

Auteur : L. Dos Santos	Titre : XFree86 / Cygwin, SSH	Version : 2.0
Date : 11/04/2005	Licence : Gnu Free Documentation Licence	IUP-ASR 2005

Thème

Serveur X sous Windows (XFree86/Cygwin), SSH et travail à distance sur une machine Unix depuis une machine Windows.

Présentation

La gestion de l'interface graphique sous Linux diffère totalement de celle utilisée sous Windows. Effectivement, sous Linux un véritable mécanisme Serveur/Client intervient au cours du lancement d'une application graphique. Afin de pouvoir utiliser les outils graphiques Linux sous Windows, nous devons mettre en place un serveur `ygwin/X` Windows. Pour sécurisé les flux entre les clients graphiques Linux et le serveur `Cygwin/X` Windows nous utiliserons une connexion SSH.

Serveur `Cygwin/X` : «X Windows»

- `Cygwin / Package X11`

Cygwin est une application regroupant un ensemble de logiciels libres permettant de compiler et d'exécuter des applications en provenance du monde Unix, sous Windows. `Cygwin` possède de nombreux packages et notamment le package **X11** permettant l'installation du serveur X pour Windows. Une fois `cygwin` et X installés, le lancement de l'application `cywin` présente un shell `bash` dans lequel nous allons pouvoir exécuter différents logiciels et notamment lancer un serveur «X Windows».

- L'intérêt des connexions «X Windows»

L'intérêt des connexions «X Windows» est de pouvoir accéder à distance à des machines de type Unix et aux divers logiciels qui peuvent être utiles tels Mozilla, éditeurs, et tous logiciels nécessitant un serveur X.

- Les connexions de «X Windows»

Le protocole «**X windows**» permettra d'établir des communications de type graphique entre 2 ordinateurs. On identifie le client comme étant l'ordinateur sur lequel une application Linux quelconque s'exécute. L'affichage de l'application ne se fera donc pas sur le poste client mais sur le poste qui exécute le serveur «X Windows». Une fois connecté, le client demande au serveur un droit d'affichage. Le serveur va alors distribuer un droit d'affichage qui permet au client de rediriger l'affichage des fenêtres sur le poste où `cygwin` et le serveur «X Windows» sont installés.

- La sécurité «X Windows»

Concernant le droit d'affichage et donc la sécurité du serveur X, deux outils distincts sont à notre disposition. On trouvera notamment l'outil «**xhost**», ou l'autorisation se fait sur la base de l'adresse ip de la machine cliente, ce qui induit que tous les utilisateurs logger sur cette machine, auront la possibilité de contrôler le serveur X. La commande «**xauth**», ou l'autorisation se fait sur la base d'un cookie spécifique au serveur et qui devra être connu au préalable des utilisateurs clients, autorisés à se connecter au serveur X. Lors du contrôle d'accès, le serveur X compare d'abord sa table «**xhost**» à l'adresse ip de la machine cliente, puis, dans le cas où l'identification est négative, le serveur X compare alors le cookie de l'utilisateur client au sien.

- Configuration client X

Bien entendu le client X a besoin dans tous les cas de connaître son «**display**», c'est-à-dire le serveur X sur lequel doit se faire l'affichage. La variable d'environnement «**\$DISPLAY**» est alors utilisée et contient cette information. Il est tout à fait possible de redéfinir la valeur de cette variable afin de spécifier aux clients graphiques l'adresse d'un serveur X bien précis. Bien entendu si le serveur X à été lancé sur la base d'une authentification par le biais d'un cookie, ayant une clé secrète précise, l'utilisateur client X devra être configuré, à l'aide de l'outil «**xauth**» afin de connaître ce cookie ; ou bien tout simplement autorisé l'accès d'une ou plusieurs adresse ip à l'aide de «**xhost**».

SSH et redirection X11

- Principe

Une méthode plus pratique et surtout chiffré consiste en l'utilisation de «**ssh**» pour se connecter à un système distant. La connexion est cryptée, et l'établissement de celle-ci peut également servir pour différents tunnels, comme par exemple le protocole X.

- Serveur SSH

Le serveur SSH tournera sur la machine où s'exécuteront les clients X. Le serveur SSH permettra à l'utilisateur distant d'accéder à tous les services auxquels il est habituellement autorisé à accéder en local. Le serveur SSH nécessite au préalable une configuration particulière du fichier «**/etc/ssh/sshd_config**» pour pouvoir effectuer du «**X11 Forwarding**», c'est-à-dire autorisé le protocole X11 à transmettre son flux via un tunnel SSH à destination du serveur «X Windows» et enfin vers le «**socket X**». Pour le daemon «**sshd**» un «**serveur X virtuel**» est créé localement, à travers lequel il assurera la «**redirection X11**» jusqu'au serveur «X Windows».

- Client SSH

Le client SSH sera exécuté sur la machine souhaitant utiliser des outils graphiques sur une machine distante. Deux paramètres au choix sont possibles et nécessaire à la commande «**ssh**» pour accepter la redirection X11 en provenance du serveur «**sshd**».

Le paramètre «**-X**» définit une connexion SSH en mode «**untrusted**», c'est-à-dire que le programme «**ssh**» se servira de la commande «**xauth**» en local afin de connaître la valeur du cookie pour pouvoir se connecter au serveur X local via la «**socket X**».

Le paramètre «**-Y**» quant à lui est dit «**trusted**» et permet à l'aide de «**xauth**», la génération à la volée d'un cookie «**untrusted**» associé au serveur X local, tout en étant différent du cookie statique «**trusted**», utilisé par les applications locales. Un autre cookie «**untrusted**» est généré et envoyé au serveur SSH. Le serveur SSH à jour, la redirection X11 se fait via le tunnel et le client `ssh` réalise l'association entre le cookie «**untrusted**» présenté et le cookie «**untrusted**» réellement utilisé niveau du serveur X.

Ce dernier à l'avantage de ne plus permettre la capture ou l'envoi d'événement au niveau du serveur X.

Réalisation

- Méthode conseillée

- Installation de `cygwin` et du package nommé X11.
- Lancement de `cygwin` et accès au shell `bash`.

- o Lancement du serveur « X Windows » : Commande XWin.exe

```
user@Serveur_X$>XWin.exe -screen 0 1024 768 -fullscreen -depth 32  
-refresh 70 -emulate3buttons 100 -nowinkill -unixkill &
```

- -screen *numéro_serveur_X largeur_pixel hauteur_pixel* : espace X
- -fullscreen : pleine écran
- -depth : résolution
- -refresh : rafraîchissement
- -emulate3buttons : émulation du bouton de souris du milieu
- -nowinkill : pas de fermeture Windows
- -unixkill : fermeture unix

- o Lancement d'un « xterm » sur le serveur « X Windows » :

```
user@Serveur_X$>xterm -display adresse_ip_serveur_X:0.0 &
```

- :0.0 = *numéro_serveur:numéro_écran*

- o Vérification de la variable \$display sur le serveur X:

```
user@Serveur_X$>echo $DISPLAY  
adresse_ip :0.0
```

- Si valeur à localhost :0.0, réaliser un « export DISPLAY=*adresse_ip*:0.0 »

- o Lancer un « Window manager » sur le serveur « X Window » :

```
Serveur_X$>twm &
```

- o Vérification des droits d'accès à partir du xterm sur X:

```
user@Serveur_X$>xhost  
Access control enabled, only authorized clients can connect  
INET: localhost  
INET: nom_netbios_serveur_X  
LOCAL :
```

```
user@Serveur_X$>xauth list  
Xauth : creating new authority file /home/$USER/.Xauthority
```

- Seule la machine hébergeant le serveur X est autorisée à se connecter et qu'aucune règle n'est définie dans « xauth »

- o Configuration du serveur SSH localement ou à l'aide d'une première connexion ssh sur la machine distante :

```
root@Machine_Distante$>vi /etc/ssh/sshd_config &
```

- Vérifier que le paramètre « X11Forwarding » est à « yes ».

- o Lancement du client SSH à partir du xterm sur X :

```
user@Serveur_X$>ssh -Y user@adresse_ip_machine_distante
```

- Renseigner login et pass pour se logger.

- o Vérification de la variable \$display sur la machine distante :

```
user@Machine_Distante$>echo $DISPLAY  
localhost :10.0
```

- Vérifier qu'il s'agit bien du serveur virtuel localhost :10.0

- o Vérification du fichier .Xauthority à l'aide de « xauth »:

```
user@Machine_Distante$>xauth list  
Xauth : adresse_ip:10.0 MIT-MAGIC-COOKIE-1 "clé_untrusted"
```

- Vérifier qu'un cookie vers le serveur virtuel 10 existe.

- o Lancement d'un client X sur le serveur X à l'aide de ssh:

```
user@Machine_Distante$>Mozilla &
```

- Constaté l'ouverture en mode graphique sur X Windows.

- o Visualisation du PID d'exécution de mozilla

```
user@Machine_Distante$>ps aux | grep mozilla
```

- Relever le PID de mozilla.

- o Fermeture de l'application

```
user@Machine_Distante$>kill -9 2534
```

- Constaté la fermeture impossible avec la croix Windows
- Constaté la fermeture de mozilla à la suite du kill.

- o Fermeture de la connexion ssh

```
user@Machine_Distante$>exit
```

- o Fermeture du serveur X

```
user@Machine_Distante$>Kill 1584
```

- Possibilité d'utiliser le kill suivi du pid
- Possibilité d'utiliser CTRL+ALT+BKSP
- En tuant le processus X vous tuez par la même le processus xterm

- Informations complémentaires

- o Paramètres supplémentaires pour XWin.exe
 - -auth fichier : fournir un fichier d'authentification utilisateur. Mots clé utile pour auth : add, remove
 - -multiwindow : mode multi fenêtré
 - --h : visualiser tous les paramètres du serveur X
- o Port d'exécution du serveur virtuel en SSH : 6000 + n avec n=10
- o Paramètres pour xhost : + ou – suivi de l'adresse ip

Références

- Manuels Cygwin
 - <http://x.cygwin.com/docs/ug/cygwin-x-ug.pdf>
 - <http://www.soe.ucsc.edu/~you/notes/cygwin-install.html>
- Système de multifenêtrage X11
 - http://www-internal.alphanet.ch/~schaefer/unixguide/unixguide_html/node203.html
- Notion relative à SSH et X11 Forwarding
 - <http://www.hsc.fr/ressources/breves/ssh-x11.html.fr>
 - <http://anoncvs.openbsd.lt/cgi-bin/viewcvs.cgi/www/openssh/fr/faq.html?rev=1.15#2.7>