



SCANNER EN TANT QU'UTILISATEUR AVEC UN SCANNER SUR PORT PARALLELE

par [Christian Chartier](#)

LinuxGraphic.org

Articles

-
- [Introduction](#)
- [Partie 1](#)
- [Partie 2](#)
- [Conclusion](#)
- [Retour Index](#)

Introduction

Cet article va vous proposer la procédure nécessaire pour pouvoir scanner en tant que simple utilisateur avec un scanner sur port parallèle.

Il comporte quelques rappels sur les services sous linux. Si vous connaissez déjà tout des services, vous pouvez sauter la page correspondante (partie 1) !

La procédure décrite concerne un scanner Umax 2000P, utilisé avec une Mandrake 8.2. Elle peut être appliquée pour tout scanner sur port parallèle (en changeant simplement le backend utilisé) et sur toute distribution récente basée sur la Red Hat (Mandrake, Suse, ...).

Comment scanner en tant qu'utilisateur sous Linux avec un scanner sur port parallèle

Sous Linux, l'accès aux scanners se fait via `sane` (Scanner Access Now Easy), qui est l'équivalent (mais en mieux :-)) de `twain` sous Windows, notamment parce qu'il est conçu pour fonctionner en réseau.

Sane se compose de deux parties : le frontend, qui est l'interface avec utilisateur, et le backend, qui est l'interface avec le scanner. Ces deux parties communiquent entre elles de manière transparente pour l'utilisateur.

Le backend implémente les fonctions de communication avec le firmware du scanner (c'est-à-dire le logiciel interne du scanner, qui permet de recevoir des commandes de l'utilisateur). Le frontend offre un accès unifié au backend.

Il existe des frontend's graphiques, comme `xsane` ou `xscanimage`, permettant l'acquisition d'image via une interface graphique. Le frontend le plus utilisé est `xsane`, notamment parce qu'il peut être lancé comme plugin de `The Gimp`, logiciel de référence de retouche d'image sous Linux. Mais il en existe d'autres, vous en trouverez la description sur ce site.

Le backend est propre à un modèle (ou à une famille) de scanner(s).

Il existe un backend particulier, appelé `net`, qui permet l'acquisition via le réseau. Nous en parlerons plus loin.

Rappels sur l'accès au matériel sous linux

Sous linux, l'accès au matériel nécessite les droits d'administrateur (droit de root). En effet, si l'accès au matériel était possible en tant qu'utilisateur, tous les mécanismes de protection du système pourraient être outrepassés (par exemple, écriture directe sur le

disque dur). De ce fait, le système ne serait plus sûr !

Ces restrictions sur l'accès au matériel ont pour conséquence de nécessiter les droits de root pour accéder au port parallèle. Cette situation est inacceptable dans un système de type linux où justement la fiabilité du système est liée (entre autres) à une séparation des droits des utilisateurs et de l'administrateur de la machine.

Mais il y a des solutions : la première qui vient à l'esprit est de donner les droits root au programme `sane` lui-même, via le bit `suid`.

Cette solution est à exclure car tout programme exécuté avec le bit `suid` s'exécute avec les droits de root est constituée donc une faille potentielle de sécurité.

La solution à préférer est celle consistant à scanner via l'interface réseau (avec le backend `net`).



SCANNER EN TANT QU'UTILISATEUR AVEC UN SCANNER SUR PORT PARALLELE

par Christian Chartier

LinuxGraphic.org

Articles

-
- [Introduction](#)
- [Partie 1](#)
- [Partie 2](#)
- [Conclusion](#)
- [Retour Index](#)

Rappels sur les services réseau sous linux

Linux est un système d'exploitation multi-tâches multi-utilisateurs orienté réseau. Il possède en standard tout ce qu'il faut pour offrir des services réseau.

Ces services réseau sont offerts par des programmes tournant en arrière plan (appelés démons, de l'anglais daemon, acronyme de Disk And Execution MONitor), qui écoutent le réseau pour savoir si une requête leur est adressée. Chaque programme écoute une adresse particulière qui s'appelle un port. A chaque port correspond un service réseau particulier. La liste des correspondances entre le numéro de port et le service figure dans le fichier `/etc/services`.

Sur un réseau, un service tournant sur une machine particulière est identifié par l'adresse IP de la machine (ou son nom) et le port sur lequel il écoute, sous la forme :

```
adresseIP:port ou nom_de_machine:port
```

Ces services (démons) peuvent être lancés au démarrage de la machine (par exemple le service d'impression cups ou lp, ou encore le serveur Web Apache) et tournent en permanence en arrière-plan. Ils consomment donc en permanence des ressources système.

Pour éviter de surcharger la machine par des services dont le besoin n'est pas permanent, il existe un « super-démon » appelé **inetd**, ou sur les distributions plus récentes **xinetd**. Ce super-démon est un service tournant en permanence, lancé au démarrage, dont le rôle est d'écouter l'ensemble des ports listés dans ses fichiers de configuration (`/etc/xinetd.conf` et les fichiers contenus dans `/etc/xinetd.d/`) et, en cas de requête sur l'un de ces ports, d'exécuter le programme correspondant.

Comment utiliser le réseau pour scanner ?

Sane offre un service réseau : le démon `saned`, qui permet de scanner via le réseau. Il n'est pas indispensable que ce service tourne en permanence, aussi préférera-t-on son lancement par le super-démon `xinetd`.

Il faut bien entendu configurer `xinetd` pour cela. L'intérêt de cette manoeuvre dans le cas qui nous intéresse est qu'on peut configurer l'utilisateur sous l'identifiant duquel sera lancé le démon `saned`. En l'occurrence, l'accès au port parallèle nécessitant les droits de root, le démon `saned` sera lancé en tant que root.

Ces informations figurent dans un fichier nommé `sane` placé dans `/etc/xinetd.d/`

Oui, mais moi, je n'ai pas de réseau !

Il n'est pas nécessaire d'avoir plusieurs machines pour utiliser des démons réseau. L'exemple du démon d'impression cups le montre bien : on peut imprimer même si l'on n'a qu'une machine.

Dans ce cas, l'accès aux services réseau s'effectue via une interface réseau particulière : l'interface `loopback`, qui est en fait une interface réseau qui fait référence à la machine elle-même. On accède donc à un service réseau tournant sur la machine même par l'adresse de loopback qui est `127.0.0.1`, ou par le nom de l'hôte local (`localhost`, ou le nom que vous avez donné à votre machine). Cependant, si vous utilisez un nom, la correspondance entre ce nom et l'adresse de loopback doit pouvoir être établie. Cette correspondance doit être établie par une entrée dans le fichier `/etc/hosts` sous la forme suivante :

```
127.0.0.1 localhost.localdomain localhost
```

```
127.0.0.1 mamachine.mondomaine mamachine
```

D'autre part, le fichier `/etc/host.conf` doit contenir une ligne du type de celle-ci :

```
order hosts, bind
multi on
```

Notez bien la présence après `order` du mot `hosts` : ce mot dit au système d'aller un premier lieu chercher la correspondance entre nom et adresse dans le fichier `hosts`.

Pourquoi puis-je imprimer en tant qu'utilisateur, alors que j'accède au port parallèle ?

Sous linux, quand on imprime, on n'accède pas au port parallèle ! On ne fait que déposer un fichier d'impression dans un répertoire particulier du système, le répertoire de spool.

Ensuite, le démon d'impression (qui est, comme je l'ai dit plus haut, un service), qui regarde en permanence s'il y a quelque chose à imprimer dans le répertoire de spool, envoie le fichier d'impression sur le port parallèle. Et ce service d'impression s'exécute avec les droits `root`, donc il peut écrire sur le port parallèle.



SCANNER EN TANT QU'UTILISATEUR AVEC UN SCANNER SUR PORT PARALLELE

par Christian Chartier

LinuxGraphic.org

Articles

-
- [Introduction](#)
- [Partie 1](#)
- [Partie 2](#)
- [Conclusion](#)
- [Retour Index](#)

Passons à la pratique

Pour cet exemple de configuration, nous utiliserons le scanner Umax 2000P, mais la configuration décrite a été également testée avec succès avec un scanner Mustek sur port parallèle (avec le backend `mustek_pp`). Si vous avez un autre modèle de scanner supporté par `sane`, remplacez dans ce qui suit le backend `umax_pp` par celui correspondant à votre modèle de scanner.

Nous utiliserons la distribution Mandrake 8.2. Cette distribution étant basée sur une Red Hat, ces informations devraient pouvoir être transposées sur toute distribution basée sur la Red Hat.

Toutes les manipulations d'installation de packages et de modification des fichiers de configuration doivent être effectués en tant que superutilisateur (`root`).

Pour cela, ouvrir une console et taper :

```
su
```

puis le mot de passe de `root`.

Installer `sane` :

Le backend nécessaire au scanner `umax 2000P` livré avec la version 1.07 de `sane` figurant sur les CD de la Mandrake 8.2 comporte un bug (corrigé depuis, mais je ne crois pas qu'il existe un package à jour). Il ne faut donc pas installer cette version de `sane`. Si vous l'avez déjà installée, commencez par la désinstaller.

```
rpm -e sane-backends  
rpm -e libsane1
```

Installer la version 1.05 (présente sur les CD de la mandrake 8.1 ou téléchargeable sur internet : faire une recherche sur www.rpmfind.net).

Vous pouvez par contre installer la version de `xsane` de la mandrake 8.2 (version 0.84).

1 télécharger `sane` (ou le récupérer sur les CDs de la Mandrake 8.1) et `xsane`

Il vous faut les packages `sane-backends`, `libsane1` et `xsane`.

Créez un répertoire de travail, et copiez-y les trois packages.

2 installer sane

Placez vous dans le répertoire contenant les trois packages , puis installez-les en tapant les commandes :

```
rpm -ivh libsane*.rpm
rpm -ivh sane*.rpm
rpm -ivh xsane*.rpm
```

ou plus rapidement :

```
rpm -ivh *.rpm --nodeps
```

2 bis vérifier si les packages nécessaires sont bien installés

```
rpm -qa | grep sane
```

Vous devez voir les trois packages ci-dessus.

3 configurer le backend umax_pp

Configurer le port parallèle sur lequel est connecté votre scanner. Il vous faut pour cela modifier à l'aide d'un éditeur de texte (vi ou kedit ou autre) le fichier `/etc/sane.d/dll.conf` comme suit (si votre scanner est connecté au premier port parallèle) :

```
# DEVICES #

# specify the port your scanner is connected to.
Possible are 0x378 (lp0)
# 0x278 (lp1)
port 0x378
```

Configurer le modèle de votre scanner

```
# model number
#
# valid values are 610, 1220, 1600 and 2000
#
option astra 2000
```

N'oubliez pas d'enregistrer vos modifications !

4 configurer sane

Pour cela, éditez le fichier `/etc/saned.d/dll.conf` et décommentez les lignes `net` et `umax_pp` (toutes les autres lignes peuvent être commentées ou même supprimées)

Pour décommenter : enlever le # en début de ligne pour commenter : mettre un # en début de ligne.

```
# enable the next line if you want to allow access
through the network:
net
# abaton
# agfafocus
# sm3600
# snapscan
.....
# sp15c
# tamarack
# umax
umax_pp
# v4l
```

N'oubliez pas d'enregistrer vos modifications !

5 tester le scanner en lançant un scan en tant que root (après avoir connecté et mis sous tension le scanner !) en tapant :

```
xsane
```

si le scanner est reconnu par sane, vous devriez voir apparaître l'écran principal de xsane. Sinon, vous aurez un message d'erreur : *xsane : aucun périphérique disponible*. Revoyez votre configuration depuis le point 2bis.

Si le scanner est bien trouvé, quittez xsane et passez aux points suivants :

6 – configurez le backend net

Pour cela, éditez le fichier `/etc/sane.d/net.conf` et décommentez la ligne `localhost` :

```
# This is the net config file.  Each line names a
host to attach to.
# If you list "localhost" then your backends can be
accessed either
# directly or through the net backend.  Going
through the net backend
# may be necessary to access devices that need
special privileges.
localhost
```

N'oubliez pas d'enregistrer vos modifications !

7 – configurez saned

Pour cela, éditez le fichier `/etc/sane.d/saned.conf` et décommentez la ligne `localhost` :

```
#
```

```

# saned.conf
#
# The contents of the saned.conf file is a list of host
# names that are permitted by saned to use local SANE
# devices in a networked configuration. The hostname
# matching is NO LONGER case-sensitive.
#
#scan-client.somedomain.firm
localhost
#

```

N'oubliez pas d'enregistrer vos modifications !

8 rajouter le service sane dans /etc/services

Pour cela, ouvrez le fichier /etc/services avec un éditeur de texte et rajoutez la ligne décrivant le service sane (vers la fin du fichier) :

```

.....
x11-ssh-offset    6010/tcp                # SSH X11
forwarding offset
sane            6566/tcp                # Network Scanner
daemon
ircd              6667/tcp                # Internet Relay
Chat
ircd              6667/udp                # Internet Relay
Chat
.....

```

N'oubliez pas d'enregistrer vos modifications !

9 créez un fichier /etc/xinetd.d/sane contenant les lignes suivantes :

```

# default: on
service sane
{
    port      = 6566
    socket_type = stream
    wait      = no
    only_from = localhost
    user      = root
    server    = /usr/sbin/saned
    log_on_failure += USERID
    disable  = no
}

```

N'oubliez pas d'enregistrer le fichier !

Notez bien les lignes suivantes :

only from = localhost précise à partir de quelle machine le service sane peut être lancé. Ici on n'autorise le lancement du service qu'à partir de la machine elle-même (localhost). Si vous vouliez scanner à

partir d'un autre poste de votre réseau local, il vous faudrait mettre ici les noms ou adresses des machines autorisées, ou du réseau autorisé. user = root précise que le programme (désigné à la ligne server =) sera lancé en tant que root
disable = no précise que le service n'est pas désactivé (donc qu'il est activé !)

10 vérifiez que xinetd est bien lancé au démarrage (avec drakxservices ou linuxconf ou tout autre utilitaire de configuration des services), ou plus simplement en tapant :

```
chkconfig list xinetd
```

le service doit être actif dans les runlevels (niveaux d'exécution) 3, 4 et 5.

Si xinetd n'est pas actif, tapez :

```
chkconfig add xinetd
```

11 lancez ou relancez le service xinetd :

Si xinetd n'était pas actif, tapez :

```
service xinetd start
```

S'il était actif, tapez :

```
service xinetd restart
```

12 repassez en mode utilisateur :

```
exit
```

13 lancez xsane

```
xsane
```

si le scanner est reconnu par sane, vous devriez voir apparaître l'écran principal de xsane. Sinon, vous aurez un message d'erreur : `xsane : aucun périphérique disponible`. Revoyez votre configuration depuis le point 5.



SCANNER EN TANT QU'UTILISATEUR AVEC UN SCANNER SUR PORT PARALLELE

par [Christian Chartier](#)

LinuxGraphic.org

Articles

-
- [Introduction](#)
- [Partie 1](#)
- [Partie 2](#)
- [Conclusion](#)
- [Retour Index](#)

Conclusion

Cet article vous a proposé la configuration nécessaire pour scanner en tant que simple utilisateur avec un scanner sur port parallèle.

Il utilise pour cela l'interface réseau de sane.

De ce fait, cette configuration est également valable pour scanner via un réseau local. Il suffira de modifier les fichiers correspondants pour permettre l'accès au serveur de scanner (celui sur lequel est connecté le scanner) à partir des autres postes du réseau.

Notes importantes :

Cet article est volontairement simplifié. Il ne prétend pas expliquer de manière exhaustive l'art et la manière d'utiliser les services sous Linux.

S'il comportait des inexactitudes ou des erreurs, je serais heureux que vous m'en fassiez part !

Si vous rencontrez des difficultés pour mettre en oeuvre cette configuration, n'hésitez pas à me contacter : cricx@free.fr