

Les services

Le contrôle de l'accès aux services du réseau est l'une des tâches de sécurité les plus importantes à laquelle un administrateur de serveur doit faire face. Il existe un certain nombre d'outils conçus pour des tâches d'administration tel qu'un pare-feu basé sur `iptables`, *IP Tables* filtre les paquets réseau indésirables ou le super serveur `xinetd` *eXtended INETD* (*) démarre les programmes fournissant des services Internet :

```
ex : /etc/rc.d/init.d/iptables permet d'exécuter le service des règles de filtrage IP
    /etc/rc.d/init.d/xinetd permet d'exécuter le super démon avec des services réseaux IP
    ps -ef | grep xinetd extrait et affiche le processus xinetd : exécution effectuée
    /etc/rc.d/init.d/httpd permet d'exécuter le démon du serveur http
    ps -ef | grep httpd extrait et affiche le processus httpd en cours d'exécution
```

(*) Note : `xinetd` est une version améliorée de `inetd`. Il permet une configuration plus fine de l'accès aux services (interdiction d'utilisateurs, d'adresses,...). Il n'y a plus un fichier unique mais un fichier `/etc/xinetd.conf` qui renvoie à un répertoire `/etc/xinetd.d`. Celui-ci contient un fichier par service configuré.

Service internet exécuté avec xinetd : telnet (exemple)

`telnet`, *Terminal NETwork* (*) est utilisé pour communiquer avec une machine *hôte* (host) distante :

```
ex : telnet ou encore telnet <nom ou ip de l'adresse de l'hôte distant>
    telnet> open 192.168.1.3 indique à l'adresse à ouvrir
    Trying 192.168.1.3... teste la connexion sur l'hôte distant
    Connected to 192.168.1.3. hôte distant connecté
    Escape character is '^]'. caractère d'échappement
    Telnet Server réponse du serveur
    login: toto invite de connexion (nom de l'utilisateur sur la machine hôte : toto)
    Password: mot de passe de l'utilisateur toto
    Last login: Sun Jan 16 18:35:20 on :0 dernière connexion effectuée
    [toto@hostname toto]$ invite de l'hôte distant sur le compte de l'utilisateur toto
```

Si le serveur (server) `telnet` est installé sur la machine hôte, ce service peut-être lancé par le super serveur ou super démon `xinetd` grâce à un script se trouvant dans le répertoire `/etc/xinetd.d` :

```
ex : vi /etc/xinetd.d/telnet édition du script telnet : afin d'exécuter telnet le mot clé
disable dans la ligne du script doit-être activé (disable=no).
```

(*) Note : `telnet` n'étant pas sécurisé on préférera `ssh`, abordé dans la rubrique suivante "Les commandes de réseau sous UNIX et GNU/Linux", puis le détail de l'installation et de la configuration dans le chapitre 12.

Vérification de l'état d'un service :

Tous les scripts de lancement se situent dans `/etc/rc.d/init.d`. Selon le `runlevel` (voir le chapitre 2) dans lequel on se situe, le service est arrêté ou démarré. Pour cela, consulter les répertoires `/etc/rc.d/rcN.d` où `N` est le numéro de `runlevel`. Ces répertoires sont constitués de liens symboliques. Si le lien commence par un `s`, le service est démarré. S'il commence par un `k`, il est arrêté. Le `s` ou `k` est ensuite suivi par un nombre à deux chiffres. Ce nombre permet de déterminer l'ordre de lancement des scripts (ordre croissant).

La plupart des scripts est utilisable avec 3 arguments : `start` pour le démarrer, `stop` pour l'arrêter et `status` pour connaître son état. Ces scripts acceptent également l'argument `restart` (`stop` puis `start`) pour redémarrer le service. C'est le cas d'un service en *exécution seule* (standalone) sans le super serveur :

```
ex : /etc/rc.d/init.d/httpd status vérification de l'état du démon httpd avec status
      httpd est arrêté le serveur httpd (Apache) ne fonctionne pas
      /etc/rc.d/init.d/httpd start démarre le serveur httpd avec l'argument start
      Démarrage de httpd : [ OK ] démarrage du serveur Apache réussi (tout va bien)
      /etc/rc.d/init.d/httpd status vérification de l'état du démon httpd
      httpd (pid 14113 14112 14111 14110 14109 14106) en cours d'exécution
```

(*) Note : le terme *standalone* désigne un service qui tourne sans être lancé et contrôlé par le super démon `xinetd`.

Création d'un script du système de démarrage des services (System V) :

La création d'un script de gestion du service dans le répertoire `/etc/rc.d/init.d` est nécessaire afin de fabriquer les liens symboliques de démarrage et d'arrêt dans les répertoires d'état de marche :

ex : pour ce faire nous examinerons un script déjà existant afin d'en comprendre l'implémentation

```
vi /etc/rc.d/init.d/sshd le script sshd permet d'exécuter le service du shell sécurisé
```

Ce qui suit permet de créer et mettre en place le script du service de `mon_serveur` pour le démarrer (`S=start`) aux niveaux 3, 4 et 5. Il faut procéder en deux étapes :

- créer un script de gestion des arguments `start/stop` : `/etc/rc.d/init.d/mon_serveur`
- créer des liens symboliques suivants :

```
ex : ln -s /etc/rc.d/init.d/mon_serveur /etc/rc.d/rc0.d/K05mon_serveur
      ln -s /etc/rc.d/init.d/mon_serveur /etc/rc.d/rc1.d/K05mon_serveur
      ln -s /etc/rc.d/init.d/mon_serveur /etc/rc.d/rc2.d/K05mon_serveur
      ln -s /etc/rc.d/init.d/mon_serveur /etc/rc.d/rc3.d/S95mon_serveur
      ln -s /etc/rc.d/init.d/mon_serveur /etc/rc.d/rc4.d/S95mon_serveur
      ln -s /etc/rc.d/init.d/mon_serveur /etc/rc.d/rc5.d/S95mon_serveur
      ln -s /etc/rc.d/init.d/mon_serveur /etc/rc.d/rc6.d/K05mon_serveur
```

La commande `ln -s` crée le lien symbolique `/etc/rc.d/rc0.d/Kxxmon_serveur` du script `mon_serveur` situé dans `/etc/rc.d/init.d/` (`xx` : est le nombre du lancement des scripts).

Configurer le démarrage d'un service : chkconfig

La commande `chkconfig` est spécifique aux distributions Redhat/Fedora et Mandrake. Elle gère les informations des niveaux d'exécution pour les services système (l'ensemble des services est intégré à une base de données). Grâce à cette commande, vous pouvez ajouter / supprimer / modifier les niveaux auxquels sont démarrés ou arrêtés un service :

ex : `chkconfig --add service` ajout d'un service à la base

`chkconfig --del service` suppression d'un service de la base

`chkconfig --level <niveaux> service on|off` configuration des niveaux auxquels le service doit démarrer/s'arrêter : où `niveaux` spécifie les valeurs des `runlevel`.

Autres services : daemon (*)

`httpd`, *HyperText Transfer Protocol (Daemon)* serveur Apache du protocole de transfert hypertexte (HTTP). Il est conçu pour fonctionner indépendamment comme un processus démon .

`named`, *Name (Daemon)* serveur de noms (DNS) ; ce processus démon permet d'associer un nom aux adresses IP des ordinateurs du réseau.

`proftpd`, *Professional File Transfer Protocol (Daemon)* serveur pour le transfert de fichiers (FTP)

`vsftpd`, *Very Secure File Transfer Protocol (Daemon)* serveur pour le transfert de fichiers (FTP) très sécurisé. Ces deux serveurs FTP sont également des démons.

(*) Note : *démon (daemon) Disk And Extension MONitor* : un démon n'est pas un programme ordinaire comme une commande shell qui se termine en affichant sa sortie, c'est à dire un processus qui n'est pas invoqué manuellement mais attend en tâche de fond que quelque chose se passe, que quelque condition se produise. Ce terme fut introduit au départ sous CTSS, *Compatible Time Sharing System*, un ancêtre du système MULTICS, lui même parent d'UNIX (Unix is not Multics).