Accès distants sécurisés

Matthieu Herrb

Octobre 2002

Chapitre 1

Introduction

À propos de l'insécurité

TCP/IP n'est pas un protocole sûr :

- pas d'authentification des connexions,
- pas de confidentialité,
- pas de garantie d'intégrité forte

$\Rightarrow$ à éviter dans un environnement réseau « hostile »

Les types d'accès à distance

Accès à distance = accès depuis adresse IP $\notin$ réseau du labo :
- fournisseur d'accès perso (PPP, ADSL, etc.)
- machine sur le réseau d'un autre labo, d'un industriel,. . .

Trois catégories d'utilisateurs nomades :
- avec son portable, ou depuis son domicile avec un accès ADSL,
- depuis une station d'un autre labo,
- accès depuis un cybercafé ou équivalent.

Les quatre grands types d'accès :
- Accès à la messagerie,
- Transferts de fichiers,
- Connexion interactive,
- Accès intranet.

[Une connexion par le serveur PPP du labo n'est pas un accès à distance. . .]

Solutions

Il existe des solutions basées sur la cryptologie.

À plusieurs niveaux :
- réseau : **IPSec**
- transport : **SSL**
- applicatif : **SSH**

Problématique : quelle(s) solution(s) pour quel(s) besoin(s) ?

Recouvrement avec le déploiement des certificats numériques au CNRS ?

Synthèse des solutions

Type d'accès	Mail	Fichiers	Connexion	Intranet
Portable / Machine à domicile	IMAPS SMTP/SSL Tunnels SSH	scp/sftp FTP/SSL	slogin	HTTPS + Certificat
Machine dans un labo	WebMail Tunnels SSH	scp/sftp	slogin	utilisateur & mot de passe
Cybercafé	WebMail	WebFTP	Applet SSH	utilisateur & mot de passe

Chapitre 2

SSL - Connexions sécurisées

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

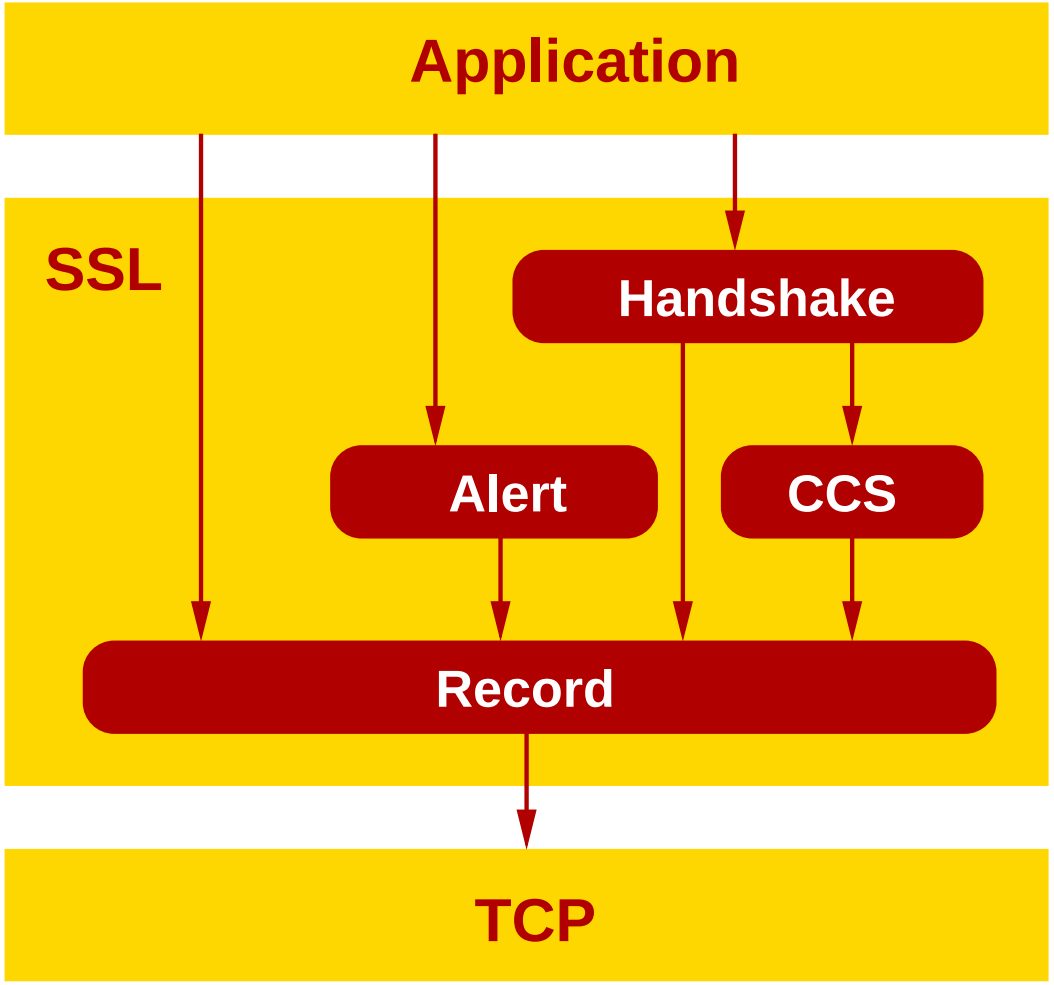
Secure **S**ocket **L**ayer

Une couche au dessus de TCP qui assure :

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

SSLv1	obsolete
SSLv2	Netscape
SSLv3	Netscape
TLSv1	RFC 2246

Fonctionnement de SSL



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 boites aux lettres
SPOP3	995	POP3	Accès aux boites aux lettres
FTPS	889/990	FTP	Transfert de fichiers
TELNETS	992	Telnet	Connexion interactive

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

Authentification

Utilise des certificats x509v3

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

Assureés par chiffrement de la session par un protocole symétrique
- négotiation du protocole et de la longueur des clés
- négotiation/echange d'un 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

SSL - implémentations

- OpenSSL (http://www.openssl.org/)
- Java Security API
- Microsoft Crypto API
- . . .

Produits utilisant SSL :

- **Navigateurs :** Netscape navigator, Internet Explorer, Opera,. . .
- **Clients messagerie :** Netscape messenger, Outlook (Express), Eudora 5, pine, kmail, evolution,. . .
- **Seveurs Web :** mod_ssl (apache), ApacheSSL, IIS,
- **Serveurs IMAP :** imap-uw, cyrus,
- **Serveurs SMTP :** sendmail, postfix,
- **Langages :** C, C++, Java, perl, python, php,...
- **Tunnels TCP/IP :** stunnel

Points à surveiller

- failles dans les implémentations
  - utiliser OpenSSL $>$ 0.9.6e
  - vérification effective des certificats (cf bug KDE)
  - . . .
- la sécurité d'un crypto-système dépend de la sécurité du sysème qui l'héberge.
  - bibliothèques partagées
  - sécurité des clés privées
  - . . .
- mettre à jour régulièrement les listes de révocation
- renouveler les certificats
- comment distribuer le certificat de l'autorité de certification CNRS vers les clients ?
- connexions chiffrés : pas de contrôle possible du contenu

Chapitre 3

Installation d'OpenSSL

Installation - compilation

http://www.openssl.orgl

– Version courante : 0.9.6g
– Ou bien version avec patches de sécurité (RedHat)

Pré-requis : perl 5

Compilation maison :

```
./config no-idea --prefix=/usr/local --openssldir=/usr/local/openssl
make
make test
make install
```

Utilisation d'OpenSSL

Commandes **openssl** :

genrsa	Génération d'un couple de clés RSA
x509	gestion de certificats x509
pkcs12	import/export de certificats au format PKCS#12
smime	signature/chiffrement de messages S/MIME
verify	vérification de certificats x509
enc	chiffrement/déchiffrement de fichiers
s_server	création d'un serveur SSL
s_client	création d'un client SSL

Exemples

- conversion d'un certificat exporté d'un navigateur vers le format PEM :
  `openssl pkcs12 -in moncert.p12 -clcerts -out moncert.pem`
- visualiser un certificat au format PEM :
  `openssl x509 -in moncert.pem -text -noout`
- récupérer les certificats de l'autorité de certification CNRS et les convertir au format PEM :

```
wget -O cnrs.der \
  'http://igc.services.cnrs.fr/cgi-bin/viewca?cmd=load\
  &CA=CNRS-Standard&ca=CNRS'
openssl -inform DER -in cnrs.der -out CNRS.pem -trustout
wget -O cnrs-standard.der \
  'http://igc.services.cnrs.fr/cgi-bin/viewca?cmd=load\
  &CA=CNRS-Standard&ca=CNRS-Standard'
openssl x509 -inform DER -in cnrs-standard.der \
   -out CNRS-Standard.pem -trustout
```

Chapitre 4

Sécurisation de la messagerie avec SSL

Accès distant à la messagerie

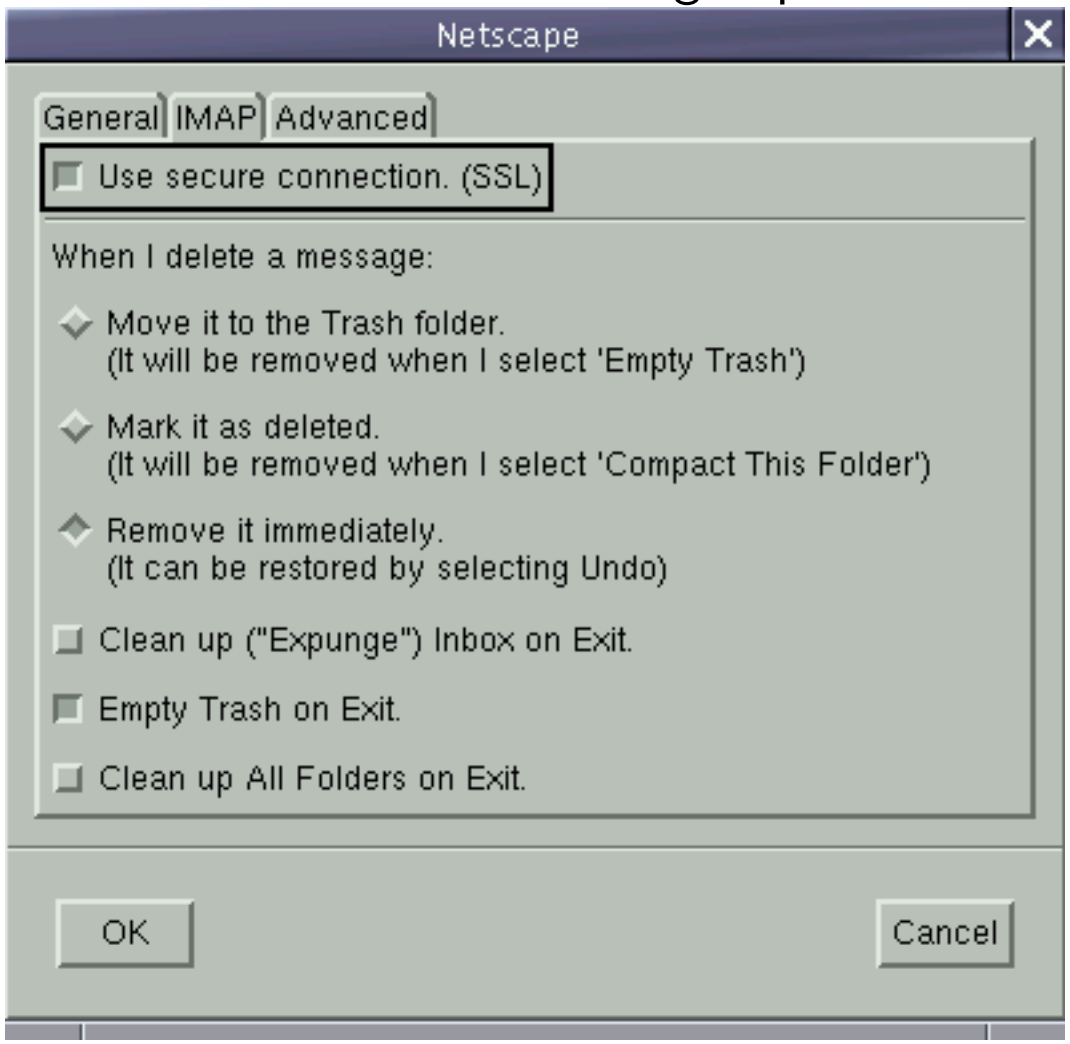
Principe :

- utilisation du protocole IMAP :
  - le courrier reste sur le serveur
  - possibilité de gérer plusieurs boîtes à lettres
  - ne télécharge pas le corps du message si on ne veut pas
- utilisation du chiffrement SSL :
  - même principe de chiffrement que HTTPS
  - la connexion entre le client et le serveur est chiffrée $\Rightarrow$ le mot de passe est protégé
  - configuration simple dans Netscape, Outlook, Eudora 5.2.
- authentification par certificat pour l'envoi (STARTTLS).
  - nécessite un certificat CNRS
  - l'authentification permet d'éviter l'utilisation du serveur mail du laboratoire comme relais de SPAM.

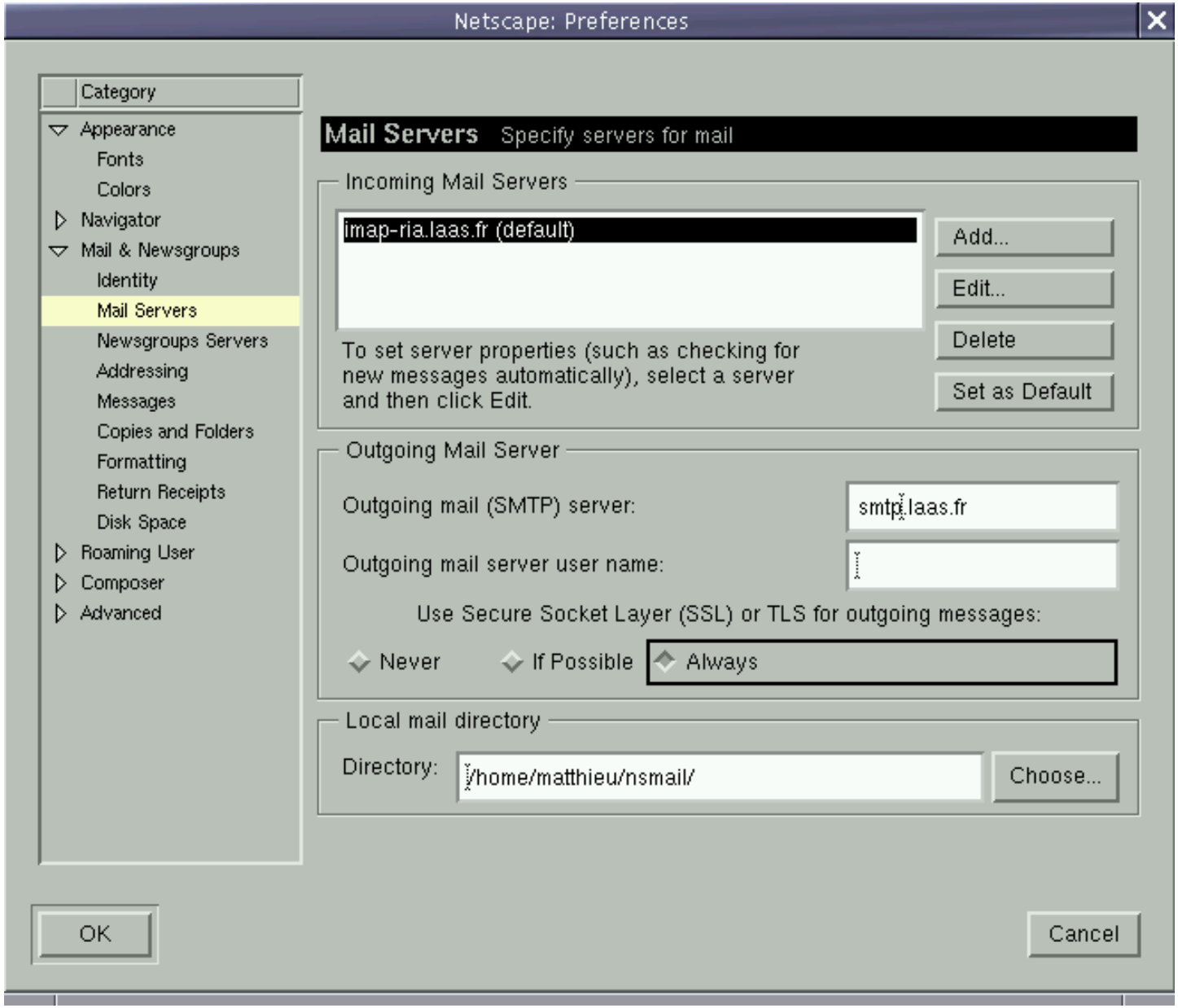
[Ne pas confondre avec S/MIME]

Accès distant à la messagerie : Netscape

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



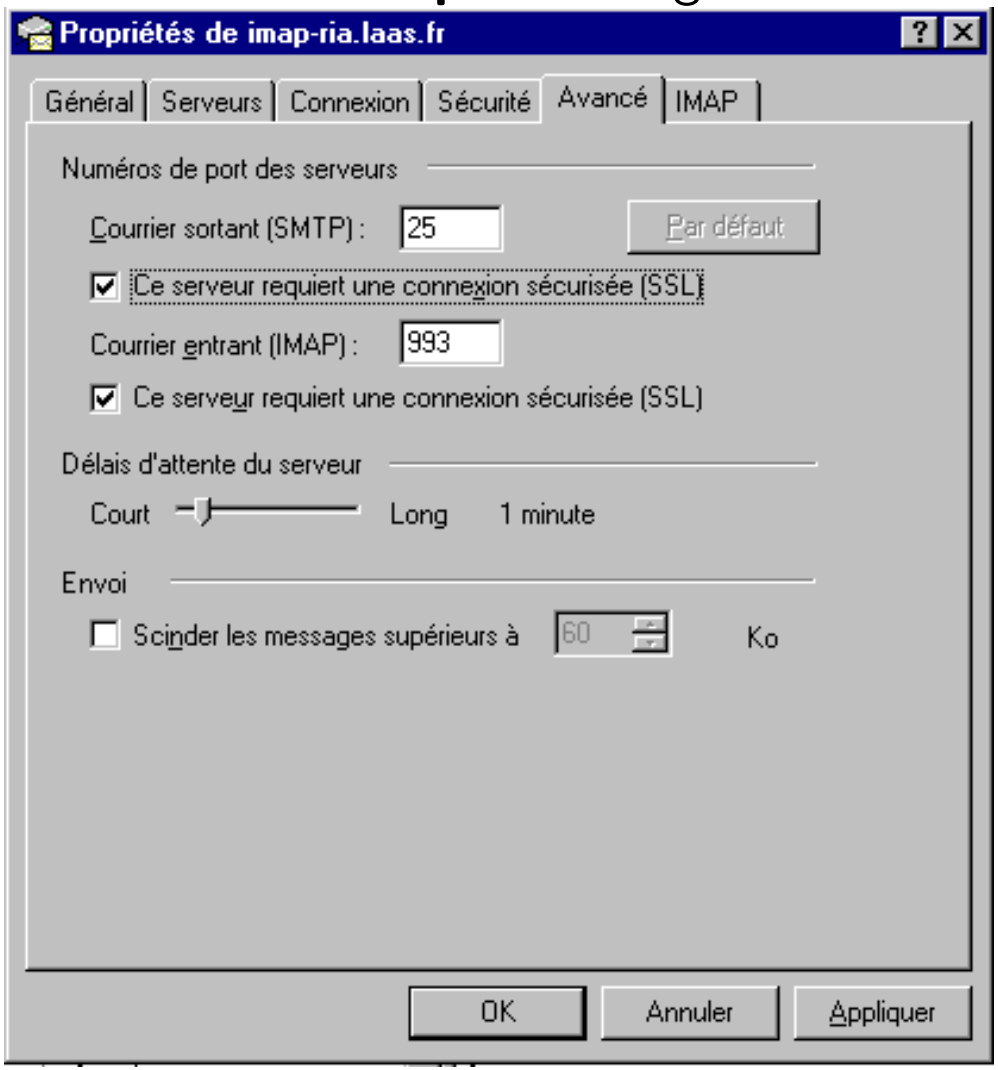
Cocher **Use secure connection (SSL)**



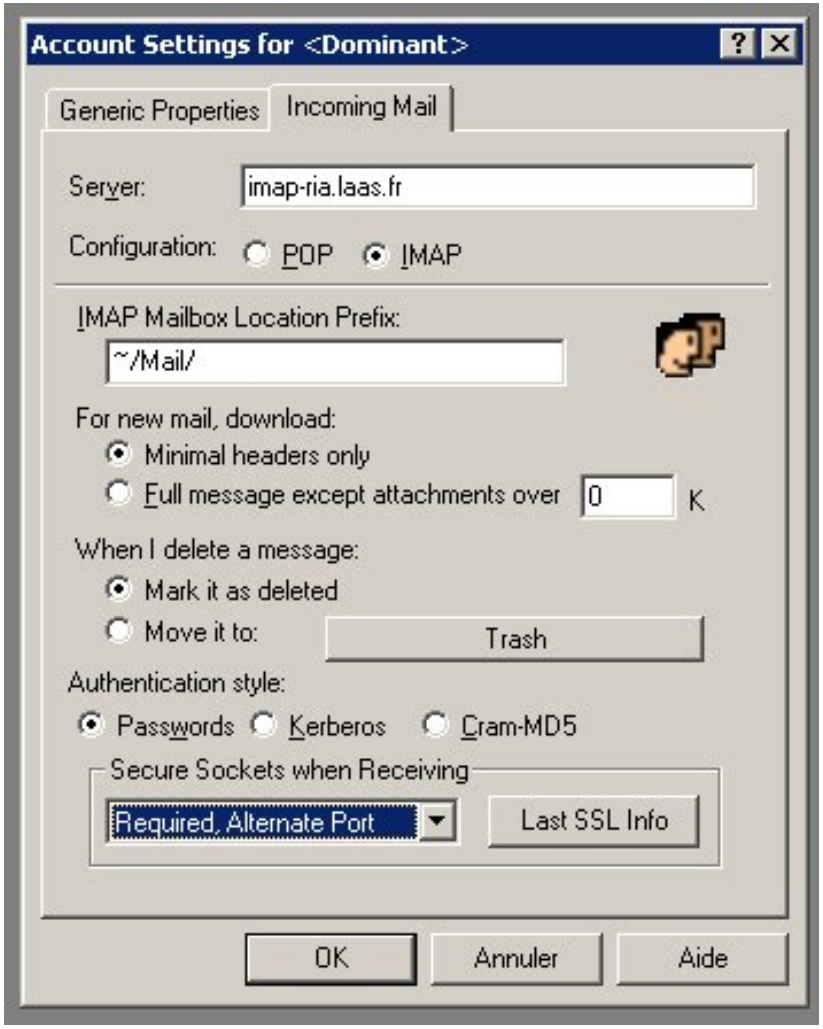
Cocher **Use SSL or TLS for outgoing messages** : **Always**

Accès distant à la messagerie : Outlook

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



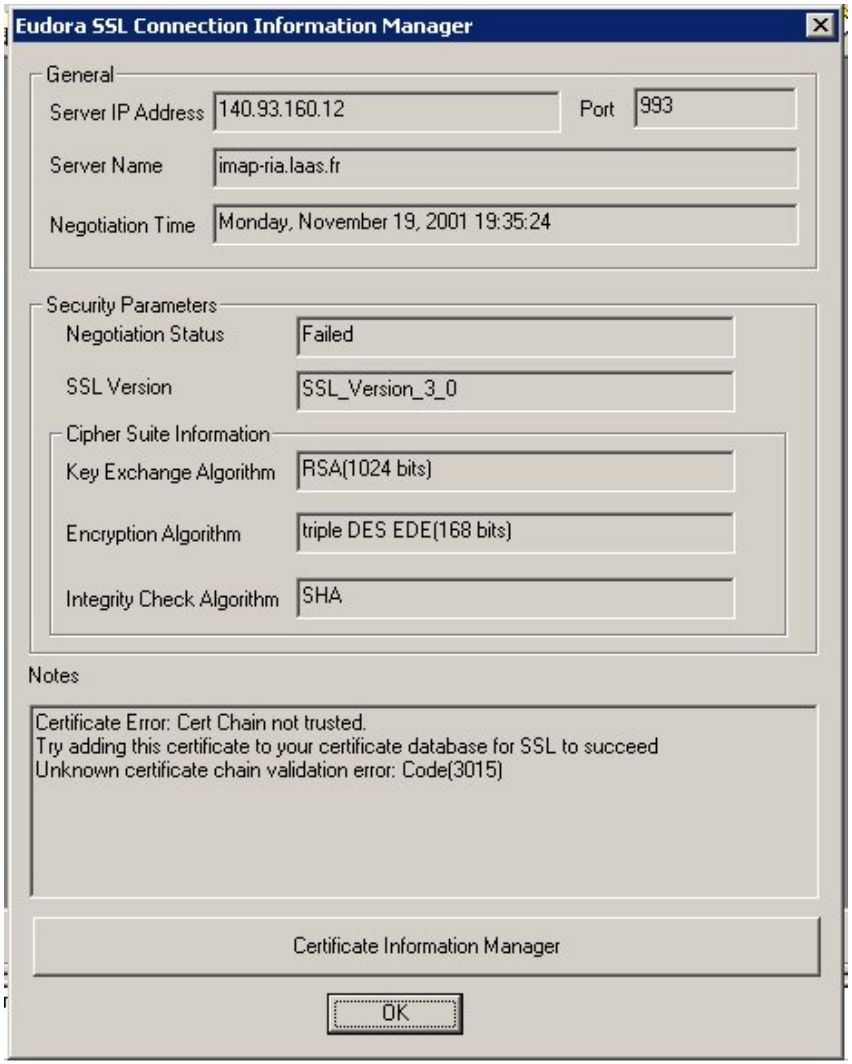
Accès distant à la messagerie : Eudora 1/2



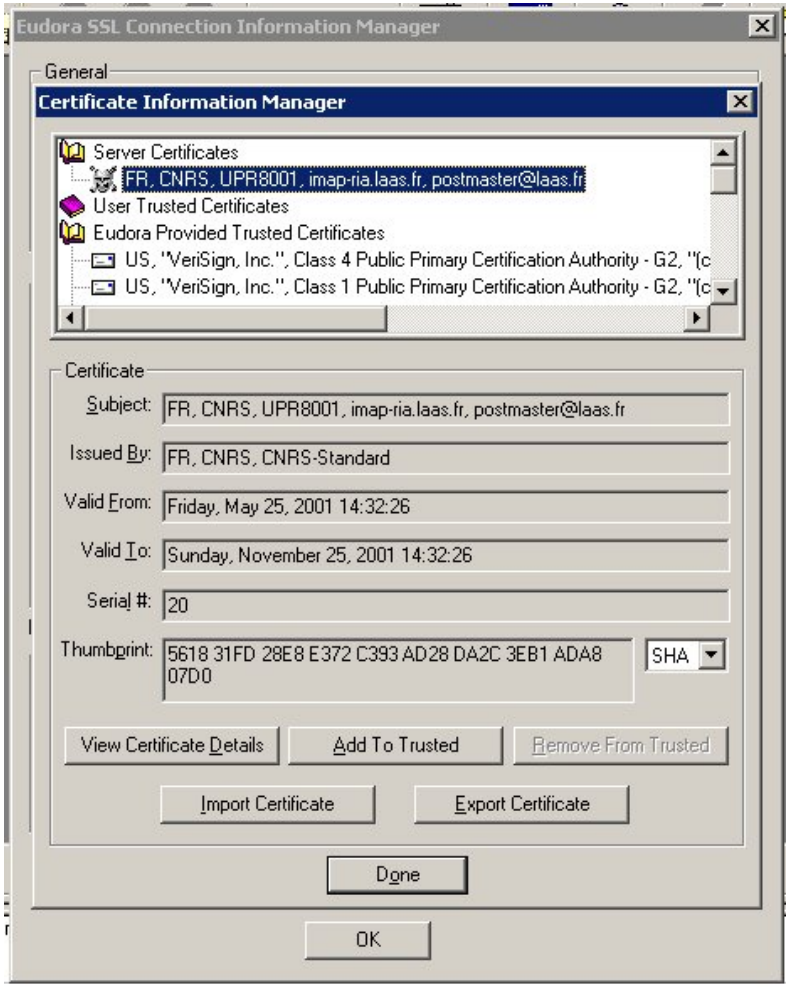
- Nécessite la version 5.
- propriétés du compte « Dominant », onglet « incoming mail », configuration Secure Socket : **Required, alternate Port**

Ne permet pas l'envoi de message par le serveur du laboratoire (Pas de SMTP+STARTTLS).

Accès distant à la messagerie : Eudora 2/2



Cliquer **Certificat Information Manager**



Sélectionner certificat
cliquer **Add to trusted**

Configuration serveur : IMAPS

Avec le serveur IMAP de l'Université de Washington :

http://www.washington.edu/imap/
ftp://www.cac.washington.edu/imap/imap-2002.RC8.tar.Z

Utilise les boîtes à lettre 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 pour le serveur (`imap.monlabo.fr` par ex.)
$\rightarrow$ 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

Récupérer les 2 fichiers `.key` et `.crt`

Supprimer les commentaires du fichier `.crt`

Concaténer les 2 fichiers :

```
cat imap.monlabo.fr.key imap.monlabo.fr.crt > /etc/ssl/imapd.pem
chmod 400 /etc/ssl/imapd.pem
```

Configuration serveur : sendmail

http://www.sendmail.org

Le protocole STARTTLS permet :
- d'authentifier le serveur SMTP
- d'authentifier le client SMTP (la machine nomade)
- d'autoriser sélectivement le relayage à partir de machines authentifiées
- de chiffrer les connexions SMTP avec les autres serveurs qui acceptent l'option STARTTLS.

Compilation de sendmail avec support STARTTLS

Pré-requis :
- OpenSSL 0.9.6g
- Sendmail 8.12.6

Compilation :

Ajouter dans **devtools/Site/site.config.m4**

```
APPENDDEF(‘conf_sendmail_ENVDEF’, ‘-DSTARTTLS’)
APPENDDEF(‘conf_sendmail_LIBS’, ‘-lssl -lcrypto’)
APPENDDEF(‘confINCDIRS’, ‘-I/usr/local/include’)
APPENDDEF(‘confLIBDIRS’, ‘-L/usr/local/lib’)
APPENDDEF(‘confLDOPTS’, ‘-R/usr/local/lib’)
```

Exécuter :

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

Configuration sendmail (1/3)

Sendmail.cf :

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

```
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’)
```

Configuration serveur : sendmail (2/3)

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

- Utiliser un CNAME DNS : `mail.monlabo.fr`
- Demander un certificat de serveur CNRS-Standard
- Copier `mail.monlabo.fr.crt` → `/etc/mail/certs/mycert.pem`
- Copier `mail.monlabo.fr.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

cd sendmail-8.12.6/cf/cf
make mysite.cf
cp mysite.cf /etc/mail/sendmail.cf
```

Configuration serveur : sendmail (3/3)

Autoriser le relayage pour les utilisateurs avec un certificat :
Éditer **/etc/mail/access** :

```
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
```

Test du serveur sendmail

Test :

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

Dans les logs :

```
Oct 17 15:03:16 laas sm-mta[19841]: [ID 702 mail.info] START-TLS=server, relay=jake.laas.fr [140.93.160.20], version=TLSv1/SSLv3, verify=OK, cipher=RC4-MD5, bits=128/128
```

Chapitre 5

Securisation du Web avec SSL

Apache + Mod_SSL

http://httpd.apache.org http://www.modssl.org

Implémentent **https**

- Authentification du serveur
- Authentification optionnelle du client
- Chiffrement de la connexion

Utilisation :
- sécuriser l'authentification basique HTTP (mot de passe)
  $\rightarrow$ chiffrée
- sécuriser des applications qui lisent un mot de passe par un formulaire (Webmail)
- authentifier les utilisateurs par leur certificat client.

Apache + Mod_SSL : compilation (1/2)

Pré-requis :
- Perl 5.6.0
- OpenSSL 0.9.6g
- mm-1.2.1 (http://www.ossp.org/pkg/lib/mm/)
- mod_ssl 2.8.11
- apache 1.3.27

Configurer mod_ssl :

```
tar xzvf apache_1.3.27.tar.gz
tar xzvf mod_ssl-2.8.11-1.3.27.tar.gz
cd mod_ssl-2.8.11-1.3.27
./configure --with-apache=../apache_1.3.27 --with-ssl=../openssl-0.9.6g
  --with-mm=../mm-1.2.1 --prefix=/usr/local/apache
```

Apache + Mod_SSL : compilation (2/2)

Configurer et compiler apache :

```
cd ../apache_1.3.27
SSL_BASE=../openssl-0.9.6g EAPI_MM=../mm-1.2.1 ./configure \
  --enable-module=ssl --prefix=/usr/local/apache
make
make install
```

Configuration certificat serveur

Utiliser un CNAME DNS : `www.monlabo.fr`

- Demande sur
  `http://igc.services.cnrs.fr/CNRS-Standard/certificat.html`
- Copier `www.monlabo.fr.crt` → `conf/ssl.crt/server.crt`
- Concaténer `CNRS.pem` et `CNRS-Standard.pem` dans `ca.crt` :
  `cat CNRS.pem CNRS-Standard.pem >/conf/server.crt/ca.crt`
- Copier les autres certificats d'autorités de certification (SSI, CNRS-Plus,. . .) dans `server.crt/`
- Exécuter :
  `cd conf/server.crt`
  `make update`
- Copier `www.monlabo.fr.key` → `conf/ssl.key/server.key`
- Exécuter :
  `chmod 400 conf/ssl.key/server.key`

Configuration du serveur

Dans `conf/httpd.conf` :

```
LoadModule ssl_modules /usr/lib/apache/libssl.so
AddModule mod_ssl.c

Listen 80
Listen 443

SSLSessionCache         dbm:/var/run/ssl_cache
SSLSessionCacheTimeout  300
SSLMutex                file:/var/run/ssl_mutex
SSLLog                  /var/www/logs/ssl_engine_log
SSLLogLevel             info
SSLRandomSeed           startup builtin
SSLRandomSeed           connect builtin
```

Configuration du serveur (2)

Dans `conf/httpd.conf` :

```
<VirtualHost _default_:443>
DocumentRoot            /usr/local/apache/htdocs-ssl
ServerName              www.monlabo.fr
ServerAdmin             you@your.address
ErrorLog                logs/error_log
TransferLog             logs/access_log
SSLEngine               on
SSLCertificateFile      conf/ssl.key/server.crt
SSLCertificateKeyFile   conf/ssl.key/server.key
SSLCertificateChainFile conf/ssl.crt/ca.crt
SSLCACertificatePath    conf/ssl.crt
</VirtualHost>
```

Certificats des clients

Exemple de controle d'accès par certificat client à `/intranet`.
Dans `conf/httpd.conf` :

```
<Files ~ "\.(cgi|shtml|phtml|php3?)$">
    SSLOptions +StdEnvVars
</Files>

<Location /intranet>
    SSLVerifyClient require
    SSLVerifyDepth  5
    SSLRequireSSL
    SSLRequire %{SSL_CLIENT_S_DN_C} eq "FR" and \
               %{SSL_CLIENT_S_DN_O} eq "CNRS" and \
               %{SSL_CLIENT_S_DN_OU} eq "UPR8001"
</Location>
```

Gestion de la liste de révocation

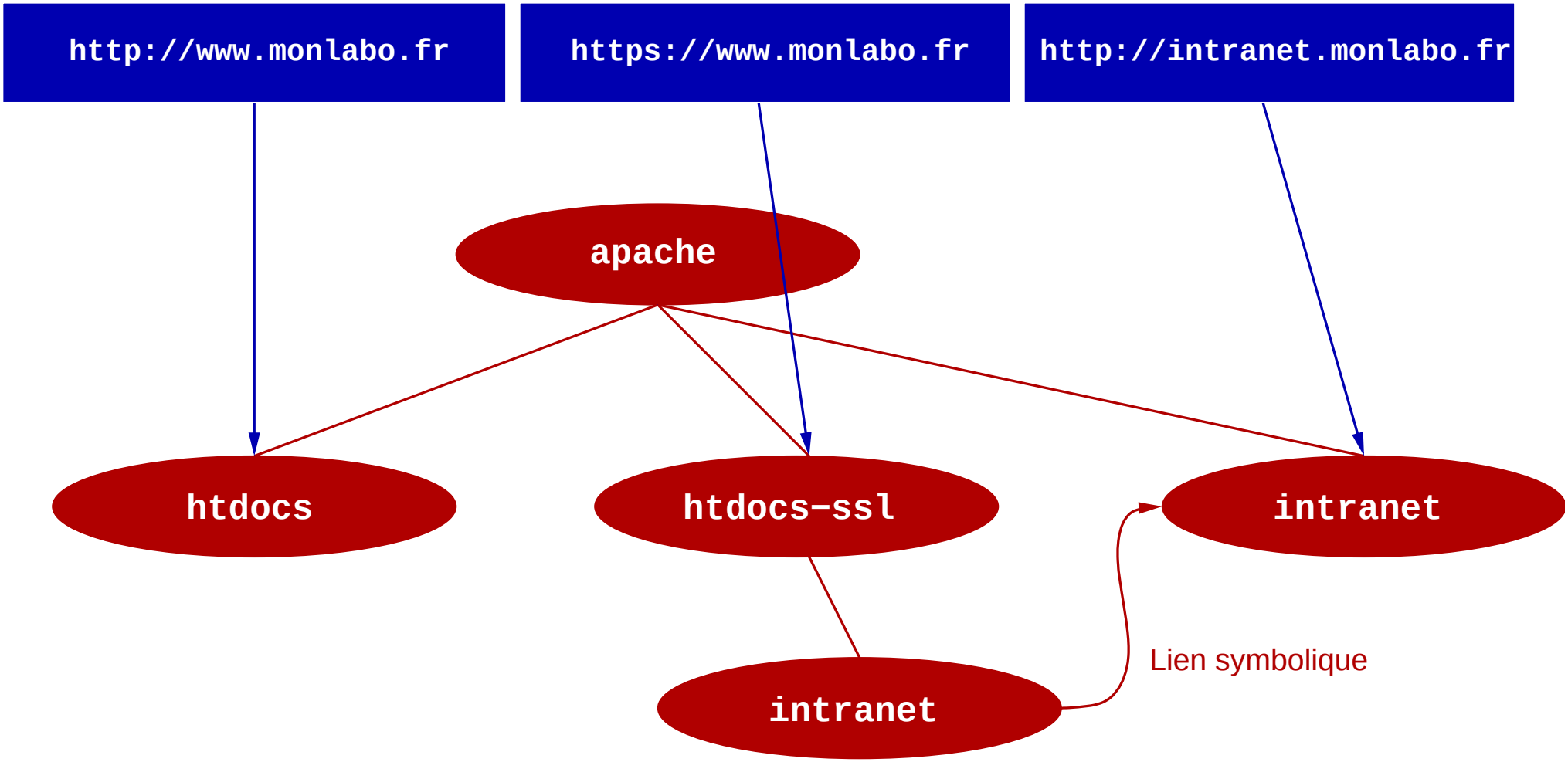
Si autorisations par certificats clients : gérer la liste de révocation.

Script get-crl à exécuter chaque jour avec cron :

```
#!/bin/sh
cd /usr/local/local/apache/conf/ssl.crl
wget -q -O CNRS.crl "http://igc.services.cnrs.fr/cgi-bin/viewcrl?cmd=save&ca=CNRS"
wget -q -O CNRS-Standard.crl \
  "http://igc.services.cnrs.fr/cgi-bin/viewcrl?cmd=save&ca=CNRS-Standard"
make update
exit 0
```

[À compléter avec les CRL d'autres CA si nécessaire]

Exemple de plan de site



Les scripts CGI et SSL

Requêtes chiffrées → permet de faire circuler des données confidentielles dans les formulaires HTML.

SSL rend les variables suivantes accessibles aux scripts :

SSL_CLIENT_S_DN	/C=FR/O=CNRS/OU=UPR8001/ CN=Matthieu Herrb/ Email=matthieu.herrb@laas.fr
SSL_CLIENT_S_DN_C	FR
SSL_CLIENT_S_DN_CN	Matthieu Herrb
SSL_CLIENT_S_DN_Email	matthieu.herrb@laas.fr
SSL_CLIENT_S_DN_O	CNRS
SSL_CLIENT_S_DN_OU	UPR8001
SSL_CLIENT_VERIFY	SUCCESS

(liste non exhaustive)

Chapitre 6

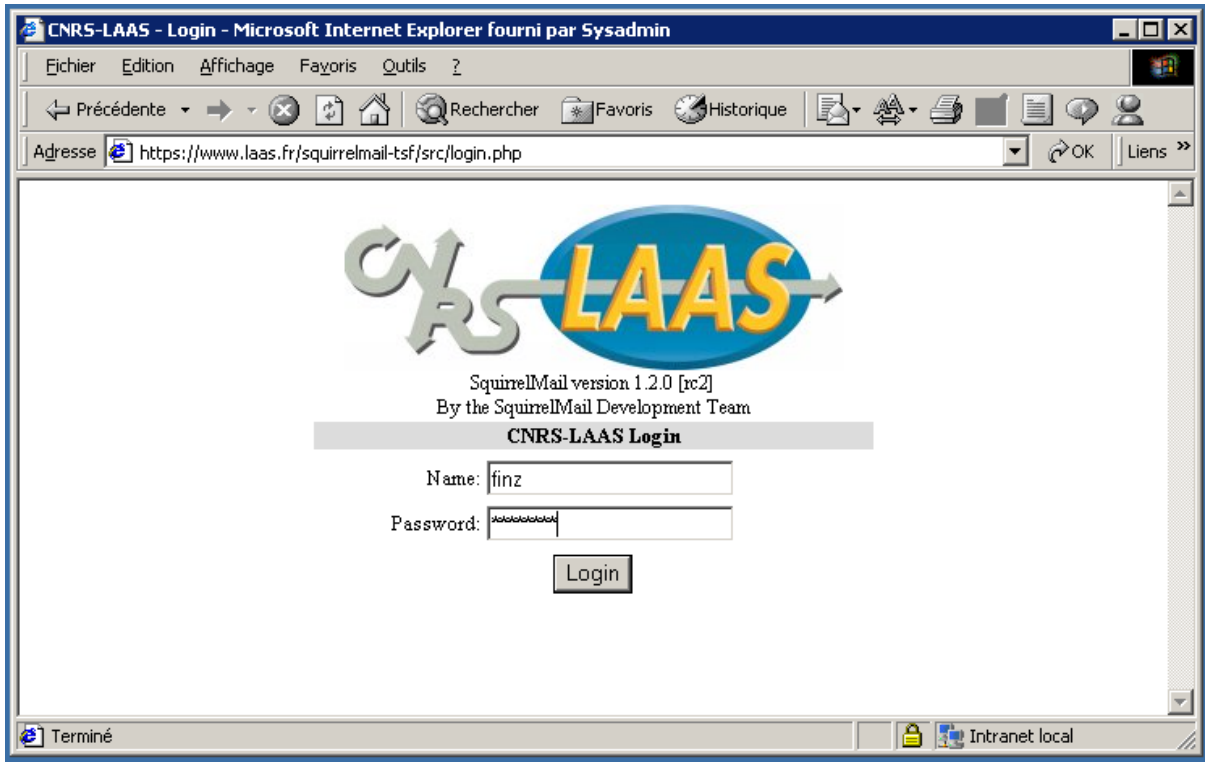
webmail

Squirrelmail

http://www.squirrelmail.org/

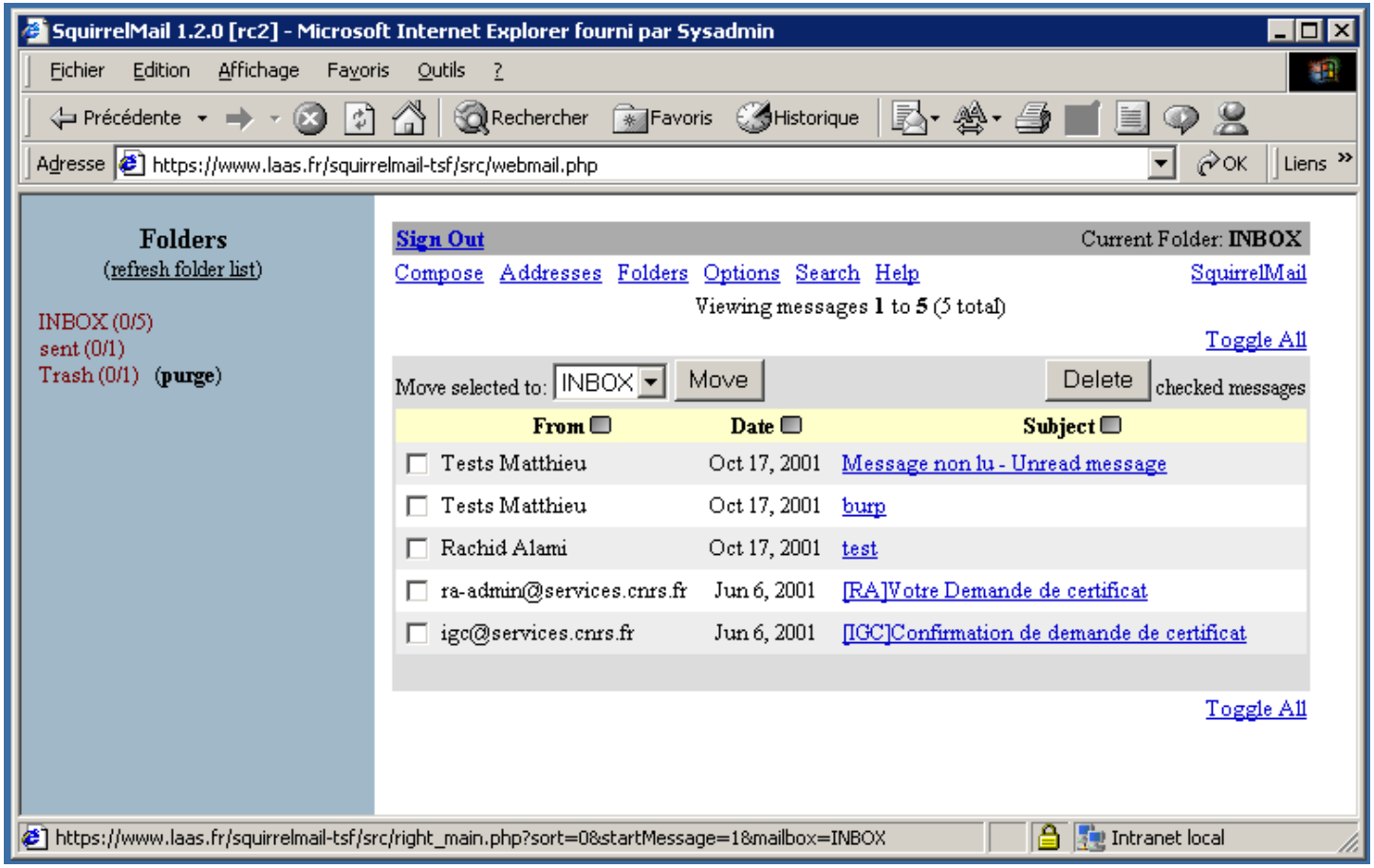
Nécessite :
- Un serveur web : apache + mod_ssl
- PHP 4.2.3
- Un serveur IMAP (IMAPS n'est pas nécessaire)
- clients acceptant les cookies

Messagerie : webmail (1/4)



Écran de login

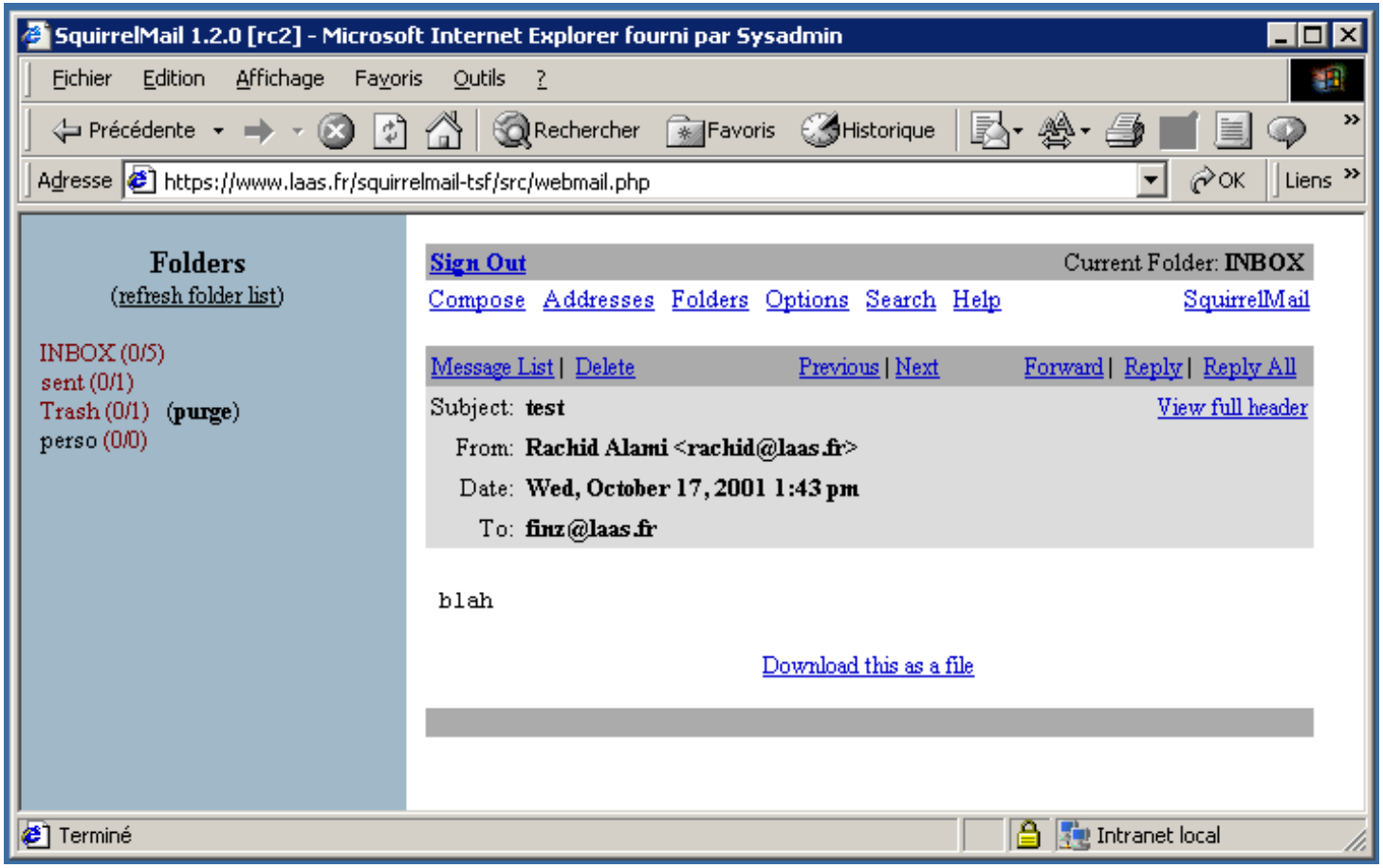
Messagerie : webmail (2/4)



Liste des messages

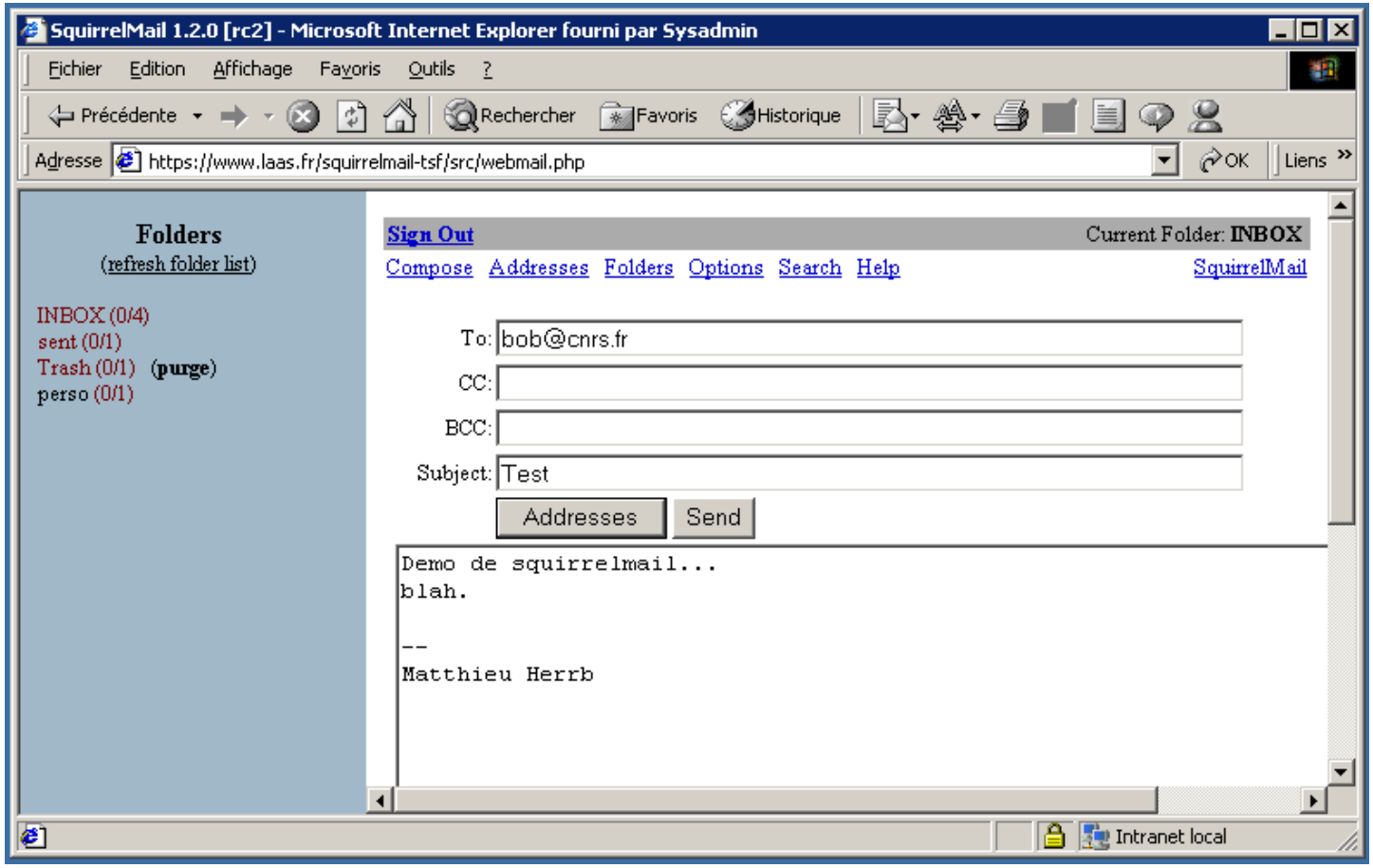
Cliquer sur **Sign Out** pour quitter

Messagerie : webmail (3/4)



Visualisation d'un message

Messagerie : webmail (4/4)



Composition d'un message

Squirrelmail : installation (1/2)

Dans le répertoire `htdocs-ssl` :

```
tar xzvf squirrelmail-1.2.8.tar.gz
mv squirrelmail-1.2.8 squirrelmail
```

Créer les deux répertoires de travail (hors de htdocs !) :

```
mkdir -p /var/squirrelmail/data
chown httpd:httpd /var/squirrelmail/data
chmod 700 /var/squirrelmail/data
mkdir -p /var/squirrelmail/attachements
chown root:httpd /var/squirrelmail/attachements
chmod 730 /var/squirrelmail/attachements
```

Squirrelmail : installation (2/2)

Exécuter le script de configuration : `cd config`
`./conf.pl`

- utiliser l'option 'D' pour définir les parametres par défaut en fonction du serveur imap. (uw)
- définir les parametres du serveur imap (2)
- définir un logo, le nom de votre labo (1)
- . . .

→ produit **config.php** que l'on peut éditer à la main en cas de besoin spécifique.

Chapitre 7

Sécurisation des transferts de fichiers

Solutions

Basées sur SSH :
- scp/sftp (ligne de commande Unix / cygwin)
- iXplorer : interface windows au dessus de SSH http://www.i-tree.org/ixplorer.htm

Basées sur FTP+SSL : Igloo FTP
Voir fiche accès distants de l'UREC

WebFTP & co. http://www.math.jussieu.fr/~jas/ads/install-webftp.html

Chapitre 8

Applets SSH

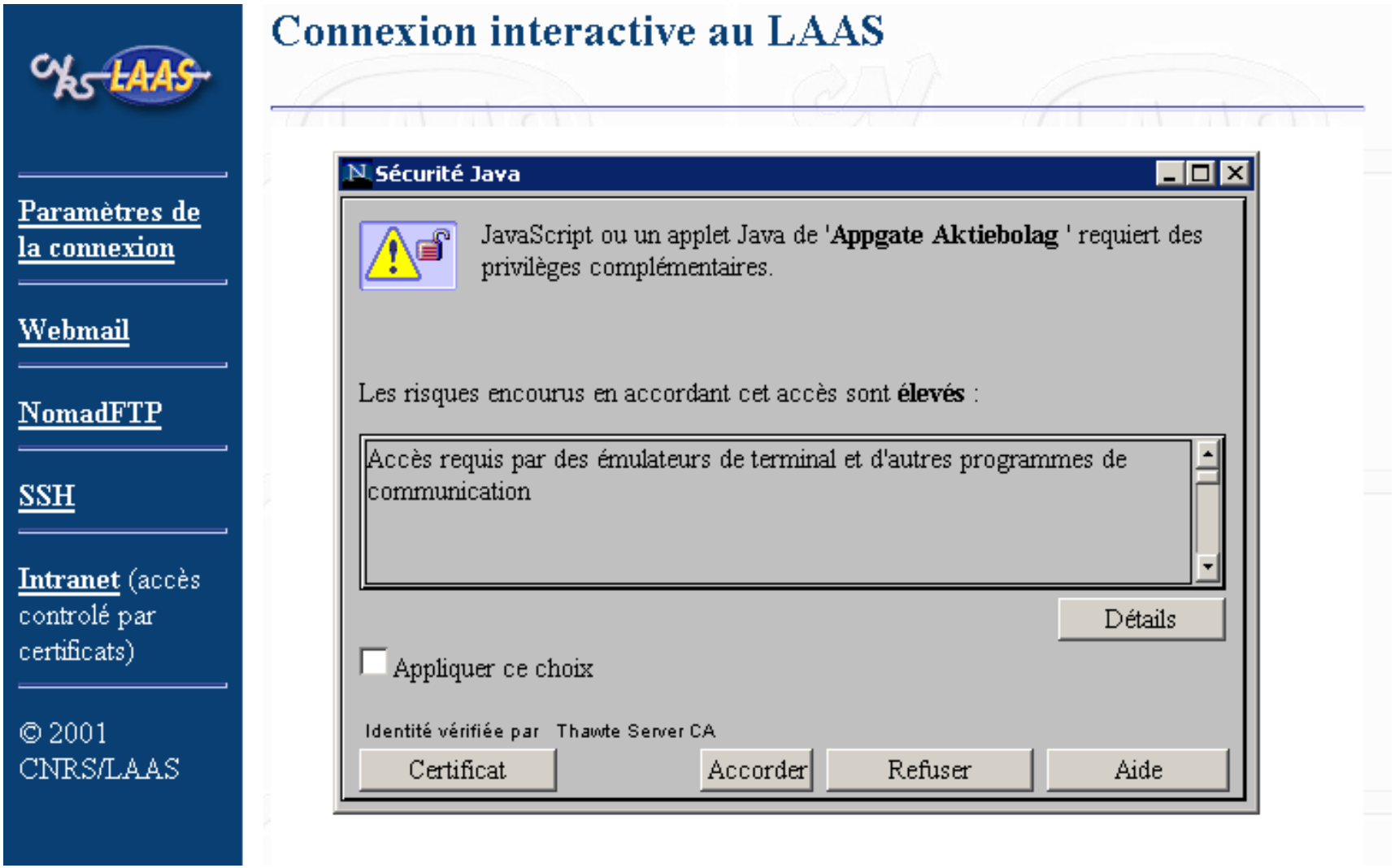
Connexion interactive : applet SSH (1/9)

Applet Java signée permettant d'ouvrir un terminal SSH.

Donner à l'applet les droits qu'elle réclame pour pouvoir se connecter sur n'importe quelle machine.

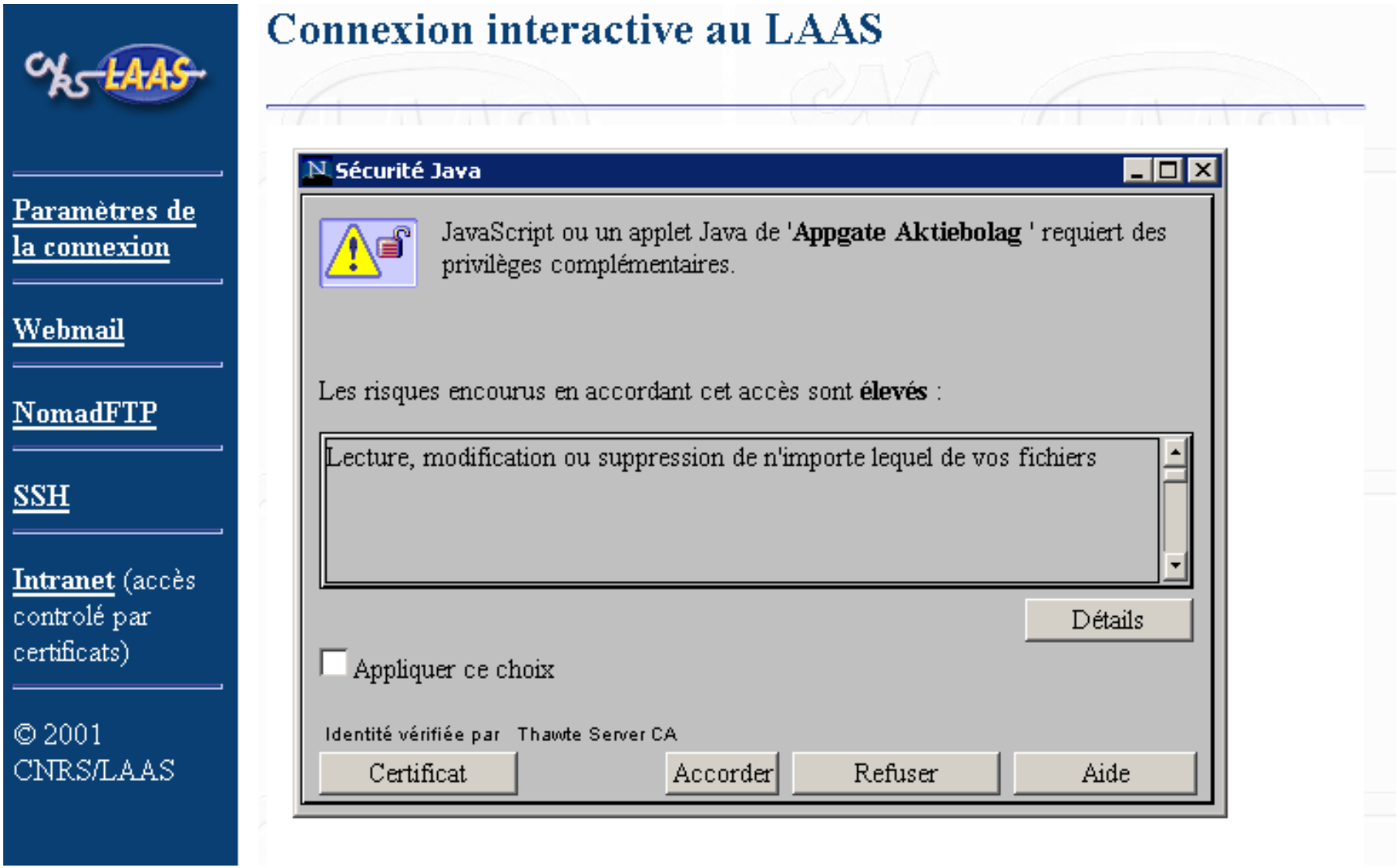
Nécessite Java dans le navigateur.

Connexion interactive : applet SSH (2/9)



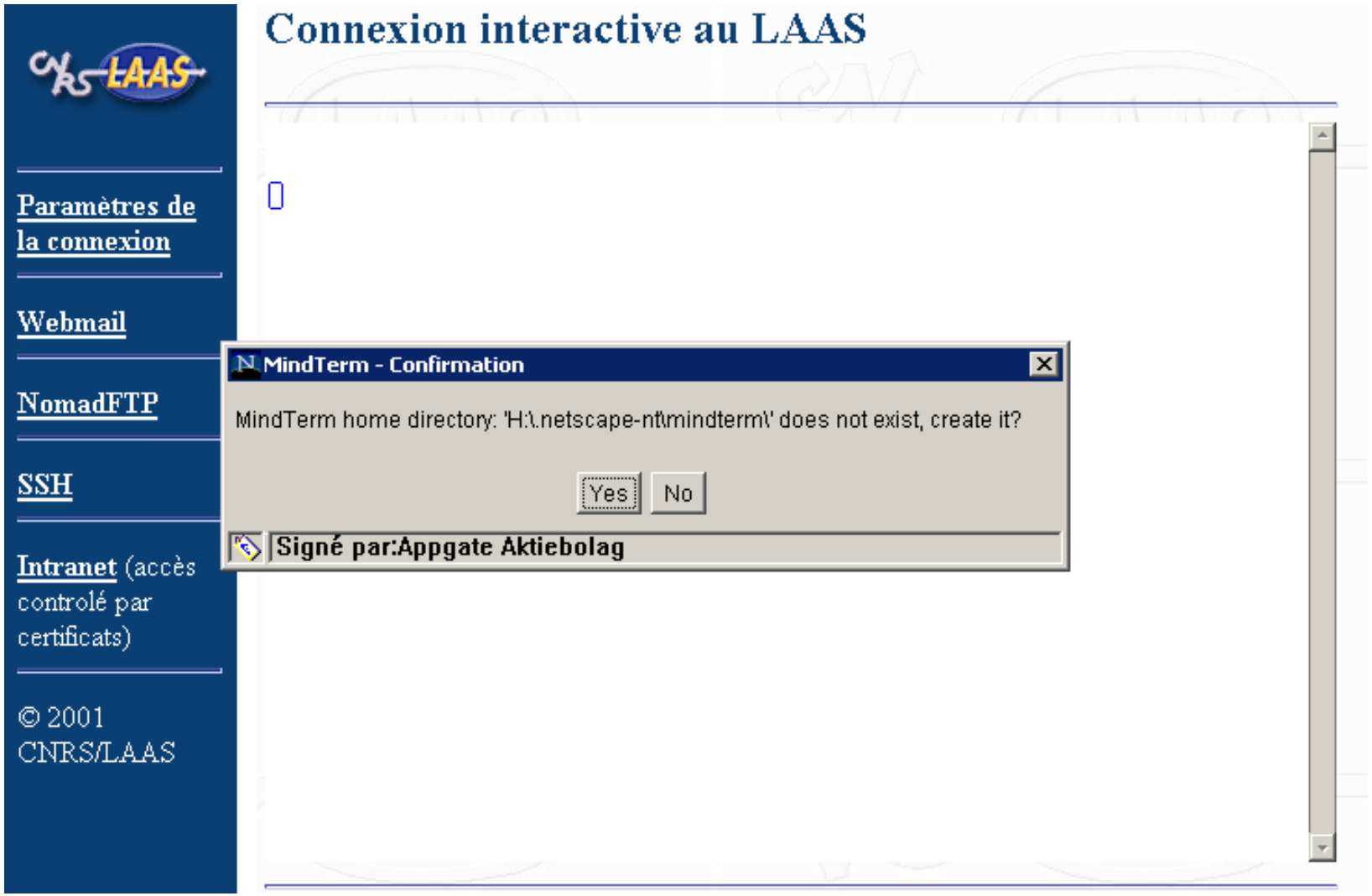
cliquer « **Accorder** »

Connexion interactive : applet SSH (3/9)



cliquer « **Accorder** »

Connexion interactive : applet SSH (4/9)

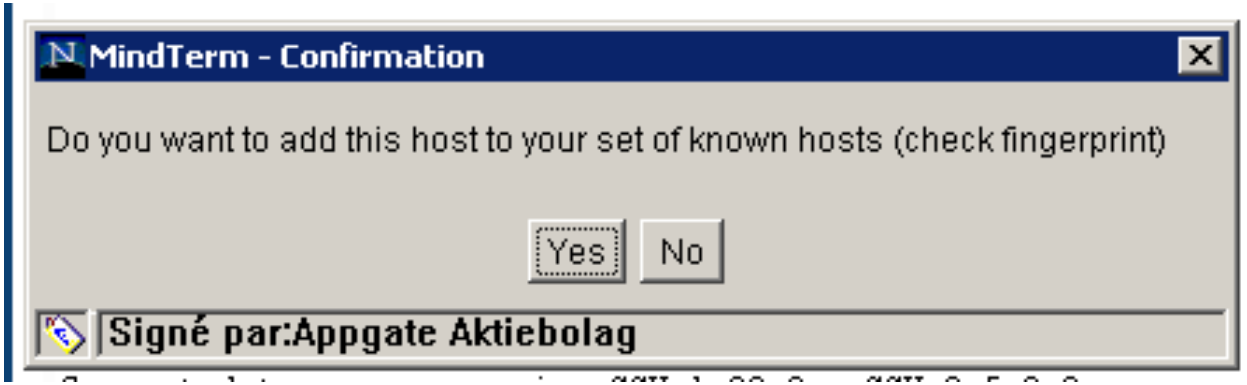


cliquer « **Yes** »

Connexion interactive : applet SSH (5/9)

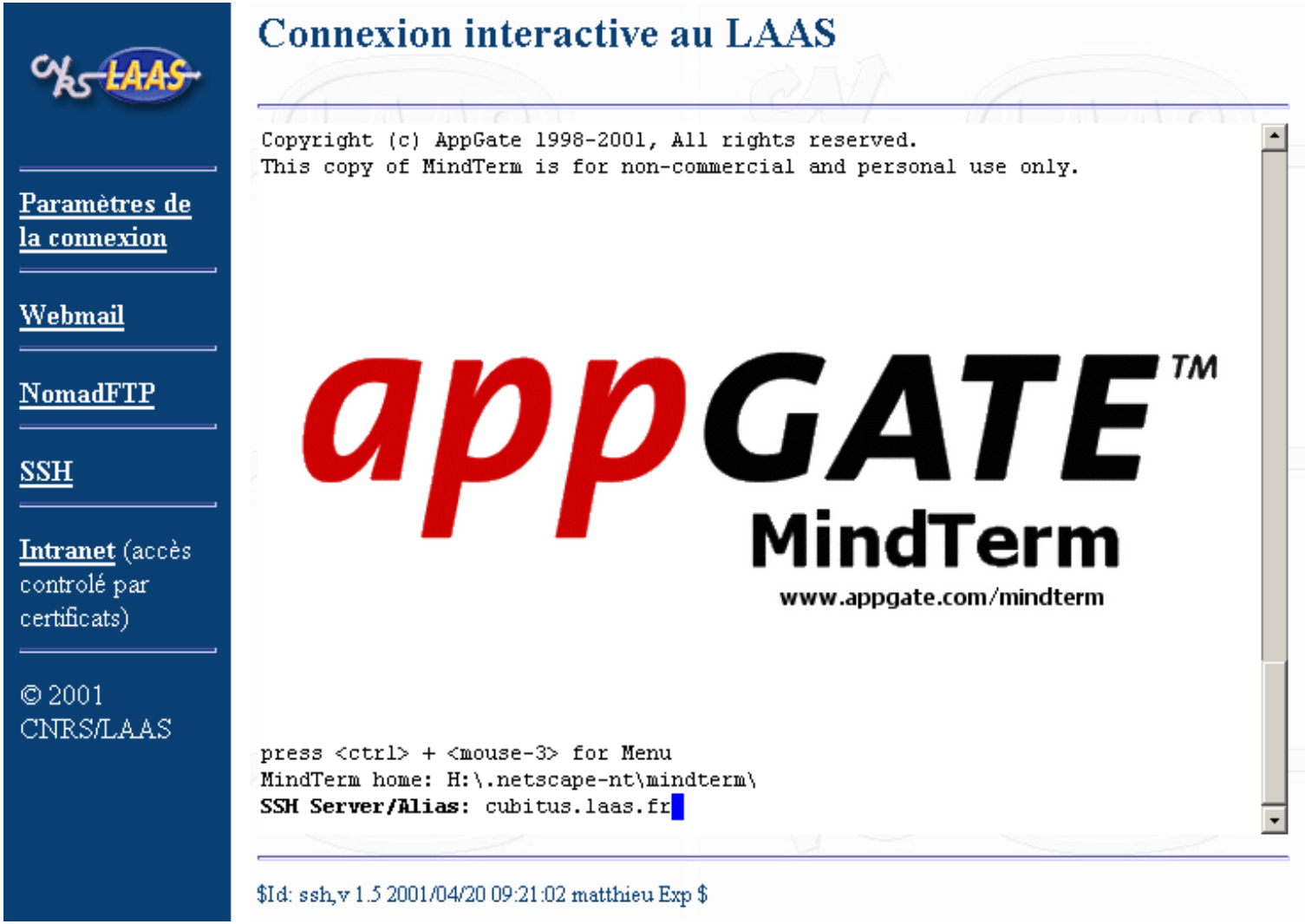
cliquer « **Yes** »

Connexion interactive : applet SSH (6/9)



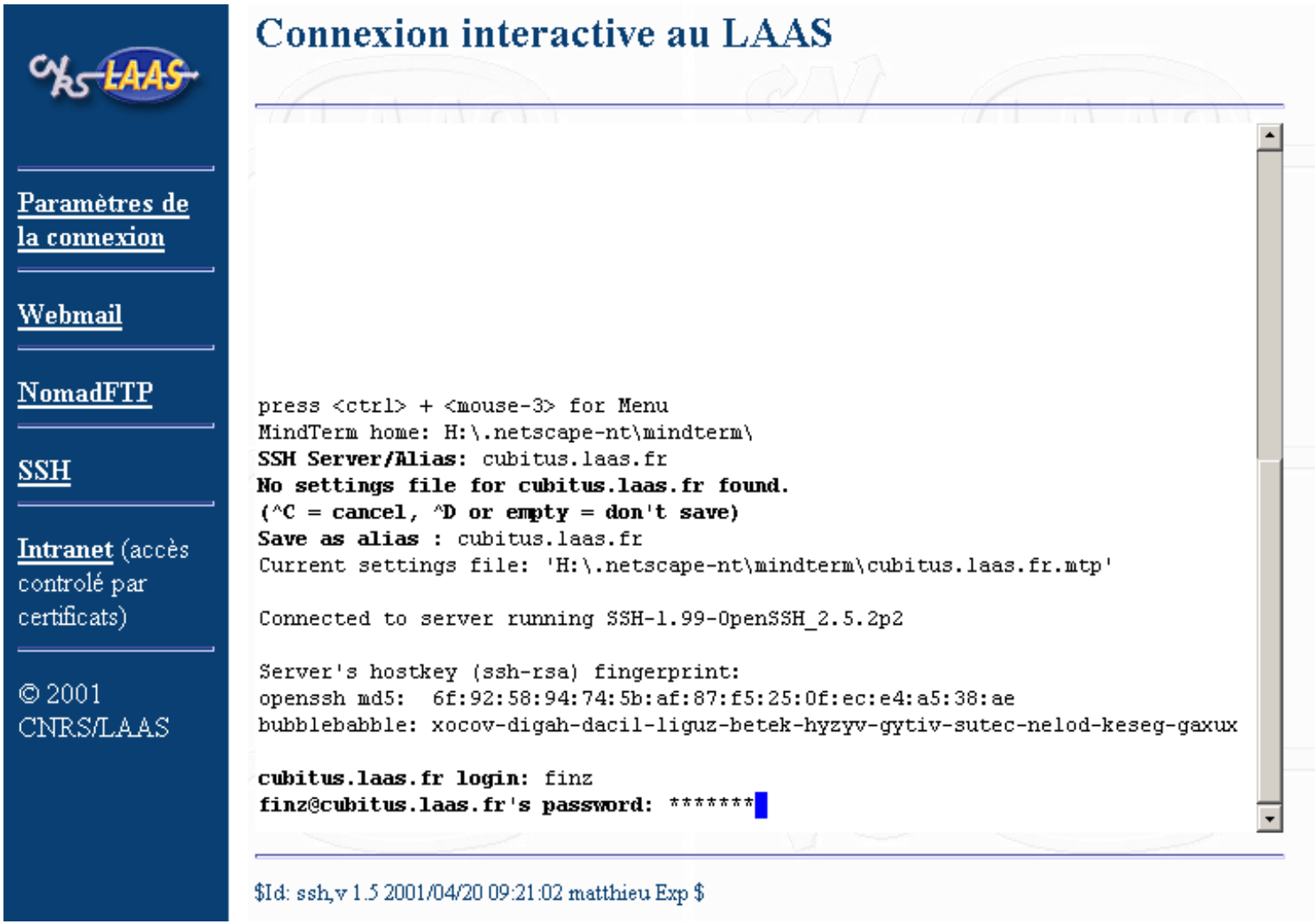
cliquer « **Yes** »

Connexion interactive : applet SSH (7/9)



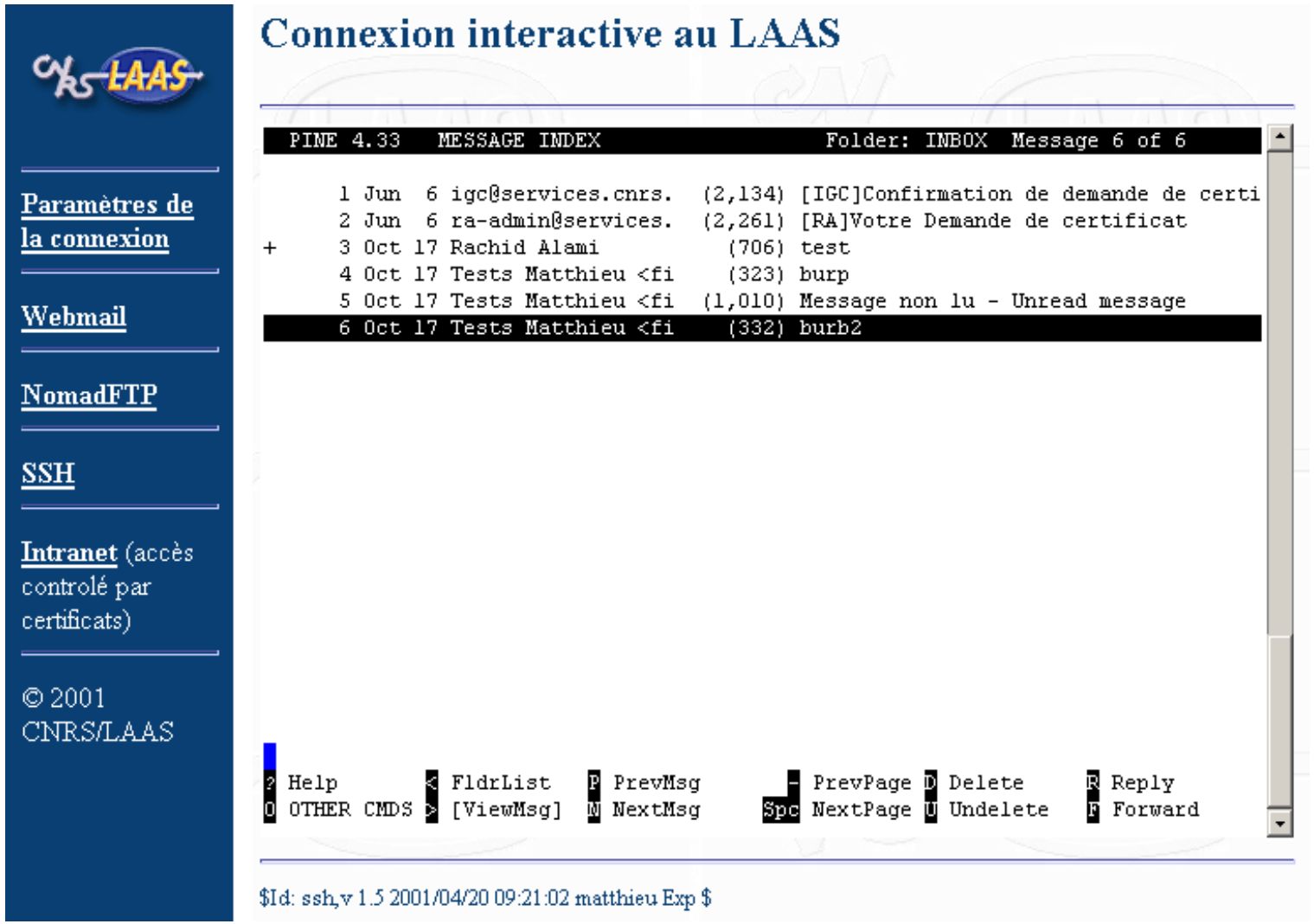
Entrez le nom de la machine « **serveur.mondomaine.fr** »

Connexion interactive : applet SSH (8/9)



Entrez votre nom de login et votre mot de passe

Connexion interactive : applet SSH (9/9)



Exemple d'application : pine

Mindterm : installation

http://www.appgate.com/products/mindterm/index.html

– gratuite pour un usage non-commercial.
– Version courante : 2.1

Installer `mindterm_ie.cab` et `mindterm_ns.jar` dans `htdocs`

Utiliser un script pour fournir la bonne version en fonction du type de client (variable `USER_AGENT`).

Mindterm : exemple de page pour Netscape

```
<html>
<applet archive="/mindterm_ns.jar"
        code="com.mindbright.application.MindTerm"
        width="580" height="400">
<param name=server value="server.monlabo.fr">
<param name=port value="22">
<param name=cipher value="blowfish">
<param name=gm value="80x25">
<param name=forcpty value="true">
<param name=sepframe value="false">
<param name=autoprops value="both">
<param name=quiet value="true">
<param name=appletbg value="white">
</html>
```

MindTerm : exemple de page pour Internet Explorer

```
<html>
<applet code="com.mindbright.application.MindTerm"
        width="580" height="400">
<param name=cabbase value="/mindterm_ie.cab">
<param name=server value="server.monlabo.fr">
<param name=port value="22">
<param name=cipher value="blowfish">
<param name=gm value="80x25">
<param name=forcpty value="true">
<param name=sepframe value="false">
<param name=autoprops value="both">
<param name=quiet value="true">
<param name=appletbg value="white">
</html>
```

Chapitre 9

Conclusion

Par où commencer ?

Priorité 1 :
supprimer les mots de passe qui circulent en clair hors du réseau local !

– Les services sur le Web sont les plus faciles à utiliser pour les utilisateurs.
– Déployer SSH

Évolutions futures

- Cartes à puces, tokens USB : transporter son certificat
- IPSec / IPv6 / mobile IP
- Passport/Security Alliance (?)
- Externaliser certains services ?

Références

Fiches du groupe de travail ADS :

https://www.services.cnrs.fr/corres-secu/

Cours SSL de R. Dirlewanger :

https://www.dr15.cnrs.fr/Cours/JRES99/rd-ssl.pdf