Sécurité: protection des serveurs X

Table of Contents

Xauth

xauth est un mécanisme permettant d'appliquer des contrôles d'accès sur des serveurs X (écrans).

Lorsqu'un serveur X est démarré, un “cookie” généré de manière aléatoire lui est attribué. Ce cookie est écrit dans un fichier détenu et lisible par l'utilisateur dont la session est en cours d'exécution sur le serveur X. Aucun autre utilisateur ne peut lire ce fichier.

Lorsqu'un client X (application) est démarré, il tente de lire et d'utiliser le cookie pour s'authentifier auprès du serveur. Si cette authentification **xauth** échoue, l'application n'est pas autorisée à se connecter au serveur et à afficher des fenêtres sur l'affichage X.

Le fichier `.Xauthority` est généré par le programme **xauth**. Il s'agit d'un système d'authentification pour les applications graphiques. Cela permet d'éviter que d'autres personnes envoient des images, des fenêtres sur votre écran - mais également que des personnes puissent “voir” ce qu'il y a sur votre écran.

Consultation des cookies

La commande `xauth list` permet d'affiche les cookies disponibles pour la session en cours:

```
$ xauth list
xfce/unix:0  MIT-MAGIC-COOKIE-1  e38904001366110fd8ef72c4d483ca4a
xx.xx.xxx.xx:0  MIT-MAGIC-COOKIE-1  c3b321e93d509eb3bac60c9a61370b2b
```

Traditionnellement, l'emplacement des cookies de chaque utilisateur a été ~/.Xauthority: le serveur X écrit ses cookies dans ce fichier au démarrage, et xauth (ainsi que d'autres clients X) recherchent dans ce fichier des cookies d'authentification.

Génération du fichier .Xauthority

Le principe est de donner une clef d'identification, en hexadécimal avec un nombre pair de caractères.

Lancer `xauth` et faire :

```
add MaMachine:0 MIT-MAGIC-COOKIE-1 MonCode
add MaMachine/unix:0 MIT-MAGIC-COOKIE-1 MonCode
```

Pour la machine locale, c'est en fait “hostname:NoDisplay”.

Un fois lancé, le serveur X interdit toute connexion, sauf si l'application :

- est exécutée sur une machine référencée ;
- possède le code.

On peut désactiver le système pour certaines machines avec un xhost +toto par exemple.

Certaines versions obligent à lancer le serveur X avec la commande suivante:

```
xinit -- -auth $HOME/.Xauthority
```

Sécurisation des connexions

Utilisation de $XAUTHORITY

Afin de se connecter à un serveur X et ouvrir une application client X sur l'affichage distant, il faut disposer des cookies xauth appropriés. Afin de sécuriser le serveur X il ne faut pas utiliser `~/.Xauthority` comme emplacement pour les cookies. Au lieu de cela, on les place dans `/tmp/.<Séquence-aléatoire-de-lettres>`, puis on définit la variable d'environnement **$XAUTHORITY** pour indiquer à **xauth** où le rechercher. Cette variable d'environnement n'est visible que pour les applications lancées à partir de l'environnement X local. Par conséquent, seul l'utilisateur local qui “possède” actuellement l'affichage X peut savoir avec certitude l'emplacement du fichier approprié. Il ne peut le faire qu'à partir de l'affichage X.

- Renommer le fichier `.Xauthority` existant en exécutant la commande suivante

```
mv .Xauthority old.Xauthority
```

- Créer le repertoire /tmp/<Séquence-aléatoire-de-lettres>

```
mkdir /tmp/.xth-$(xxd -l 16 -p /dev/urandom)
```

- Récupérer le chemin du répertoire ainsi créé

```
ls /tmp -al

drwxrwxr-x 1 jacques.nougat jacques.nougat    0 juil.  3 12:51
.xth-3bcd598058f6b8455458648a7a628b97
```

- Créer le fichier .Xauthority existe

```
touch /tmp/.xth-3bcd598058f6b8455458648a7a628b97/.Xauthority
```

- Définir la variable d'environnement $XAUTHORITY

```
export XAUTHORITY=/tmp/.xth-3bcd598058f6b8455458648a7a628b97/.Xauthority
```

- Générer le clef pour le domaine socket unix (seule cette clé est nécessaire pour X11 sur SSH)

```
xauth generate :0 . trusted
```

- générer une nouvelle clé, xauth nécessite un encodage hex 128 bits

```
xauth add x.x.x.x:0 . $(xxd -l 16 -p /dev/urandom)
```

Afin d'autoriser sous ssh le serveur X depuis l'extérieur et exécuter une application X sur l'affichage, il faut définir $XAUTHORITY pour qu'il pointe vers ce fichier permetant aux applications client X de se connecter au serveur X et de l'utiliser.

Contrôle de la session

Seul xdm assure un contrôle de session X Window correct. La directive DontZap, placée dans la section ServerFlags du fichier de configuration de XFree86 limite aussi les possibilités de gourance.

Si on n'emploie pas xdm : afin d'interdire aux malintentionnés d'utiliser les touches de “basculement” des consoles virtuelles (Alt-F1, Alt-F2 ...) il suffit de placer dans `/etc/profile` une ligne :

```
alias x='(startx >/dev/null &);clear;logout'
```

Puis d'invoquer x en lieu et place de startx.

Protection de la machine contre l'extérieur

Une solution pour éviter les connexions externes est d'utiliser TCP/Wrappers. Il est très fortement conseillé de le recompiler !

L'installation est assez intuitive. En bref, il suffit d'indiquer le nom des machines autorisées dans le fichier `/etc/hosts.allow` et les machines interdites dans `/etc/hosts.deny`. On peut permettre l'envoi de courrier lorsqu'une machine tente de se connecter alors qu'elle est interdite en mettant par exemple dans le fichier `/etc/hosts.deny`:

```
wu.ftpd:        ALL: twist = /usr/sbin/real-daemon-dir/safe_finger -l @%h
|/bin/mail -s %d-%h root
```