UPMC
PARIS UNIVERSITAS

Préparation LPI

Exam 102

110. Securité

Sommaire

- inetd et xinetd
- Configuration SSH
- GPG

1.113.1 inetd et xinetd

- Une machine Linux peut proposer un certain nombre de service au monde extérieur accessible via le réseau
- un service = un démon (programme toujours actif) qui « écoute » un port particulier en attente d'une requête d'un client
  - httpd : démon en attente d'une requête d'une page html sur le port TCP 80
- Problème : chaque démon est consommateur de ressource (mémoire)
- Risque de saturation du système
- On souhaite activer les services à la demande uniquement. Solution : gestionnaire de services
- `inetd` ou `xinetd`

1.113.1 inetd et xinetd

- *Inetd* est un meta-serveur qui s'intercale entre le client et le service. Rôle de « chef-d'orchestre »
- En écoute constante sur certains ports, il active les démons correspondant aux services « à la demande » des clients
  - exemple de configuration `/etc/inetd.conf` pour le service ftp

```
# grep ^ftp /etc/inetd.conf
ftp stream  tcp^   nowait   root  /usr/local/sbin/ftpd
```

  - la correspondance entre nom de service et numéro de port se trouve dans `/etc/services`

```
# grep ^ftp /etc/services
ftp-data        20/tcp
ftp             21/tcp
```

1.113.1 inetd et xinetd

- Champs du fichier `/etc/inetd.conf`
  - service_name : nom du service définit dans `/etc/services`
  - socket-type : type de communication du service (`stream` pour protocole tcp ou `dgram` pour protocole udp habituellement)
  - proto : `tcp` ou `udp`
  - flags : `wait/nowait.connmax`. Généralement nowait pour les services tcp (multithreadé) et wait pour les services udp. connmax : nombre d'instances par minute
  - user[.group] : user et éventuellement groupe sous lequel le service doit s'exécuter
  - server_path : chemin de l'exécutable du service ou tcpd lorsqu'il est utilisé
  - args : arguments du programme serveur si nécessaire

1.113.1 inetd et xinetd

- inetd lancé par un script au démarrage
- En cas de modification du fichier de configuration, il faut envoyer à inetd un signal de reconfiguration
  - `kill -HUP `cat /var/run/inetd.pid``
  - `killall -HUP inetd`

1.113.1 inetd et xinetd

- Il manque à inetd la sécurisation d'accès aux services (filtrage)
- TCP_Wrapper est un outil destiné à apporter cet élément à inetd
- Il s'installe entre inetd et les démons qui doivent être lancés
- Il permet de filtrer l'adresse IP source des clients
- Il permet d'enregistrer dans des journaux le traffic (accepté ou refusé) engendré par les services

```
# grep ^ftp /etc/inetd.conf
ftp stream  tcp^   nowait   root  /usr/sbin/tcpd /usr/local/sbin/ftpd
```

1.113.1 inetd et xinetd

- La sécurisation d'accès est conditionnée par les fichiers `/etc/hosts.allow` et `/etc/hosts.deny`
  - Accès autorisé si correspondance trouvée dans `/etc/hosts.allow`
  - Accès refusée si correspondance trouvée dans `/etc/hosts.deny`
  - Sinon: accès autorisé
- On trouve dans ces 2 fichiers des couples "`service: client`"
  -
```
ftp: .ipsl.jussieu.fr
rsync:  134.157.12.87
cvs:    ALL
in.tftp:  LOCAL
```
- cf. `man hosts.allow` pour les détails de configuration

1.113.1 inetd et xinetd

- Le fichier `/etc/hosts.deny` est consulté après `/etc/hosts.allow` : possibilité de définir des exceptions même si la règle par défaut est de tout interdire

Quelques conseils LPI...

- Connaître le contenu et les directives contenue dans inetd.conf
- Connaître la combinatoire hosts.allow et hosts.deny
- Quels services peuvent être pris en charge par inetd
- Comment reconfigurer inetd

1.113.1 inetd et xinetd

- **`xinetd`** associe en un seul logiciel inetd et TCP_Wrapper avec des fonctionnalités supplémentaires
  - Accès horaire
  - Journalisation des accès
  - Association d'un service avec une @ IP particulière
- Installé en standard dans les distributions récentes
- Configuration :
  - `/etc/xinetd.conf` : fichier de configuration global. Contient la configuration par défaut pour tous les services
  - `/etc/xinetd.d` : répertoire contenant des fichiers de configuration spécifiques à chaque service
- détails : `man xinetd.conf`

1.113.1 inetd et xinetd

- **xinetd** ne prend pas en compte les fichiers contenu dans xinetd.d qui contiennent un point (.) ou qui se terminent par le caractère tilde (~)
  - But : éviter de prendre en compte les sauvegardes

1.113.1 inetd et xinetd

- /etc/xinetd.conf

```
# more /etc/xinetd.conf
#
# Simple configuration file for xinetd
#
# Some defaults, and include /etc/xinetd.d/

defaults
{
        instances                       = 60
        log_type                        = SYSLOG authpriv
        log_on_success                  = HOST PID
        log_on_failure                  = HOST
        cps                             = 25 30
}

includedir /etc/xinetd.d
```

- journalisation
  - **log_type** : *determine la sortie des journaux*
    - *SYSLOG syslog_facility [syslog_level] :* prise en charge par syslog
    - *FICHIER fichier [soft_limit [hard_limit]] :* écriture dans un fichier spécifique (/var/log/xinetd.log par exemple)

1.113.1 inetd et xinetd

- /etc/xinetd.d/rsync

```
# more /etc/xinetd.d/rsync
# default: off
# description: The rsync server is a good addition to an ftp server,
as it \
#       allows crc checksumming etc.
service rsync
{
        disable = yes
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/bin/rsync
        server_args     = --daemon
        log_on_failure  += USERID
        only_from       = 192.168.45.0/24
}
```

1.113.1 inetd et xinetd

- Chaque section décrivant un service doit respecter la syntaxe suivante :

```
service nom_du_service
{
        attribut opérateur valeur(s)
}
```

- opérateurs :
  - « = » : fixe la valeur d'un attribut
  - « += » : ajoute un élément à la liste de valeurs
  - « -= » : retire un élément à la liste de valeurs

1.113.1 inetd et xinetd

- attributs :
  - **disable** : *yes* ou *no* pour autoriser ou interdire le service
  - **socket_type** :*stream* pour TCP; *dgram* pour UDP
  - **wait** : *yes* si le service n'accepte qu'un seule connexion simultanée; *no* sinon
  - **user** : id de l'utilisateur qui exécute le service
  - **server** : nom du programme correspondant au service
  - **server-args** : arguments optionnels du service
  - **only_from** : liste des adresses IP autorisées à se connecter au service
  - **log_on_success(failure)** : niveau de détail des informations enregistrées dans le journal en cas de réussite (échec)

1.113.1 inetd et xinetd

- attributs suite:
  - **bind** : *association avec une adresse IP particulière*
  - **id** : étiquette particulière présente dans les journaux
- format des adresses IP (clause only_from)
  - une ou plusieurs adresses IP de machine séparées par un espace
    only_from = 192.168.45.5 192.168.45.10 225.158.16.45
  - une ou plusieurs adresses de réseau en notation CIDR
    only_from = 192.168.45.0/24

1.113.1 inetd et xinetd

- Comment faire pour démarrer

1. Vérifier le chemin et le nom du service
2. Copier le fichier de configuration d'un service existant, l'enregistrer sous un autre nom et la modifier en fonction du service à ajouter
3. Relancer le démon xinetd afin qu'il prenne en compte votre nouveau service

```
# kill -HUP `cat /var/run/xinetd.pid `
```

4. Vérifier dans le journal que la reconfiguration de xinetd est correcte

```
# tail /var/log/messages

Sep 14 10:51:10 ephora xinetd[1841]: Reconfigured: new=0 old=1
dropped=0 (services)
```

5. Tester la connexion au service

Sommaire

- inetd et xinetd
- Configuration SSH
- GPG

Configuration SSH

- Configuration serveur ssh
  - `/etc/.ssh/sshd_config`
  - Options
    - Protocol [1[,2]] : version du protocole SSH supportée par le serveur
    - PermitRootLogin [yes|no] : autorise ou non la connexion ssh avec l'utilisateur root
    - X11Forwarding [yes|no] : active le tunneling du protocole X

Configuration SSH

- Commande `ssh-keygen`
  - Permet la création d'un coule de clef privée et publique
  - Création des clef dans le répertoire ~/.ssh
  - Noms par défaut (fonction de l'algo) :
    - ~/.ssh/id_rsa et ~/.ssh/id_rsa.pub
    - ~/.ssh/id_dsa et ~/.ssh/id_dsa.pub
  - Options
    - -t [dsa|rsa] : choix de l'algorithme de chiffrement
    - -f : permet de spécifier un nom de fichier de clef spécifique

Configuration SSH

- Commande `ssh-keygen...`
  - Permissions correctes pour la K privée : 600

Configuration SSH

- Commande `ssh-agent`
  - Permet de conserver en mémoire 1 ou plusieurs clefs privées
  - A lancer au démarrage d'une session X ou au login
  - A associer avec la commande `ssh-add`
- Commande `ssh-add`
  - Ajoute à l'agent d'authentification une clef privée
  - Tente de charger par défaut : ~/.ssh/id_rsa, ~/.ssh/id_dsa, ~/.ssh/identity
  - Demande de la passphrase

Configuration SSH

- Fichier `~/.ssh/authorized_keys`
  - Contient les clefs publiques pour un compte donné
- Fichier `/etc/ssh_known_hosts`
  - Contient les clefs d'hôtes disponibles pour l'ensemble des comptes de la machine
- Fichier `~/.ssh/known_hosts`
  - Contient les clefs d'hôtes disponibles pour un compte donné

Sommaire

- inetd et xinetd
- Configuration SSH
- GPG

Configuration GPG

- Commande `gpg --gen-key`
  - Génère un couple de clé privée son fichier keyring

- Clé stockées dans `/~.gnupg`

```
$ ll .gnupg/
total 32
-rw------- 1 franck franck 9364 2010-02-15 10:43 gpg.conf
-rw------- 1 franck franck 1192 2010-03-31 22:55 pubring.gpg
-rw------- 1 franck franck 1192 2010-03-31 22:55 pubring.gpg~
-rw------- 1 franck franck  600 2010-03-31 22:55 random_seed
-rw------- 1 franck franck 1341 2010-03-31 22:55 secring.gpg
-rw------- 1 franck franck 1280 2010-03-31 22:55 trustdb.gpg
```

Configuration GPG

- Commande `gpg --export nom`
  - Permet d'exporter sa clé publique
  - nom : nom complet saisi lors de la création de la clef ou adresse mail
  - Option :
    - `--armor` : sortie ASCII pour un envoi par courriel de la clé publique
    - `--output nomfichier` : sortie dans un fichier plutôt que stdout

Configuration GPG

```
$ gpg --export --armor
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.4.9 (GNU/Linux)

mQGiBEuztmgRBADqE/YuVEJXYdcWsEZyzfz1bSRe9zSURHAEMnlXtKdPV93LbAIU
g6l/nJoVFTMwjX85bHju+oGs+O7gSKDMDI5h0SID5Y7d6ch30LZ6Gr7BnG2OJcEc
x0Fzwz3bbf8mDAnmyBc1F2bfMla3T3zCKHZJxhmO8C42/frrnObShe5LdwCgqYDc
3gyFle+DMcLr1KHeyX0mAFcD/2v89PQgc6/Tuf/CVLPkRFj18+4XHFDtfkHDq1+3
aK7kS2iKDGa1XUQtcVCQe9FKWJYmaioY53Qu+nktc3UGCUd+JEePu9bZBJMMDRQO
kVnKEtRNstm4GxsYr0xzJaJvNzMZBAeAzmjRURnY9PBSYehpn20vgeWOTP/qTvzZ
RB4dA/0XrGNljTWkW3LsWeKibpF+XmWDhxUy3sLla744CCZ1XUTdXlscqh/6qxp/
...
i/Q0VGStuH2Y5QCgqQVbwSf9RBgBolDUp209BXTI4NQ=
=5Rgb
-----END PGP PUBLIC KEY BLOCK-----
```

Configuration GPG

$ **gpg --export --armor --output .gnupg/ma_k_gpg.pub** \
franck.corsini@ipsl.jussieu.fr
franck@port-105:~$ ll .gnupg/
total 36
-rw------- 1 franck franck 9364 2010-02-15 10:43 gpg.conf
-rw-r--r-- 1 franck franck 1718 2010-03-31 23:07 ma_k_gpg.pub
-rw------- 1 franck franck 1192 2010-03-31 22:55 pubring.gpg
-rw------- 1 franck franck 1192 2010-03-31 22:55 pubring.gpg~
-rw------- 1 franck franck 600 2010-03-31 22:55 random_seed
-rw------- 1 franck franck 1341 2010-03-31 22:55 secring.gpg
-rw------- 1 franck franck 1280 2010-03-31 22:55 trustdb.gpg

Configuration GPG

- Commande `gpg --import cle-publique`
  - Permet d'importer une clé publique
  - Importation dans le fichier `~/.gnupg/pubring.gpg`

Configuration GPG

- Commande `gpg --list-keys`
  - Affiche la liste des K publiques de son keyring

```
$ gpg --list-keys
/home/franck/.gnupg/pubring.gpg
------------------------------
pub   1024D/33C3A9C9 2010-03-31
uid                  Franck Corsini (Admin IPSL) <franck.corsini@ipsl.jussieu.fr>
sub   2048g/4C588C56 2010-03-31
```

Configuration GPG

- Commande `gpg --out fichier-chiffré --recipient uid --armor --encrypt fichier-origine`
  - Chiffre un fichier
  - uid : spécifie l'uid (sinon gpg le demande) de l'utilisateur et sa k publique destinataire

Configuration GPG

```
$ cat fic_enclair.txt
Texte en clair ....
$ gpg --out fic_chiffré.gpg --recipient franck.corsini@ipsl.jussieu.fr --armor --encrypt
fic_enclair.txt
$ cat fic_chiffré.gpg
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.4.9 (GNU/Linux)

hQIOAzUkGuVMWIxWEAgArkNLFVJnZUZgUKkFUViEvzae6s5NrgOPKLpOiuW8oNhO
wV0U/gKgN5JaOTfVyUP8KMmck8aWAms5mhv+7k6vpFU83W2SmI45FrBEE4nvy0tx
ony6hVlh5r+WBj8JRuknU14HEmA5YyXLyqa9mAT1LbLR0IYSxT971VlDv9kM7rmS
WZJdGknrpwrsUaBN036X4IUG3n3Y5qTQdwqgIx0TMzKJt9h6vqNn31uzbjtUj7GM
....
t5fUjUpZ0Heg8b1VsMPObkjsnNMZQZEjSqKYk5hHkI0r/Rf4taIN0Tq5d4aT4f4Z
g4snOfB6qUaOXZ+ycbKFJ69lMOChUJxfUnPvXWq8yde9icVj5jtEX2ic2KHMC387
wtJZAYNEc4HTjei5rpz24F0pyRhoWy3w8fJBk3uWq0lT4kuSdGYU2x0SSJP895oK
bHhpIsxfgkqp5AAgzU6+HY9aDqEoWlNRyc8gBznwZ0wOZcK2veC7rwPO7Gk=
=Bivu
-----END PGP MESSAGE-----
```

Configuration GPG

- Commande `gpg --out fichier-déchiffré --decrypt fichier-chiffré`
  - déchiffre un fichier

```
$ gpg --decrypt fic_chiffré.gpg

Vous avez besoin d'une phrase de passe pour déverrouiller la
clé secrète pour l'utilisateur: « Franck Corsini (Admin IPSL) <franck.corsini@ipsl.jussieu.fr> »
clé de 2048 bits ELG-E, ID 4C588C56, créée le 2010-03-31 (ID clé principale 33C3A9C9)

gpg: gpg-agent n'est pas disponible dans cette session
gpg: chiffré avec une clé de 2048 bits ELG-E, ID 4C588C56, créée le 2010-03-31
      « Franck Corsini (Admin IPSL) <franck.corsini@ipsl.jussieu.fr> »
Texte en clair ....
```