

Filtrage sur le contenu : principe

Classifier les messages en fonction du texte complet du message

- Par mots clés / patterns : SpamAssassin
- Par analyse statistique : Classification Bayesienne (bogofilter, bmf, etc.)
- Base de donnée de spams : Vipul's Razor.

SpamAssassin™

http://www.spamassassin.org/

Ensemble de tests sur le contenu
Chaque test attribue des points
Somme des points → score

Marque les messages qui dépassent un seuil.

Écrit en Perl.

Plusieurs modes de fonctionnement :
- filtre simple (utilisation avec procmail)
- filtre client d'un démon (meilleurs performances) (`spamc`)
- au travers de l'API milter (sendmail) → filtrage global

SpamAssassin - Exemple

```
SPAM: -------------------- Start SpamAssassin results ----------------------
SPAM: This mail is probably spam.  The original message has been altered
SPAM: so you can recognise or block similar unwanted mail in future.
SPAM: See http://spamassassin.org/tag/ for more details.
SPAM:
SPAM: Content analysis details:   (8.10 hits, 5 required)
SPAM: SUBJECT_MONTH      (-0.5 points) Subject contains a month name - probable
newsletter
SPAM: NO_REAL_NAME       (1.3 points)  From: does not include a real name
SPAM: LOSE_POUNDS        (0.5 points)  Subject talks about losing pounds
SPAM: DIET               (0.4 points)  BODY: Lose Weight Spam
SPAM: FULL_REFUND        (0.4 points)  BODY: Offers a full refund
SPAM: CLICK_BELOW        (0.3 points)  BODY: Asks you to click below
SPAM: SPAM_PHRASE_08_13  (1.4 points)  BODY: Spam phrases score is 08 to 13 (med
ium)
SPAM:                    [score: 8]
SPAM: DATE_IN_FUTURE_06_12 (1.1 points)  Date: is 6 to 12 hours after Received:
date
SPAM: RCVD_IN_DSBL       (3.2 points)  RBL: Received via a relay in list.dsbl.or
g
```

```
SPAM:                       [RBL check: found 251.102.96.210.list.dsbl.org]
SPAM:
SPAM: ---------------------- End of SpamAssassin results ---------------------
```

Classificateurs bayesiens

Classification bayesienne :
http://www.mathpages.com/home/kmath267.htm

Eric Horvitz & al. (Microsoft) 1998 :
http://research.microsoft.com/~horvitz/junkfilter.htm

Paul Graham 2002-2003 :
http://www.paulgraham.com/spam.html

Adaptive latent semantic analysis (Apple Mail.app) :
http://www.knowledge-technologies.com/papers/dp1.LSAintro.pdf

Implémentations libres :
http://bogofilter.sourceforge.net/ (Eric S. Raymond)
http://sourceforge.net/projects/bmf/
http://spambayes.sourceforge.net/
http://www.mozilla.org/mailnews/spam.html (Mozilla Mail)
http://www.fourmilab.ch/annoyance-filter/

Rappels

$$P(C = c_k|X = x) = \frac{P(X = x|C = c_k)P(C = c_k)}{P(X = x)}$$

Ici deux classes : C_0 HAM
C_1 SPAM

Bayes naïf :

$$P(X = x|C = C_k) = \prod_i P(X_i = x_i|C = c_k)$$

Paul Graham

Décomposition en tokens du texte complet (y compris en-têtes)

Base d'apprentissage classée à la main. Pour chaque token calcule la probabilité d'être SPAM.

Nouveaux messages :
- assigne initialement 0.4 comme probabilité aux mots inconnus ,
- garde les 15 probabilités les plus significatives (les plus éloignées de 0.5),
- calcule la probabilité conditionnelle que le mail soit un SPAM,
- en fonction de la décision recalcule les probabilités des tokens.

Implémentations :
- bogofilter
- bmf
- un module dans SpamAssassin

Classification par score ou probabiliste ?

Qu'y a-t-il derrière un score ? Comment les attribuer ?

Scores $\longrightarrow$ difficiles à personnaliser

Probabilités $\longrightarrow$ valeurs absolues

Apprentissage personnalisé automatique (2 boutons : SPAM/HAM)

Permet de suivre l'évolution des contenus du SPAM.

Les pièges à SPAM

Principe : pénaliser les diffuseurs de SPAM en ralentissant jusqu'à l'insupportable la transaction SMTP s'il s'agit de SPAM.

- **OpenBSD : spamd**
  http://www.openbsd.org/cgi-bin/man.cgi?query=spamd&manpath=OpenBSD+Current&format=html
  Basé sur des listes noires et sur le filtre de packets **pf** pour rediriger les connexions des spammeurs vers spamd.
- **Spam tarpit - tarproxy**
  http://www.martiansoftware.com/articles/spammerpain.html
  Utilise un filtre baysien pour analyser le message et ralentir la transaction si le score augmente.

Mimedefang + Spam Assassin

`mimedefang-filter` est un script Perl.

Utiliser SpamAssassin :

Dans `filter_end` :

```
    # verifier si SpamAssassin est installe
    if ($Features{"SpamAssassin"}) {
        if (-s "./INPUTMSG" < 100*1024) {
            # ne scanner que les messages de taille inferieure a 100ko
            my($hits, $req, $names, $report) = spam_assassin_check();
            if ($hits >= $req) {
# Marquer les messages
                action_change_header("X-Spam-Score", "$hits $names");
                action_change_header("X-Spam-Status", "Yes");
            }
        }
    }
```

Mimedefang + SpamAssassin pour les utilisateurs

Chaque message soupçonné par par SpamAssassin d'être un SPAM est marqué par des entêtes :

```
X-Spam-Score: 7.9 (*******) ALL_CAP_PORN,CLICK_BELOW,CLICK_BELOW_CAPS,
  CTYPE_JUST_HTML,DATE_MISSING,FROM_ENDS_IN_NUMS,LINES_OF_YELLING,
  LINES_OF_YELLING_2,LINES_OF_YELLING_3,PORN_4,SPAM_PHRASE_05_08,
  SUPERLONG_LINE,UPPERCASE_50_75
X-Spam-Status: Yes
X-Scanned-By: MIMEDefang 2.28 (www . roaringpenguin . com / mimedefang)
```

Filtrage par chaque utilisateur avec :

- procmail (cf plus haut)
- logiciel de messagerie : Eudora, Netscape messenger, ...

Mimedefang - Autres exemples (1)

Alias interne seul

```
sub filter_recipient {
        my ($recipient, $sender, $ip, $hostname, $first, $helo) = @_;
        if ($recipient =~ /^<?all\@example\.net>?$/i) {
                if ($sender !~ /\@example\.net>?$/i) {
                        return ('REJECT', 'User unknown');
                }
                return ('CONTINUE', 'ok');
        }
}
```

Rejette les messages à l'adresse `all@example.net` si l'expéditeur n'est pas dans le domaine `example.net`.

Mimedefang - Autres exemples (2)

Anti-spoofing sur HELO/EHLO

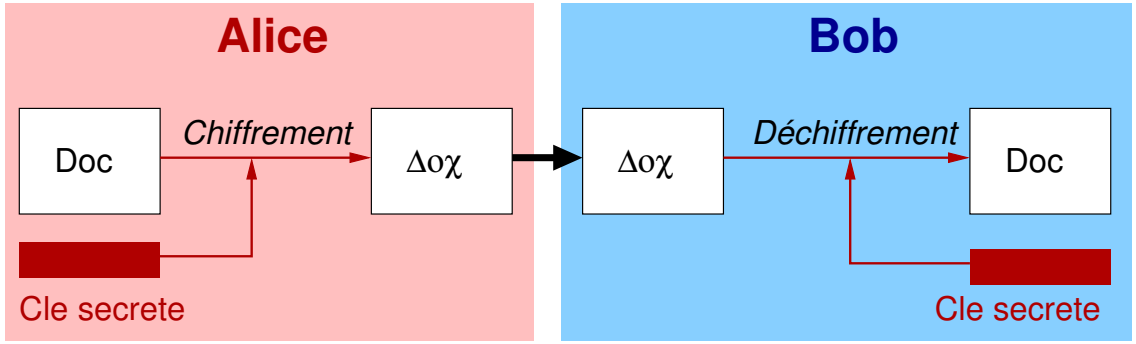
```
sub filter_relay {
        my($ip, $name, $helo) = @_;
        if ($helo =~ /example\.net/i) {
          if ($ip ne "127.0.0.1" and
              $ip !~ "^192\.200\.7\.") {
                return('REJECT', "Go away... $ip is not in example.net");
          }
        }
        return ('CONTINUE', "ok");
}
```

Rejette les connexions qui essayent de passer `example.net` en paramètre de la requête SMTP **HELO** ou **EHLO**.

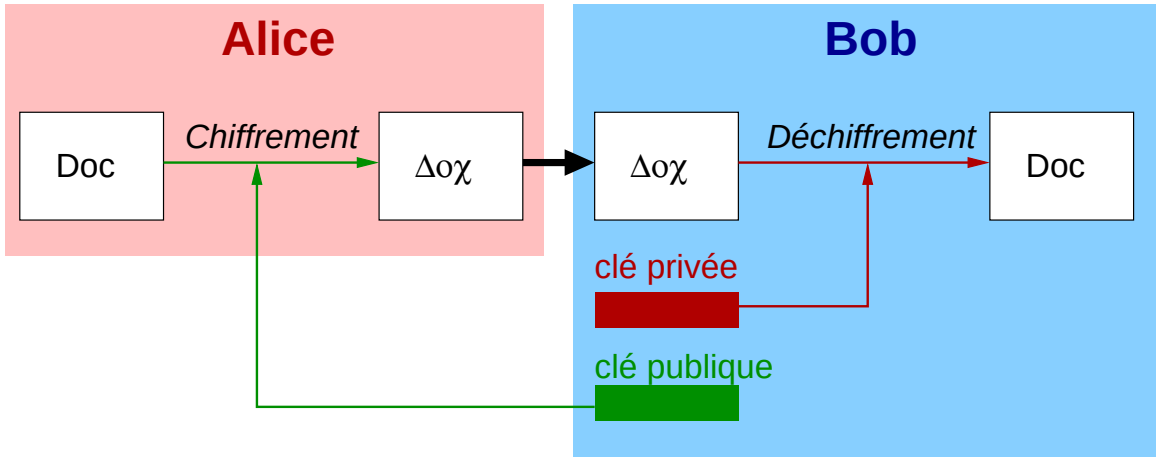
Sécurisation du contenu

Chiffrement symétrique et asymétrique

Symétrique : DES / AES



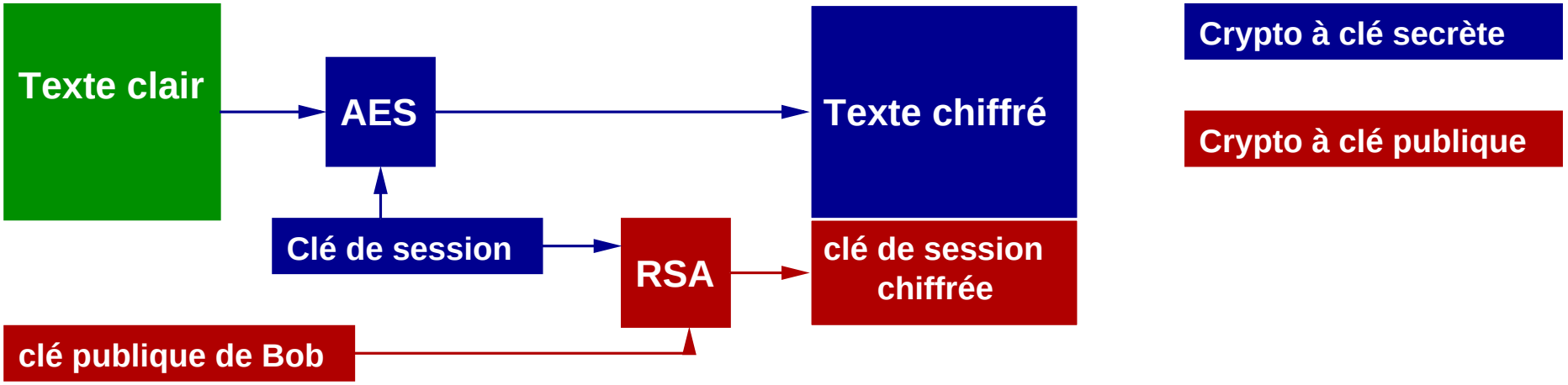
Assymétrique : RSA (DSA - El-Gamal)



Chiffrement pratique

Fonctions à clé publique très coûteuses $\longrightarrow$ utilisation d'une **clé de session**

Chiffrement par Alice :

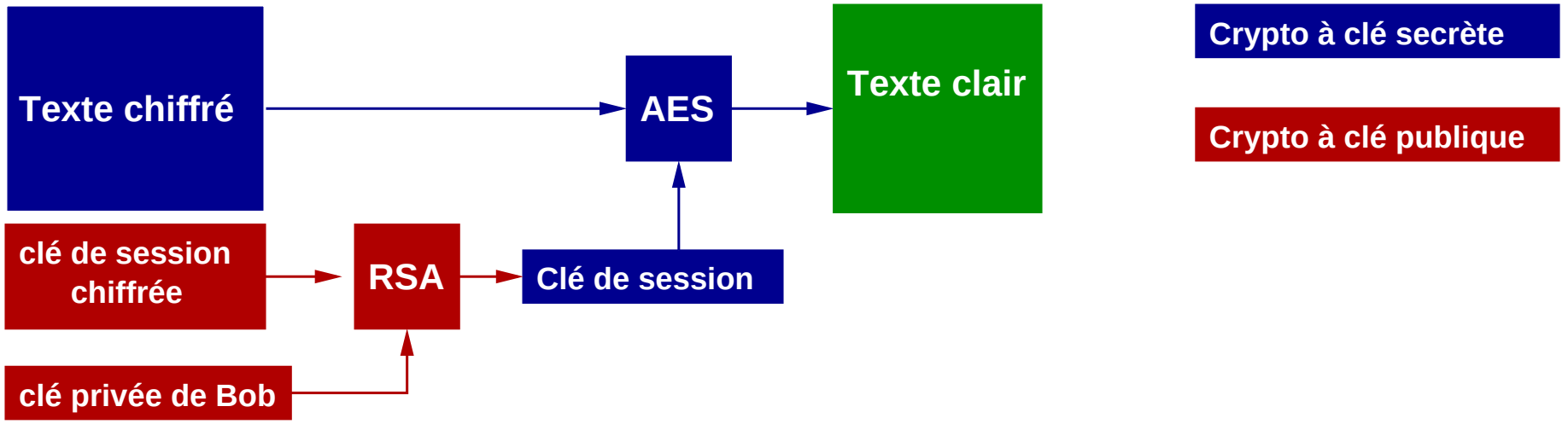


Longueur des clés sûres (2003) :

clé secrète : 128 bits
clé publique/privée : 1024 bits

Déchiffrement pratique

Déchiffrement par Bob :



Fonctions de hachage

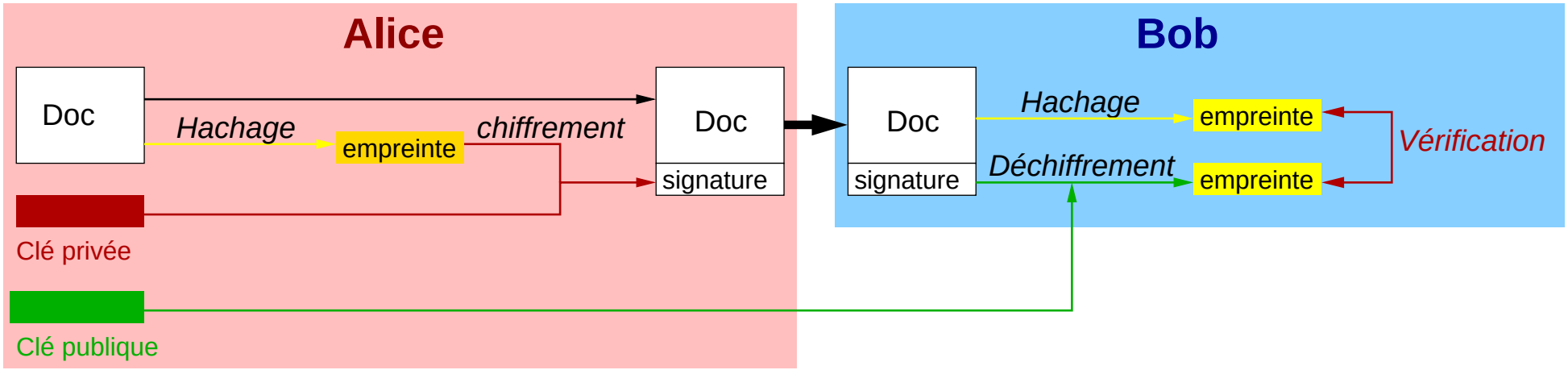
But : obtenir une « empreinte » de petite taille (128 bits) à partir d'un document avec les propriétés suivantes :

- impossible de prédire la modification du document à partir d'une modification de l'empreinte,
- risque nul (quasi-nul) de collision entre 2 versions d'un même document.

Algorithmes principalement utilisés : **MD5**, **SHA1**.

Signature numérique

Principe d'une signature :



Diffusion des clés publiques :

- Infrastructure de Gestion des Clés
- « Web of trust »

Authentification

Prouver son identité

Techniques :

- mot de passe
- challenge - réponse (CHAP)
- biométrie
- certificats

Certificats

Clé publique d'un utilisateur signée par une autorité de certification (de confiance).

Permet de distribuer la confiance dans les clés publiques, donc authentifier les propriétaires de ces clés.

$\longrightarrow$ Infrastructure de Gestion des Clés (IGC).

Format de certificat standard : X.509v3.

Utilisés par S/MIME, SSL/TLS, IPSec, etc.

Deux niveaux d'autorités de confiance :

Autorité d'enregistrement valide l'identité de la personne, signe la demande de certificat (Mairie).

Autorité de certification reçoit les demandes signées, signe la clé publique et publie le tout (Préfecture).

PGP

Pretty **G**ood **P**rivacy.

Dévelopé par Phil Zimmerman en 1991. → OpenPGP RFC 1991, 2015, 2440.

Fonctions : Signature et/ou chiffrement.

Clés publiques signées par des personnes tierces - notion de *Web of trust* : trouver l'homme qui a vu l'homme qui a vu la clé.

Système populaire et bien développé. Outils pour toutes plateformes.

Évolue vers la possibilité d'utiliser une IGC.

PGP - exemple

```
From: matthieu.herrb@example.net
To: matthieu.herrb@example.net
Subject: Test PGP
Date: Sun, 5 Oct 2003 16:50:45 +0200

-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Ceci est un message signé par PGP.
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.2.1 (OpenBSD)

iD8DBQE/gC+8yHzmqagv7mERAs4KAJ9MkwmyuzIojDLg7rk7TlM+hCOUfwCguAP/
gspsnaXr3zlbR/sBQRIWsnM=
=tv14
-----END PGP SIGNATURE-----
```

PGP - Liens utiles

OpenPGP - http://www.openpgp.org/

GNU Privacy Guard - http://www.gnupg.org/

PGP.com (Phil Zimmerman) - http://www.pgp.com

Enigmail (Plugin pour Mozilla/Thunderbird)
http://enigmail.mozdev.org/index.html

S/MIME

RFC 2311-2312 / 2632-2633

Développé par RSA

http://www.imc.org/ietf-smime/index.html

Utilise les certificats X.509v3 pour signer/chiffrer.

Nécessite une infrastructure de gestion des clés.

Supporté par Netscape / Mozilla / Outlook express (RIP).

S/MIME - exemple

```
MIME-Version: 1.0
Content-Type: multipart/signed; protocol="application/x-pkcs7-signature";
    micalg=sha1; boundary="----F2F414CB86C93232A3F2D82F6E113F1E"
From: Matthieu Herrb <matthieu.herrb@example.net>
To: john.doe@other.example.net
Subject: test S/MIME
Date: Sun, 5 Oct 2003 16:57:05 +0200

This is an S/MIME signed message

------F2F414CB86C93232A3F2D82F6E113F1E
Content-Type: text/plain; charset=iso-8859-1
Content-Transfer-Encoding: 8bit

Voici un message signé par S/MIME.


------F2F414CB86C93232A3F2D82F6E113F1E
Content-Type: application/x-pkcs7-signature; name="smime.p7s"
Content-Transfer-Encoding: base64
```

```
Content-Disposition: attachment; filename="smime.p7s"

MIIG7wYJKoZIhvcNAQcCoIIG4DCCBtwCAQExCzAJBgUrDgMCGgUAMAsGCSqGSIb3
DQEHAaCCBKIwggSeMIIDhqADAgECAgIE/DANBgkqhkiG9w0BAQQFADA0MQswCQYD
[....]
jtsD

------F2F414CB86C93232A3F2D82F6E113F1E--
```

S/MIME - liens utiles

Infos générales :
http://www.dartmouth.edu/~pkilab/pages/Using_SMIME_e-mail.html

Groupe de travail IETF :
http://www.ietf.org/html.charters/smime-charter.html

Exemples avec OpenSSL :
http://www.bacus.pt/Net_SSLeay/smime.html

S/MIME dans Mozilla
http://www.mozilla.org/projects/security/pki/nss/smime/

S/MIME vs OpenPGP

Mandatory Features	**S/MIME v3**	**OpenPGP**
Message format	Binary, based on CMS	Binary, based on previous PGP
Certificate format	Binary, base on X.509v3	Binary, based on previous PGP
Symmetric encryption algorithm	3-DES	3-DES
Signature algorithm	Diffie-Hellman with DSS	ElGamal with DSS
MIME encapsulation of signed data	`multipart/signed` or CMS format	`multipart/signed` with ASCII armor
MIME encapsulation of encrypted data	`application/pkcs7-mime`	`multipart/encrypted`

Source : Internet Mail Consortium

Aspects légaux

La règlementation française en matière de cryptologie s'applique.

Recommendations de la DCSSI :
- clés séparées pour signature et chiffrement
- séquestre des clés de chiffrement

La messagerie chiffrée pour des usages réels à grande échelle est loin d'être prête.

Sécurité du transport

Protocoles généraux de sécurité

SSL - protocole au dessus de TCP (Entre TCP et l'application).

IPsec - protocole au niveau transport. Voir tutorial à coté.

Pourquoi SSL ?

Avantages :
- Compatible avec TCP/IP « Classique »
- Protocole standardisé
- Ne nécessite pas de sécuriser tout d'un coup
- Gestion de la confiance par IGC,
- Disponible...

Inconvénients :
- Retard de l'implémentation de certains services (telnet, ftp,...)
- Coût du chiffrement
- Nécessite des certificats
- Quelques failles dans l'implémentation...

SSL - Concepts de base

Secure **S**ocket **L**ayer

Une couche au dessus de TCP qui assure :

- l'authentification du serveur
- l'authentification optionnelle du client
- la confidentialité
- l'intégrité
- la compression (optionnelle)

SSLv1	obsolete
SSLv2	Netscape
SSLv3	Netscape
TLSv1	RFC 2246

SSL Authentification

Utilise des certificats X.509v3

1. Le serveur présente son certificat au client
2. Le client vérifie la signature du certificat
3. Le serveur demande un certificat au client
4. Le client transmet un certificat
5. Le serveur vérifie la signature du certificat du client.

Nécessite de connaître les certificats des autorités de certification de chaque coté (+ listes de révocation).

Confidentialité - intégrité

Assurées par chiffrement de la session par un protocole symétrique
- négotiation du protocole et de la longueur des clés
- négotiation/échange d'une clé de session
- possibilité de re-négocier (renouveler) la clé en cours de session

Algorithmes :
- SSLv2 : RC4(128), RC2(123), 3DES(168), DES(56), RC4(40), RC2(40)
- SSLv3 : RC4(128), 3DES(168), DES(56), RC4(56), DES-CBC(56), RC4(40), RC2(40),

MAC (SSLv3 uniquement) : MD5, SHA1

Services sur SSL

Ports dédiés :

Protocole sécurisé	port	protocole non sécurisé	Application
HTTPS	443	HTTP	Web sécurisé
SSMTP	465	SMTP	Transport du courrier
SNNTP	563	NNTP	Transport des news Usenet
SSL-LDAP	636	LDAP	Annuaires
IMAPS	993	IMAP4	Accès aux boîtes aux lettres
SPOP3	995	POP3	Accès aux boîtes aux lettres
FTPS	889/990	FTP	Transfert de fichiers
TELNETS	992	Telnet	Connexion interactive

Au dessus d'un service existant (STARTTLS - RFC2487) :
– SMTP

SMTP et STARTTLS

La commande STARTTLS dans SMTP permet de passer en mode TLS :
- d'authentifier le serveur
- d'authentifier le client (machine nomade...)
- d'autoriser sélectivement le relayage à partir de machines authentifiées
- de chiffrer les connexions SMTP.

Inconvénients :
- Nécessite une IGC
- Charge CPU
- Mauvaises implémentations/configurations $\longrightarrow$ pertes de connectivité

IPsec

Seulement avec sites bien définis à l'avance (ou avec PKI)

Plus difficile à mettre en oeuvre que SSL

« Gratuit » si IPsec existant.

Autres tunnels

Permet des sécurisations point à point si les solutions ci-dessus ne marchent pas.

Exemple : tunnel SSH pour accès nomade.

Sécurité des boîtes à lettres - accès distants

Protocoles d'accès aux boîtes à lettres

protocoles d'accès à la boîte aux lettres :

– **POP3** ou **IMAP** : identification de l'utilisateur par mot de passe simple.

  ⇒ problème de circulation en clair du mot de passe...
  (et confidentialité /intégrité, mais ce n'est pas à ce niveau qu'il faut agir)

IMAPS - SPOP3

Chiffrement SSL de la connexion POP3 ou IMAP → confidentialité du mot de passe.
Avantage supplémentaire : authentification du serveur.

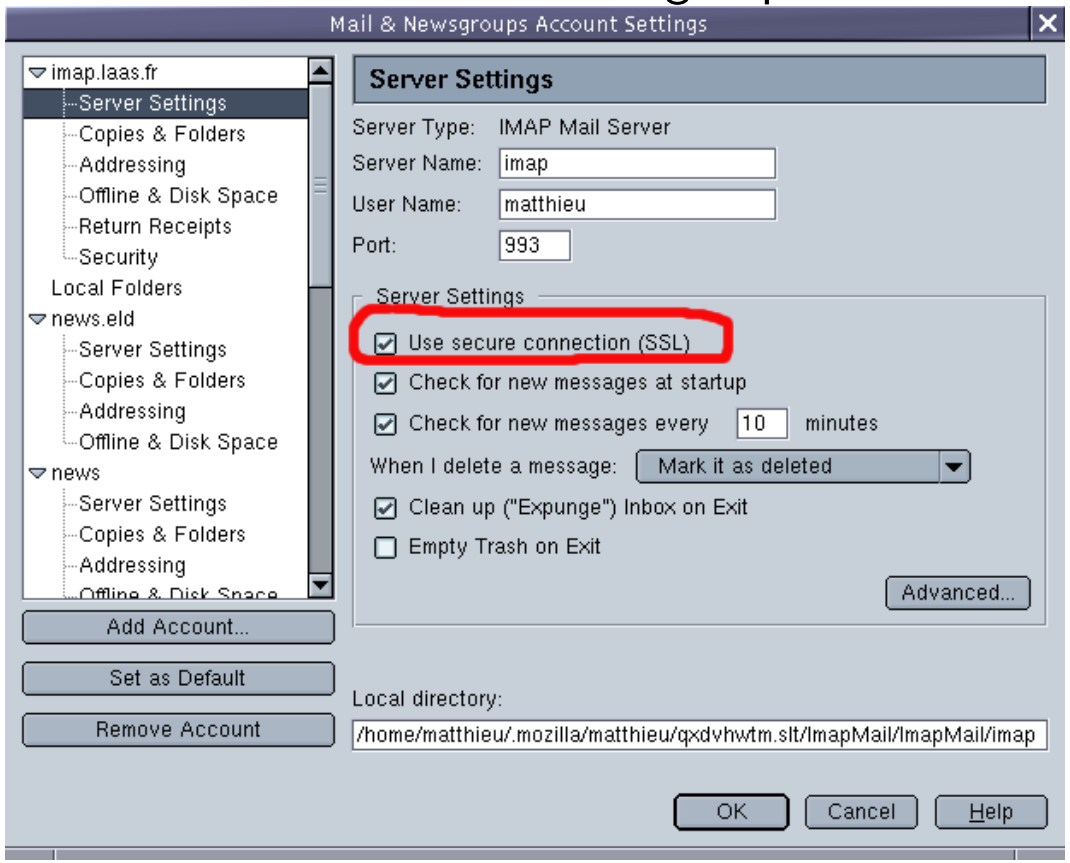
Autres solutions :

– authentification CHAP ou MS-CHAP. Pas beaucoup de clients qui supportent cette option.
– tunnel SSH (ou autre). Besoin du logiciel pour le tunnnel, mais compatible avec tous les clients POP3 ou IMAP.

Problème de charge du serveur (crypto).

Accès distant à la messagerie : Netscape

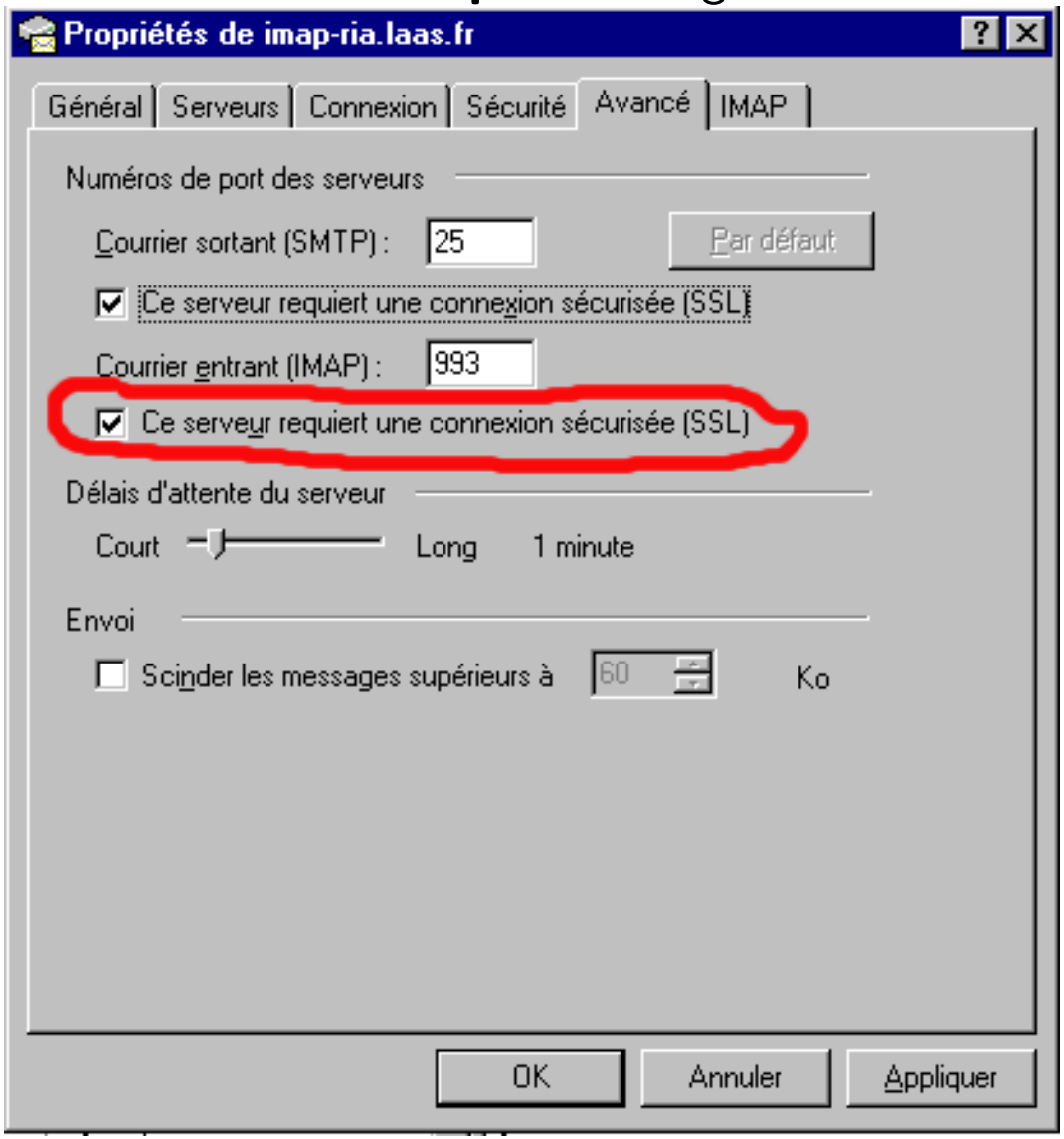
Menu Edit→Preferences→Mail & Newsgroups→Mail Servers→Edit



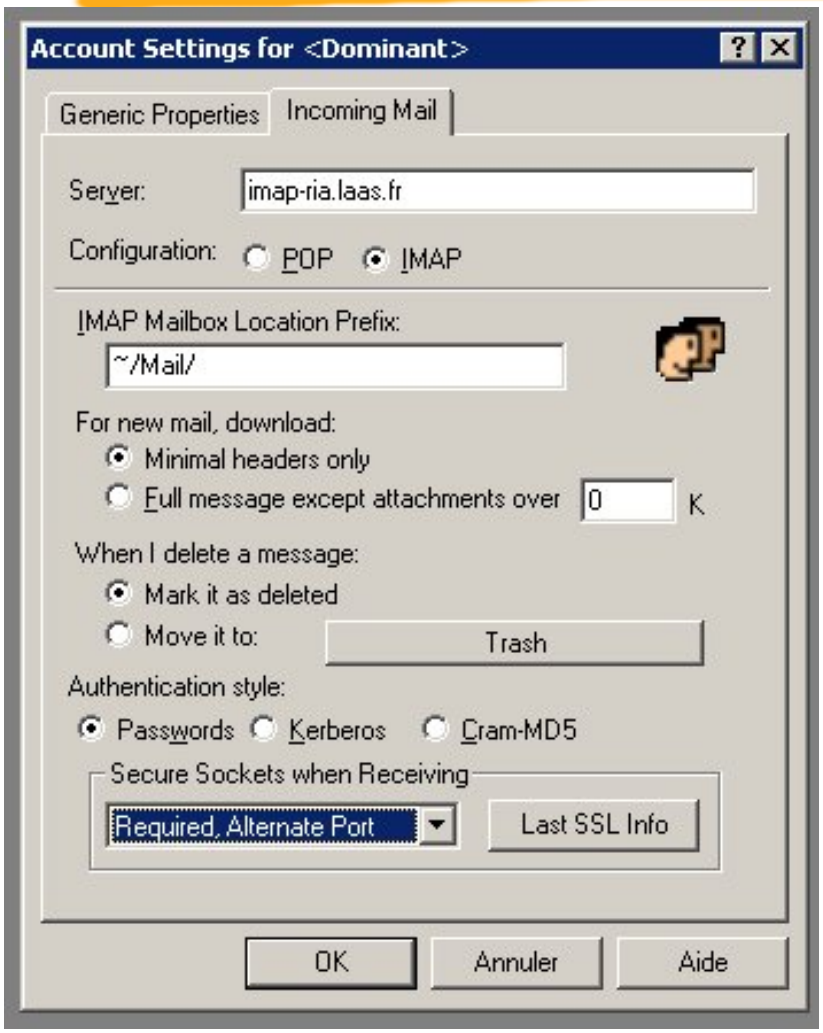
Cocher **Use secure connection (SSL)**

Accès distant à la messagerie : Outlook

Menu **Outils**→**Options** Onglet **Avancé**



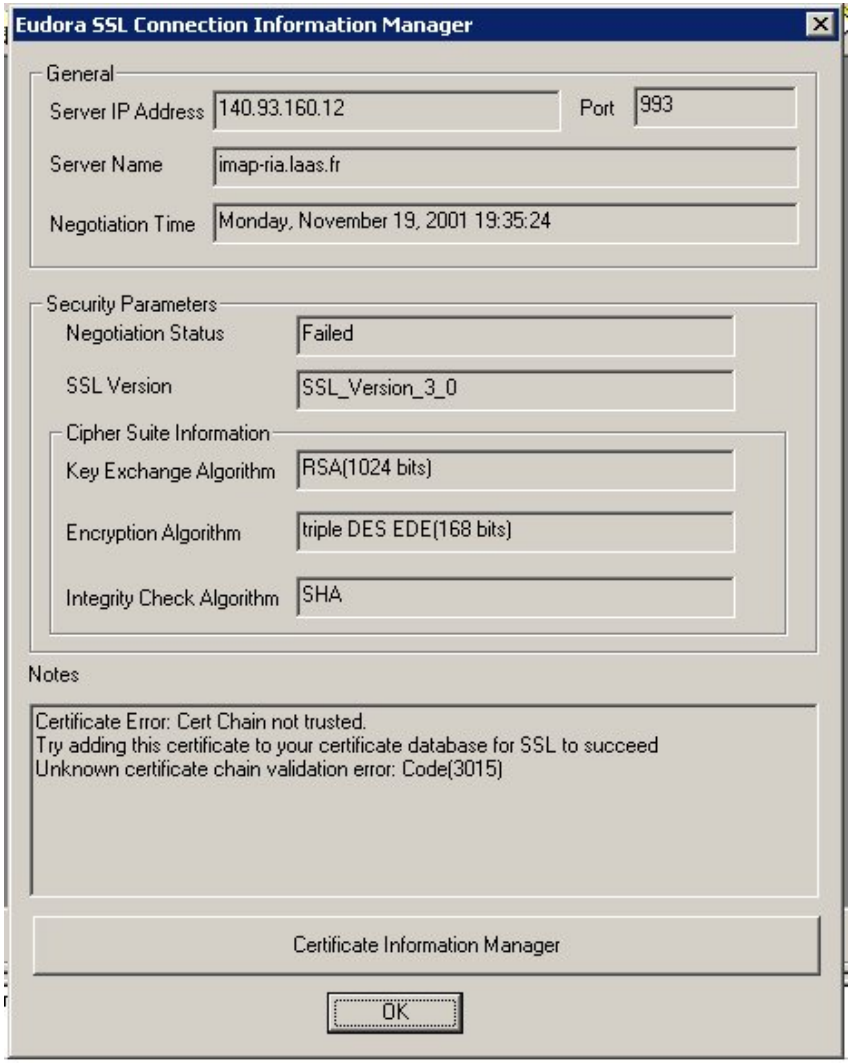
Accès distant à la messagerie : Eudora 1/2



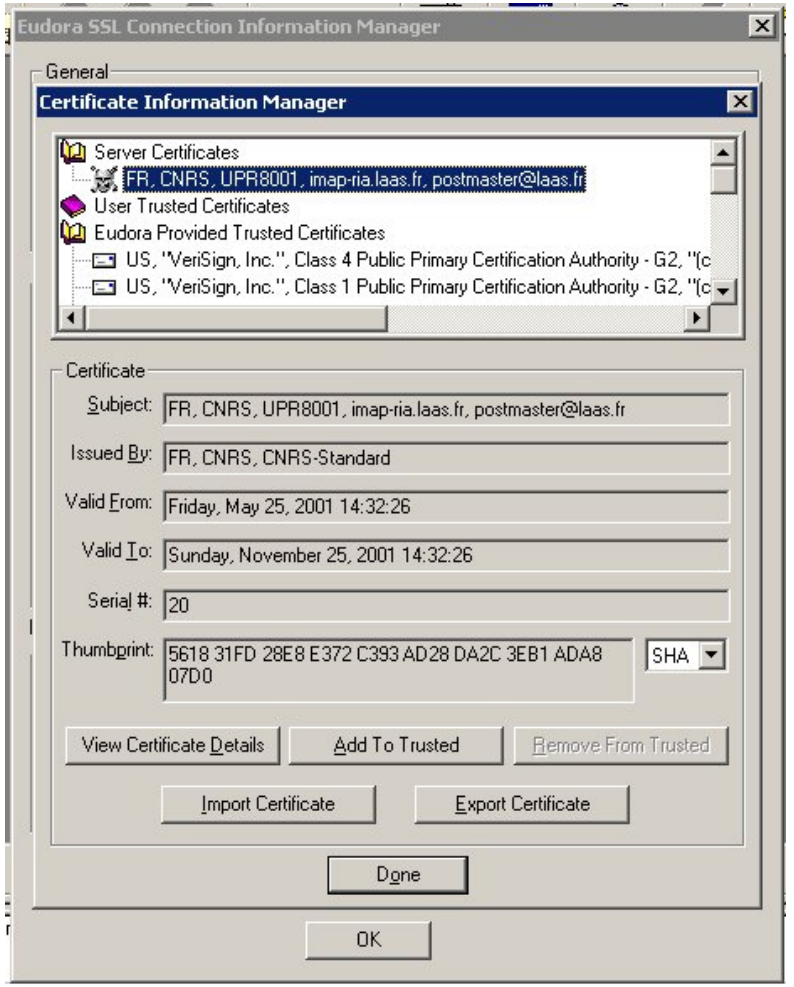
- Nécessite la version 5.2
- propriétés du compte « Dominant », onglet « incoming mail », configuration Secure Socket : **Required, alternate Port**

(Pas de SMTP+STARTTLS).

Accès distant à la messagerie : Eudora 2/2



Cliquer **Certificat Information Manager**



Sélectionner certificat
cliquer **Add to trusted**

Envoi du mail....

POP et IMAP (et les versions SSL-isées) ne gèrent que la récupération (lecture) du courrier. Pour un agent itinérant, il faut aussi un moyen d'envoyer du courier : un serveur SMTP.

Mais, pas de relayage SMTP sur les sites modernes.

sendmail AUTH - STARTTLS

Autoriser le relayage depuis une machine distante après authentification. Soit par mot de passe (AUTH) soit par certificat (STARTTLS).

sendmail SMTP AUTH

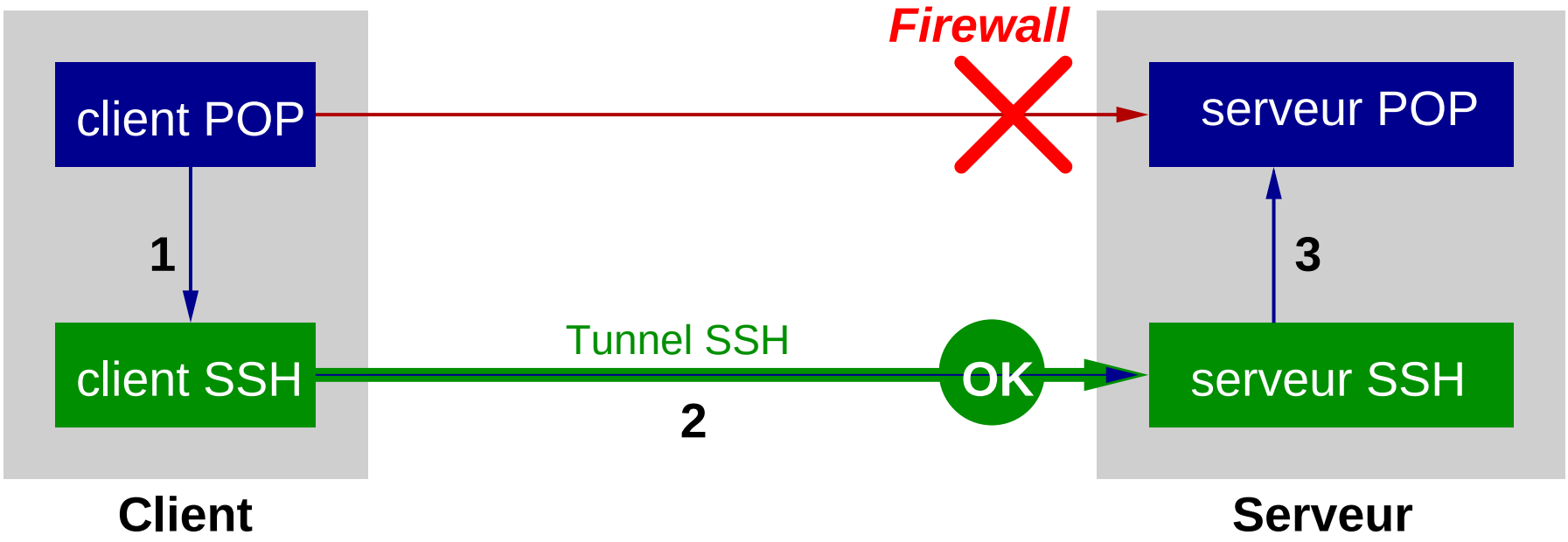
Basé sur la bibliothèque d'authentification SASL. RFC 2554.

Permet d'autoriser le relayage après authentification par mot de passe ou bien challenge/réponse (CRAM-MD5).

Clients supportés : Netscape, Outlook, Apple Mail, Eudora (?)

Problème : base d'authentification séparée...

Tunnel SSH



Commande pour créer un tunnel SMTP :

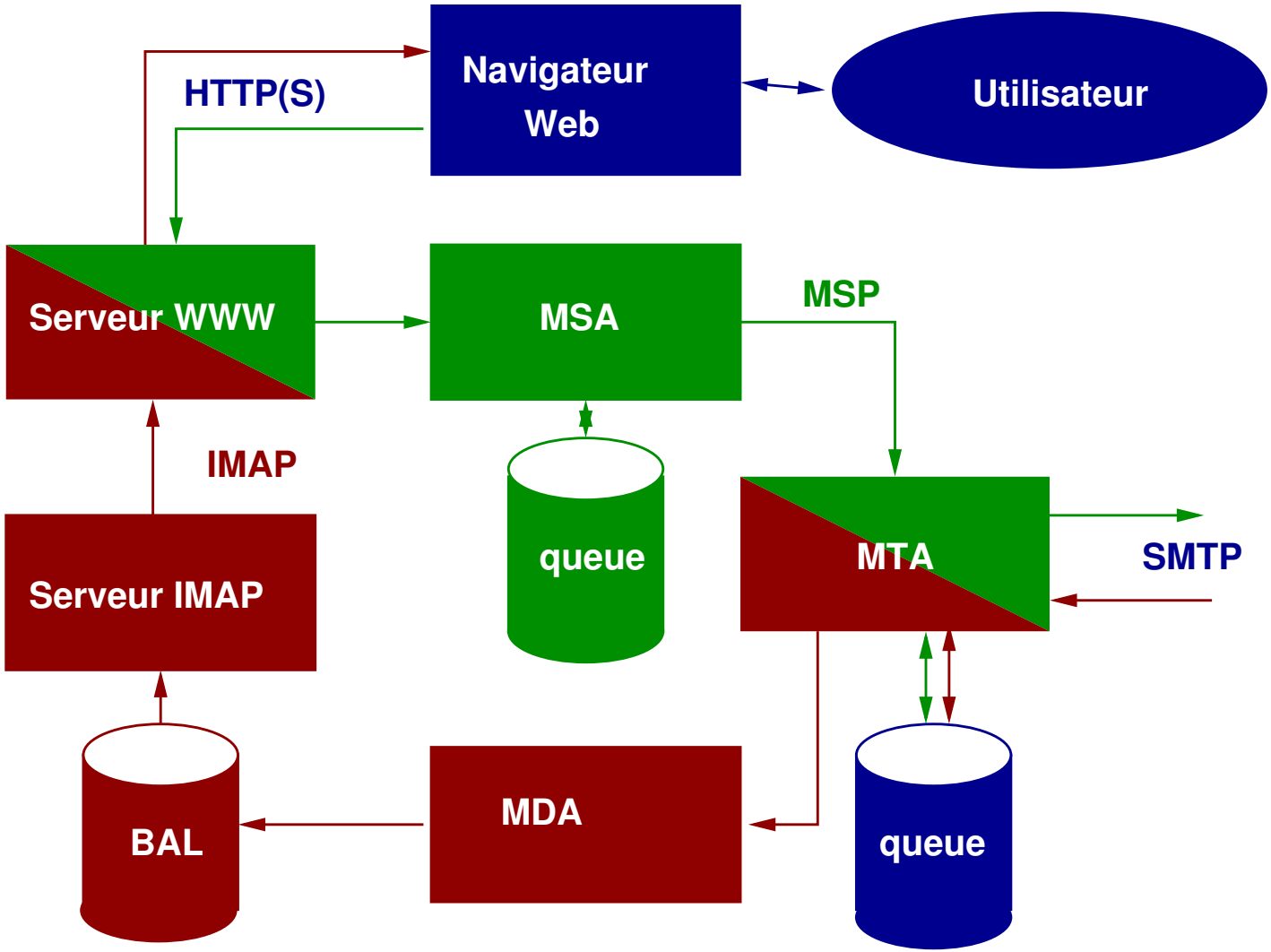
ssh -f -N -L 25 :smtpserver.example.net :25 user@sshserver.example.net

Webmail

Solution souvent la plus simple pour les utilisateurs

- ne nécessite pas de logiciel particulier coté client. La plupart des navigateurs Web suffisent (avec cookies ou javascript souvent quand même).
- coté serveur : un serveur IMAP + un serveur SMTP + un serveur https + fonctions spécifiques.
  exemple : squirrelmail : utilise PHP.
  Coté client : Cookies (obligatoires) + Javascript (optionnel - facilite la navigation si actif).

Webmail - architecture



Webmail - Limitations

- Interface lente
- Documents attachés limités aux applis du poste client
- En général pas de S/MIME ni de PGP (pb de la gestion de la clé privée)
- https utilisé surtout pour la confidentialité du mot de passe - charge CPU serveur https
- Risques si le poste client est compromis/malveillant.

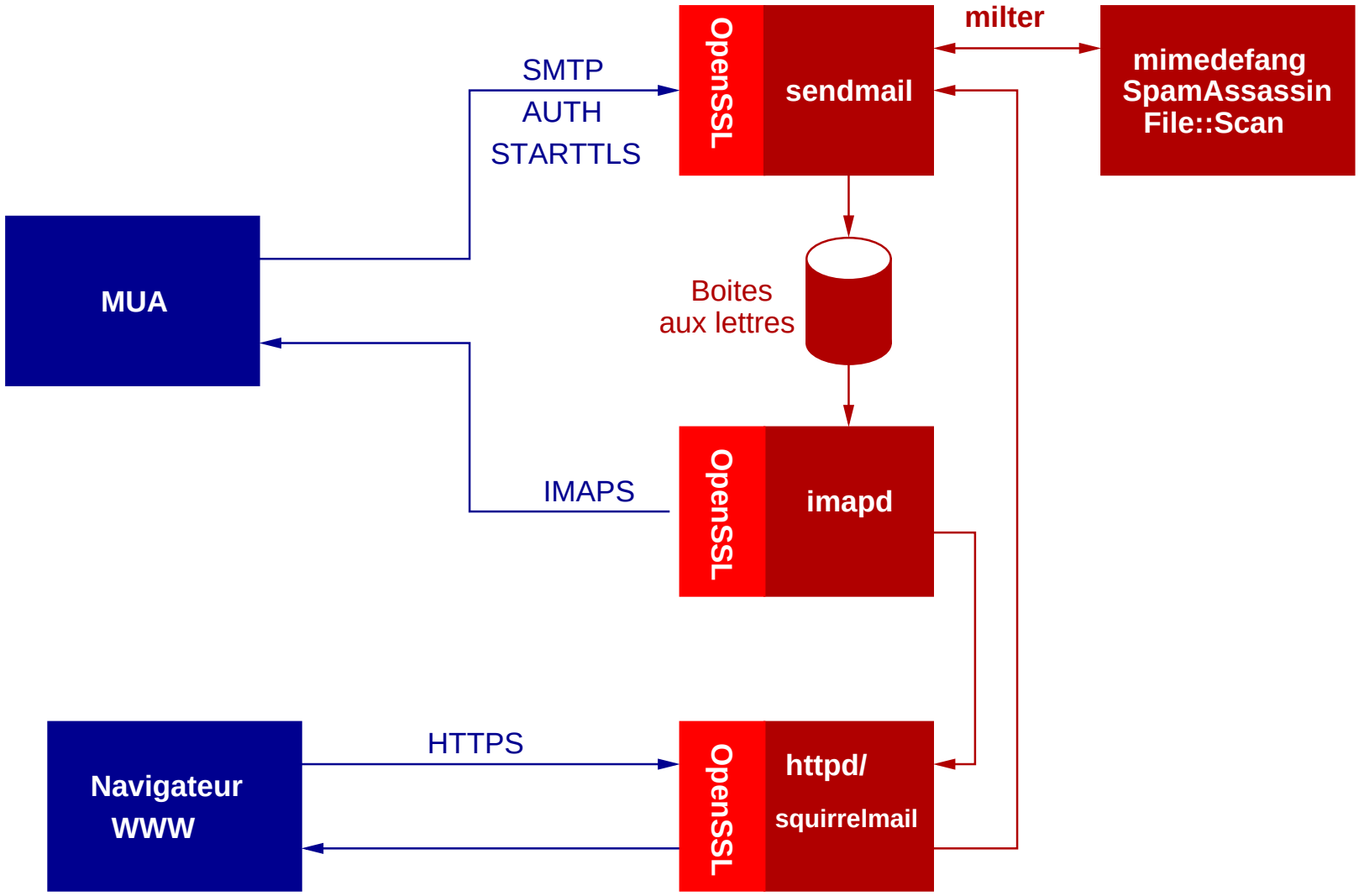
Chapitre 4

Exemple de mise en place d'un serveur

Architecture

- un serveur SMTP
- filtrage anti-virus/anti-spam
- un serveur IMAP(s)
- un webmail

Architecture (2)



Outils

OpenSSL	0.9.7c	http://www.openssl.org/
Cyrus-SASL	1.5.28	ftp://ftp.andrew.cmu.edu/pub/cyrus-mail/
Sendmail	8.12.10	http://www.sendmail.org/
mimedefang	2.38	http://www.mimedefang.org/
file :scan	0.74	http://www.cpan.org/authors/id/H/HD/HDIAS/
imapd-uw	2002d	ftp://ftp.cac.washington.edu/imap/
apache	1.3.28	http://httpd.apache.org/
mod_ssl	2.8.15	http://www.modssl.org/
php	4.3.3	http://www.php.net/
squirrelmail	1.4.2	http://www.squirrelmail.org/

OpenSSL configuration en compilation

http://www.openssl.org/

```
./config no-threads no-shared no-idea --prefix=/usr/local \
   --openssldir=/usr/local/share/openssl
make
make install
```

Supprime IDEA (algo breveté - non libre pour usage commercial)
Pas de bibliothèque partagée

OpenSSL génération de certificats auto-signés

Pour ceux qui n'ont pas d'IGC sous la main.

Possible d'acheter des certificats chez un fournisseur commercial.

Générer une paire de clés :

```
openssl genrsa -out server.key 1024
```

Générer une demande de certificat et l'auto-signer :

```
openssl req -new -key server.key -out server.csr
openssl x509 -req -days 365 -in server.csr \
    -signkey server.key -out server.crt
```

Remplir (laisser les autres champs vides) :

Country name	fr
Organization Name	example
Common Name	`server.example.com`
Email Address	`admin@example.com`

Garder **server.key** et **server.crt**.

SASL configuration et compilation

Récupérer SASL v1 : http://asg.web.cmu.edu/sasl/

```
./configure --enable-login
make
make install
```

Installation dans /usr/local/lib/sasl. Créer un lien symbolique dans /usr/bin/sasl.

Sendmail - compilation

Sendmail 8.12.10 http://www.sendmail.org/

Vérifier la signature PGP de la distribution !

Activer : MILTER, SASL et STARTTLS

configuration des options de compilation locales dans
devtools/Site/site.config.m4

```
APPENDDEF(‘conf_sendmail_ENVDEF’, ‘-DSTARTTLS -DSASL -DMILTER’)
APPENDDEF(‘conf_libmilter_ENVDEF’, ‘-D_FFR_MILTER_ROOT_UNSAFE’)
APPENDDEF(‘conf_sendmail_LIBS’, ‘-lsasl -lssl -lcrypto’)
APPENDDEF(‘confINCDIRS’, ‘-I/usr/local/include -I/usr/local/include/sasl’)
APPENDDEF(‘confLIBDIRS’, ‘-L/usr/local/lib’)
```

et

```
APPENDDEF(‘confLDOPTS’, ‘-R/usr/local/lib’)
```

ou

```
APPENDDEF(‘confLDOPTS’, ‘-Wl,-rpath /usr/local/lib’)
```

si nécessaire.

Sendmail - compilation (2)

Exécuter :

```
./Build
./Build install
```

Installer libmilter dans `/usr/local/lib`

```
cd libmilter
./Build
./Build install
```

Sendmail fichier de configuration

Dans **cf/cf/mysite.mc**

Mimedefang

```
INPUT_MAIL_FILTER(‘mimedefang’,
    ‘S=unix:/var/spool/MIMEDefang/mimedefang.sock,T=S:1m;R:1m’)
```

STARTTLS

```
define(‘CERT_DIR’, ‘MAIL_SETTINGS_DIR‘’certs’)
define(‘confCACERT_PATH’, ‘CERT_DIR’)
define(‘confCACERT’, ‘CERT_DIR/CNRS.pem’)
define(‘confSERVER_CERT’, ‘CERT_DIR/mycert.pem’)
define(‘confSERVER_KEY’, ‘CERT_DIR/mykey.pem’)
define(‘confCLIENT_CERT’, ‘CERT_DIR/mycert.pem’)
define(‘confCLIENT_KEY’, ‘CERT_DIR/mykey.pem’)
```

SASL

```
dnl Allow relaying based on SMTP AUTH
TRUST_AUTH_MECH(‘CRAM-MD5’)
```

SpamAssassin : installation

http://www.spamassassin.org Dernière version : 2.60.

Modules perl prérequis :

- `ExtUtils::MakeMaker` 6.16
- `File::Spec` 0.8
- `Pod::Usage` 1.10
- `HTML::Parser` 3.29
- `Sys::Syslog`
- `DB_File`
- `Net::DNS`

```
perl Makefile.PL
make
make install
```

SpamAssassin : configuration

Dans /etc/mail/spamassassin/sa-mimedefang.cf

```
required_hits           5
ok_locales              en fr
skip_rbl_checks 1
```

Pour utilisateurs de Eudora en versions françaises :

```
# QUALCOMM Eudora
header __FR_EUDORA \
  X-Mailer =~ /\bEudora\s+(?:(?:Pro|Light)\s+)?F[1345]\.[0-9a-z.]+\b/
# Note: uses X_LOOP and X_MAILING_LIST as subrules
meta FORGED_MUA_EUDORA  (__EUDORA_MUA && !__EUDORA_MSGID &&
  !__UNUSABLE_MSGID && !X_LOOP && !X_MAILING_LIST && !__MAC_EUDORA_MUA
  && !__OLD_EUDORA1 && !(__OLD_EUDORA2 && !__QUALCOMM) && !__FR_EUDORA)
```

File Scan : installation

http://www.cpan.org/authors/id/H/HD/HDIAS/
Dernière version : 0.74

```
perl Makefile.PL
make
make install
```

Mimedefang - installation (1/2)

Pré-requis :

- Les modules Perl suivants :
  - `MIME::tools` 5.410
  - `IO::Stringy` 1.212
  - `MIME::Base64` 2.11
  - `MailTools` 1.1401
  - `Digest::SHA1` 2.00
- Un utilisateur `defang` pour exécuter le démon sans privilèges

```
./configure
make
make install
```

Créer :

- `/etc/mail/mimedefang-filter` - règles de filtrage
- `/var/spool/MIMEDefang` (propriété de defang, mode 700).

Mimedefang - installation (2/2)

- **Lancer les démons mimedefang avant Sendmail**
  Utiliser le script fournit dans `examples/init-script`
- **Configuration de SpamAssassin pour Mimedefang**
  fichier `/etc/mail/spamassassin/sa-mimedefang.cf`
- **Relancer sendmail**

- **Maintenance**
  Après modification de mimedefang-filter, envoyer SIGHUP au processus `mimedefang-multiplexor`

TLS : installation des certificats

Remplir **/etc/mail/certs** :

- Utiliser un CNAME DNS : `mail.example.net`
- Demander un certificat de serveur CNRS-Standard
  (ou générer un certificat auto-signé).
- Copier `mail.example.net.crt` → `/etc/mail/certs/mycert.pem`
- Copier `mail.example.net.key` → `/etc/mail/certs/mykey.pem`
- Copier `CNRS.pem` et `CNRS-standard.pem` dans `/etc/mail/certs`
- Copier le contenu de `/usr/local/openssl/certs/` dans `/etc/mail/certs`

Exécuter :

```
cd /etc/mail/certs
make update
```

TLS : Contrôle du relayage par certificats

Autoriser le relayage pour les utilisateurs avec un certificat :
Éditer **/etc/mail/access** (Ex. certificats CNRS) :

```
CERTIssuer:/C=FR/O=CNRS/CN=CNRS-Standard RELAY
CERTSubject:/C=FR/O=CNRS/OU=UPR8001 RELAY
```

Exécuter :

```
cd /etc/mail
makemap dbm access < access
```

Relancer sendmail :

```
sh /etc/init.d/sendmail stop
sh /etc/init.d/sendmail start
```

TLS Test du serveur sendmail

Test :

```
telnet localhost 25
EHLO localhost
...
250-ETRN
250-STARTTLS
...
quit
```

Dans les logs :

```
Oct 17 15:03:16 mail sm-mta[19841]:  [ID 702 mail.info]
STARTTLS=server, relay=test.example.net [192.9.200.20],
version=TLSv1/SSLv3, verify=OK, cipher=RC4-MD5, bits=128/128
```

SASL - configuration

Utiliser `saslpasswd` pour créer des utilisateurs. Exemple :

```
saslpasswd -u mail.example.net alice
```

Configuration du serveur IMAPS

Avec le serveur IMAP de l'Université de Washington :

http://www.washington.edu/imap/
ftp://ftp.cac.washington.edu/imap/imap-2002e.tar.Z

Utilise les boîtes à lettres format Unix sur le serveur,

dans les répertoires des utilisateurs.

Avantages :
- pas de modification du serveur SMTP
- cohabitation avec tous types de clients de messagerie
- cohabitation avec un serveur POP possible

Petits problèmes :
- pas très rapide (problème avec grosses boîtes à lettres)
- quelques soucis de verrouillage si utilisé avec NFS

Compilation / installation

Lire `docs/SSLBUILD`

Éditer `src/osdep/unix/Makefile` :

```
SSLDIR=/usr/local
SSLCERTS=/etc/ssl
```

Voir `Makefile` pour cibles. `lnp` → Linux PAM.

```
make SSLTYPE=unix lnp
```

Installation :

```
cp imapd/imapd /usr/local/sbin/imapd
```

Configuration imapd

- Éditer **inetd.conf** :
  `imaps stream tcp nowait root /usr/local/sbin/imapd imapd`
- Créer **/etc/ssl/imapd.pem** :
  Utiliser un nom générique (`imap.example.net` par ex.)
  → définir un CNAME dans le DNS.
- Demander un certificat serveur de l'autorité de certification CNRS-Standard pour ce nom générique :
  http://igc.services.cnrs.fr/CNRS-Standard/certificats.html
  (Ou créer un certificat auto-signé).
- Récupérer les 2 fichiers `.key` et `.crt`
- Supprimer les commentaires du fichier `.crt`
- Concaténer les 2 fichiers :

```
cat imap.example.net.key imap.example.net.crt > /etc/ssl/imapd.pem
chmod 400 /etc/ssl/imapd.pem
```

Extensions possibles

- MX secondaire
- Autres filtres
- Boîtes à lettres séparées
- Gestionnaire de liste de diffusions - Sympa
- Annuaire LDAP
- ...

Conclusion

Messagerie - Géant aux pieds d'argile

Solutions pour la sécurisation :
- Filtrage
- Outils cryptographiques : signature/chiffrement

Solutions légales ?

Nouvelles technologies : messagerie instantanée / téléphonie mobile $\longrightarrow$ vers la fin de la messagerie SMTP ?

Bibliographie

- R. Costales & E. Allman **Sendmail, 3rd edition**, O'Reilly.
- R. Blum, **Postfix**, SAMS.
- E. Rescorla, **SSL and TLS, designing and building secure systems**, Addison Wesley.
- M. Bauer, **Building secure servers with Linux**, O'Reilly.
- P. Graham - A plan for spam
  http://www.paulgraham.com/spam.html
- La page sur les virus du Fonctionnaire de défense du CNRS :
  http://www.cnrs.fr/Infosecu/Virus.html
- Fiches du groupe de travail Accès Distants Sécurisés du CNRS :
  https://www.services.cnrs.fr/corres-secu/
- R. Dirlewanger, Tutorial SSL, JRES 1999 Montpellier :
  http://www.dr15.cnrs.fr/Cours/JRES99/rd-ssl.pdf