sshd_config

>>> Traduction du fichier de configuration

Description :

Voici la traduction du fichier SSHD_Config.
SSHD lit des données de configuration dans le fichier /etc/ssh/sshd_config (ou du fichier spécifié à l'aide de l'option -f sur la ligne de commande). Ce fichier contient des paires mot-clef/argument, une par ligne.

sshd_config

>>> Traduction du fichier de configuration

I) Synopsis

```
nano /etc/ssh/sshd_config
```

II) Description

Les lignes qui commencent par le caractère « # » et les lignes vides sont interprétées comme des commentaires. Les mots-clefs possibles et leur signification sont listés ci-après (Note : les mots-clefs ne sont pas sensibles à la casse, mais les arguments le sont) :

```
AFSTokenPassing :
```

Spécifie si on peut rediriger un jeton AFS (AFS token) vers le serveur. Par défaut « no ».

```
AllowGroups :
```

Ce mot-clef peut être suivi d'une liste de motifs de nom de groupe, séparés par des espaces. S'il est spécifié, seuls les utilisateurs dont le groupe principal ou les groupes supplémentaires correspondent à un des motifs sont autorisés à se connecter. On peut utiliser les caractères « * » ou « ? » comme jokers. Seuls les noms de groupes sont valides ; les identifiants de groupes (GID) numériques ne sont pas reconnus. Par défaut, la connexion est autorisée pour tous les groupes.

```
AllowTcpForwarding :
```

Spécifie si les redirections TCP sont autorisées. Par défaut « yes ». Note : la désactivation des redirections TCP n'améliore pas la sécurité si les utilisateurs ont un accès à un interpréteur de commandes (shell), car ils peuvent toujours installer leurs propres outils de redirections.

```
AllowUsers :
```

Ce mot-clef peut être suivi d'une liste de motifs de noms d'utilisateurs, séparés par des espaces. S'il est spécifié, seuls les noms d'utilisateurs correspondant à un des motifs sont autorisés à se connecter. On peut utiliser les caractères « * » ou « ? » comme des jokers. Seuls les noms d'utilisateurs sont valides ; les identifiants d'utilisateurs (UID) ne sont pas reconnus. Par défaut, la connexion est autorisée pour tous les utilisateurs. Si le motif est de la forme UTILISATEUR@MACHINE, alors UTILISATEUR et MACHINE sont vérifiés séparément, en restreignant les connexions à des utilisateurs en particulier sur des machines en particulier.

```
AuthorizedKeysFile :
```

Spécifie le fichier contenant les clefs publiques à utiliser pour l'authentification de l'utilisateur. AuthorizedKeysFile peut contenir des jetons de la forme %T qui sont substitués lors des réglages de la connexion. Les jetons suivant sont définis : %% est remplacé par le caractère « % », %h est remplacé par le répertoire de base (home directory) de l'utilisateur qui s'authentifie et %u est remplacé par le nom de cet utilisateur. Après substitution, AuthorizedKeysFile peut être un chemin absolu ou relatif au répertoire de base de l'utilisateur. Par défaut « .ssh/authorized_keys ».

```
Banner :
```

Pour certaines juridictions, l'envoi d'un message avant l'authentification est nécessaire pour disposer d'une protection légale. Le contenu du fichier spécifié est envoyé à l'utilisateur distant avant d'autoriser la connexion. Cette option n'est disponible qu'avec la version 2 du protocole. Par défaut, on n'affiche pas de message.

```
ChallengeResponseAuthentication :
```

Spécifie si on autorise l'authentification par stimulation-réponse (challenge response). Toutes les formes d'authentification de login.conf5 sont gérées. Par défaut « yes ».

```
Ciphers :
```

Spécifie le cryptage autorisé pour le version 2 du protocole. On peut en spécifier plusieurs en les séparant par des virgules. Par défaut :

« aes128-cbc,3des-cbc,blowfish-cbc,cast128-cbc,arcfour,

aes192-cbc,aes256-cbc »

```
ClientAliveInterval :
```

Règle un intervalle de temporisation en secondes après lequel, si aucune donnée n'est reçue de la part du client, sshd envoie un message dans le canal crypté pour demander une réponse du client. Par défaut 0, ce qui signifie que ces messages ne sont pas envoyés au client. Cette option ne s'applique qu'à la version 2 du protocole.

```
ClientAliveCountMax :
```

Règle le nombre de messages de maintien de la connexion (voir ci-dessus) à envoyer sans réponse de la part du client pour sshd Si ce seuil est atteint tandis que les messages de maintien de la connexion ont été envoyés, sshd déconnecte le client et termine la session. Il est important de noter que ces messages de maintien de la connexion sont très différents de l'option KeepAlive (ci-dessous). Les messages de maintien de la connexion sont envoyés par le tunnel crypté, et par conséquent ne sont pas falsifiables. Le maintien de la connexion au niveau TCP activé par l'option KeepAlive est falsifiable. Le mécanisme de maintien de la

connexion est intéressant quand le client ou le serveur ont besoin de savoir si la connexion est inactive. Par défaut 3. Si l'option ClientAliveInterval (ci-dessus) est réglée à 15, et ClientAliveCountMax est réglée à sa valeur par défaut, les clients ssh qui ne répondent pas sont déconnectés après environ 45 secondes.

```
Compression :
```

Spécifie si on autorise la compression. L'argument doit être « yes » ou « no ». Par défaut « yes ».

```
DenyGroups :
```

Ce mot-clef est suivi d'une liste de motifs de noms de groupes, séparés par des espaces. Les utilisateurs dont le groupe principal ou les groupes secondaires correspondent à un des motifs ne sont pas autorisés à se connecter. Dans les motifs, on peut utiliser les caractères « * » et « ? » comme des jokers. On ne spécifie que des noms de groupes ; les identifiants numériques de groupes ne sont pas autorisés. Par défaut, tous les groupes sont autorisés à se connecter.

```
DenyUsers :
```

Ce mot-clef est suivi d'une liste de motifs de noms d'utilisateurs, séparés par des espaces. Les utilisateurs dont le nom correspond à un des motifs ne sont pas autorisés à se connecter. Dans les motifs, on peut utiliser les caractères « * » et « ? » comme des jokers. On ne spécifie que des noms d'utilisateurs ; les identifiants numériques d'utilisateurs ne sont pas autorisés. Par défaut, tous les utilisateurs sont autorisés à se connecter. Si le motif est de la forme UTILISATEUR@MACHINE, UTILISATEUR et MACHINE sont vérifiés séparément, et la connexion est restreinte à certains utilisateurs de certaines machines.

```
GatewayPorts :
```

Spécifie si les machines distantes sont autorisées à se connecter à des ports redirigés par le client. Par défaut, sshd branche les redirections de ports à l'adresse de bouclage (loopback address). Ceci évite que les autres machines distantes ne se connectent aux ports redirigés. On peut utiliser l'option GatewayPorts pour spécifier que sshd doit brancher les redirections de ports distantes à l'adresse joker, et par conséquent autoriser les machines distantes à se connecter à des ports redirigés. L'argument doit être « yes » et « no ». Par défaut « no ».

```
HostbasedAuthentication :
```

Spécifie si on autorise une authentification par rhosts ou /etc/hosts.equiv conjointement avec une authentification de machine cliente réussie par clef publique (authentification par machines). Cette option est similaire à l'option RhostsRSAAuthentication et ne s'applique qu'à la version 2 du protocole. Par défaut « no ».

```
HostKey :
```

Spécifie un fichier contenant une clef privée de machine utilisée par SSH. Par défaut /etc/ssh/ssh_host_key pour la version 1 du protocole, et /etc/ssh/ssh_host_rsa_key et /etc/ssh/ssh_host_dsa_key pour la version 2

du protocole. Note : sshd refuse d'utiliser un fichier accessible au groupe ou aux autres. On peut avoir plusieurs fichiers de clef de machine. Les clefs « rsa1 » sont utilisées pour la version 1 du protocole, et les clefs « dsa » ou « rsa » sont utilisées pour la version 1 du protocole SSH.

```
IgnoreRhosts :
```

Spécifie que l'on utilise pas les fichiers .rhosts et .shosts pour les authentification activées par les options RhostsAuthentication RhostsRSAAuthentication ou HostbasedAuthentication
Les fichiers /etc/hosts.equiv et /etc/ssh/shosts.equiv sont néanmoins utilisés. Par défaut « yes ».

```
IgnoreUserKnownHosts :
```

Spécifie si sshd doit ignorer le fichier $HOME/.ssh/known_hosts de l'utilisateur lors des authentifications des options RhostsRSAAuthentication ou HostbasedAuthentication Par défaut « no ».

```
KeepAlive :
```

Spécifie si le système doit envoyer des messages TCP de maintien de la connexion. Si ces messages sont envoyés, la rupture d'une connexion ou le plantage d'une des machines seront correctement signalés. Toutefois, cela signifie que la connexion sera interrompue si la route entre le serveur et le client est temporairement coupée, et quelques personnes trouvent ceci gênant. D'un autre côté, si on n'envoie pas de messages de maintien de la connexion, il est possible que des sessions restent en suspens indéfiniment sur le serveur, en laissant des utilisateurs fantômes, et en consommant les ressources du serveur.
Par défaut « yes » (pour envoyer les messages de maintien de la connexion), et le serveur signale les coupures réseau ou les plantages des machines. Ceci évite les sessions indéfiniment en suspens. Pour désactiver ces messages de maintien de la connexion, il faut régler cette valeur à « no ».

```
KerberosAuthentication :
```

Spécifie si on autorise l'authentification Kerberos. Elle peut être de la forme d'un ticket Kerberos, ou si l'option PasswordAuthentication est réglée à « yes », le mot de passe fourni par l'utilisateur est validé par le KDC Kerberos. Pour utiliser cette option, le serveur a besoin d'un servtab Kerberos qui autorise la vérification de l'identité du KDC. Par défaut « no ».

```
KerberosOrLocalPasswd :
```

Si cette option est réglée, alors si l'authentification par mot de passe par Kerberos échoue, le mot de passe est validé via n'importe quel mécanisme local tel que /etc/passwd Par défaut « yes ».

```
KerberosTgtPassing :
```

Spécifie si on redirige un TGT Kerberos vers le serveur. Par défaut « no », car ceci ne fonctionne que si le KDC Kerberos est vraiment un kaserver AFS.

```
KerberosTicketCleanup :
```

Spécifie si on détruit automatiquement le fichier cache du ticket de l'utilisateur à la déconnexion. Par défaut « yes ».

```
KeyRegenerationInterval :
```

Dans la version 1 du protocole, la clef éphémère du serveur est regénérée automatiquement après ce nombre de secondes (si elle a été utilisée). Le but de la regénération est d'éviter le décryptage de sessions capturées en s'introduisant plus tard sur la machine et en volant la clef. La clef n'est jamais stockée nulle part. Si la valeur est 0, la clef n'est jamais regénérée. Par défaut 3600 (secondes).

```
ListenAddress :
```

Spécifie l'adresse locale d'écoute sur laquelle le démon sshd doit attendre les connexions. On peut utiliser les formes suivantes :

```
ListenAddress host | IPv4_addr | IPv6_addr

ListenAddress host | IPv4_addr : port

ListenAddress [host | IPv6_addr : port ]
```

Si port n'est pas spécifié, le démon sshd écoute sur l'adresse et toutes les options Port spécifiées au préalable. Par défaut, on écoute sur toutes les adresses locales. On peut spécifier de multiples options ListenAddress En outre, toute option Port doit précéder cette option pour les adresses sans port spécifié.

```
LoginGraceTime :
```

Le serveur se déconnecte après ce délai si l'utilisateur ne s'est pas connecté. Si la valeur est 0, il n'y a aucune limite de temps. Par défaut 600 (secondes).

```
LogLevel :
```

Donne le niveau de verbosité utilisé lors de l'enregistrement des messages du démon sshd Les valeurs possibles sont : QUIET, FATAL, ERROR, INFO, VERBOSE, DEBUG, DEBUG1, DEBUG2 and DEBUG3. Par défaut INFO. DEBUG et DEBUG1 sont équivalents. DEBUG2 et DEBUG3 spécifient des niveaux plus élevés de sortie de débogage. L'enregistrement à l'aide d'un niveau DEBUG a tendance à empiéter sur la vie privée des utilisateurs et n'est pas recommandé.

```
MACs :
```

Spécifie les algorithmes MAC (code d'authentification de message) disponibles.
L'algorithme MAC est utilisé dans la version 2 du protocole pour la protection de l'intégrité des données. On peut spécifier plusieurs algorithmes en les séparant par des virgules. Par défaut « hmac-md5,hmac-

sha1,hmac-ripemd160,hmac-sha1-96,hmac-md5-96 ».

```
MaxStartups :
```

Spécifie un nombre maximal de connexions concurrentes au démon sshd non authentifiées. Les connexions supplémentaires sont purgées si elles ne peuvent pas s'authentifier ou si le délai de grâce défini à l'aide de l'option LoginGraceTime expire pour une connexion. Par défaut 10.
Par ailleurs, on peut activer une purge hâtive aléatoire en spécifiant un triplet « début:taux:total » (par exemple, « 10:30:60 »). sshd refuse les tentatives de connexion avec une probabilité de « taux/100 » (30 %) s'il y a « début » (10) connexions non authentifiées en cours. La probabilité augmente linéairement et toutes les tentatives de connexion sont refusées si le nombre de connexions non authentifiées atteint « total » (60).

```
PAMAuthenticationViaKbdInt :
```

Spécifie si l'authentification PAM par stimulation/réponse est autorisée. Ceci permet d'utiliser la plupart des modules d'authentification PAM par stimulation/réponse, mais peut autoriser l'authentification par mot de passe sans se soucier de l'option PasswordAuthentication.

```
PasswordAuthentication :
```

Spécifie si l'authentification par mot de passe est autorisée. Par défaut « yes ».

```
PermitEmptyPasswords :
```

Si l'authentification par mot de passe est autorisée, spécifie si le serveur autorise les connexions à des comptes dont les mots de passe sont des chaînes de caractères vides. Par défaut « no ».

```
PermitRootLogin :
```

Spécifie si root peut se connecter par ssh(1). L'argument est « yes », « without-password », « forced-commands-only » ou « no ». Par défaut « yes ».
– Si cette option est réglée à « without-password », l'authentification par mot de passe est désactivée pour root.
– Si cette option est réglée à « forced-commands-only », les connexions de root sont autorisées avec une authentification par clef publique, mais seulement si l'option command est spécifiée (ce qui peut être utile pour effectuer des sauvegardes à distance même si les connexions de root sont normalement interdites). Toutes les autres méthodes d'authentification sont désactivées pour root.
– Si cette option est réglée à « no », root n'est pas autorisé à se connecter.

```
PidFile :
```

Spécifie l'emplacement du fichier contenant l'identifiant du processus du démon sshd Par défaut /var/run/sshd.pid.

```
Port :
```

Spécifie le port d’écoute du démon sshd Par défaut 22. On peut spécifier plusieurs de ces options. Voir aussi ListenAddress.

```
PrintLastLog :
```

Spécifie si sshd doit afficher la date et l’heure de la dernière connexion de l’utilisateur. Par défaut « yes ».

```
PrintMotd :
```

Spécifie si sshd doit afficher le contenu du fichier /etc/motd quand un utilisateur se connecte en mode interactif. Sur certains systèmes, il est aussi affiché par l’interpréteur de commandes (shell), par le fichier /etc/profile ou équivalent. Par défaut « yes ».

```
Protocol :
```

Spécifie les versions du protocole que le démon sshd gère. Les valeurs possibles sont « 1 » et « 2 ». Pour spécifier plusieurs versions, il suffit de les séparer par des virgules. Par défaut « 2,1 ».

```
PubkeyAuthentication :
```

Spécifie si on autorise l’authentification par clef publique. Par défaut « yes ». Note : Cette option ne s’applique qu’à la version 2 du protocole.

```
RhostsAuthentication :
```

Spécifie si l’authentification par les fichiers rhosts ou /etc/hosts.equiv est suffisante. Normalement, cette méthode ne devrait pas être autorisée parce qu’elle n’est pas sécurisée. Il est préférable d’utiliser RhostsRSAAuthentication à la place, parce qu’elle effectue une authentification de machine basée sur RSA en plus de l’authentification normale par rhosts ou /etc/hosts.equiv. Par défaut « no ». Cette option ne s’applique qu’à la version 1 du protocole.

```
RhostsRSAAuthentication :
```

Spécifie si on autorise une authentification par rhosts ou /etc/hosts.equiv couplée avec une authentification de machine RSA réussie. Par défaut « no ». Cette option ne s’applique qu’à la version 1 du protocole.

```
RSAAuthentication :
```

Spécifie si on autorise la pure authentification RSA. Par défaut « yes ». Cette option ne s’applique qu’à la version 1 du protocole.

```
ServerKeyBits :
```

Définit le nombre de bits de la clef éphémère pour la version 1 du protocole. La valeur minimale est 512 et la valeur par défaut est 768.

```
StrictModes :
```

Spécifie si sshd doit vérifier les modes et le propriétaire des fichiers de l'utilisateur et du répertoire de base (home directory) de l'utilisateur avant d'accepter une connexion. C'est normalement souhaitable, parce que quelquefois, les novices laissent accidentellement leur répertoire ou leurs fichiers en accès complet à tout les monde. Par défaut « yes ».

```
Subsystem :
```

Configure un sous-système externe (par exemple un démon de transfert de fichiers). Les arguments doivent être un nom de sous-système et une commande à exécuter lors d'une requête à ce sous-système. La commande sftp-server8 implémente le sous-système de transfert de fichiers « sftp». Par défaut, aucun sous-système n'est défini. Note : Cette option ne s'applique qu'à la version 2 du protocole.

```
SyslogFacility :
```

Donne le code de facilité utilisé lors de l'enregistrement des messages du démon sshd Les valeurs possibles sont : DAEMON, USER, AUTH, LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, LOCAL7. Par défaut AUTH.

```
UseLogin :
```

Spécifie si on utilise login(1) pour les connexions à des sessions interactives. Par défaut « no ». Note 1 : On n'utilise jamais login(1) pour l'exécution de commandes à distance. Note 2 : Si cette option est activée, on désactive X11Forwarding parce que login(1) ne sait pas traiter les cookies xauth(1). Si on spécifie l'option UsePrivilegeSeparation elle sera désactivée après l'authentification.

```
UsePrivilegeSeparation :
```

Spécifie si sshd sépare les privilèges en créant un processus fils non privilégié pour prendre en charge le trafic réseau entrant. Après une authentification réussie, un autre processus est créé avec les privilèges de l'utilisateur authentifié. Le but de la séparation de privilèges est d'éviter l'escalade de privilèges si le processus non privilégié est corrompu. Par défaut « yes ».

```
VerifyReverseMapping :
```

Spécifie si sshd essaie de vérifier que le nom de la machine distante et le nom obtenu par résolution de l'adresse IP distante correspondent à la même adresse IP. Par défaut « no ».

```
X11DisplayOffset :
```

Spécifie le premier numéro d'affichage disponible pour les redirections X11 de sshd Ceci évite à sshd d'interférer avec les vrais serveurs X11. Par défaut 10.

```
X11Forwarding :
```

Spécifie si on autorise les redirections X11. Par défaut « no ». Note : La désactivation des redirections X11 n'améliore en aucun cas la sécurité, puisque les utilisateurs peuvent installer leurs propres redirecteurs. La redirection X11 est automatiquement désactivée si l'option UseLogin est activée.

```
X11UseLocalhost :
```

Spécifie si sshd branche le serveur X11 de redirection sur l'adresse de bouclage (loopback address) ou sur l'adresse joker (wildcard address). Par défaut sshd branche le serveur de redirection sur l'adresse de bouclage et règle la partie du nom de machine de la variable d'environnement DISPLAY à « localhost ». Ceci évite à des machines distantes de se connecter à de faux affichages. Néanmoins, quelques vieux clients X11 peuvent ne pas fonctionner avec cette configuration. On peut régler X11UseLocalhost à « no » pour spécifier que le serveur de redirection est branché sur l'adresse joker. L'argument doit être « yes » ou « no ». Par défaut « yes ».

```
XAuthLocation :
```

Spécifie l'emplacement du programme xauth(1). Par défaut /usr/X11R6/bin/xauth.
Formats de durées :
On peut exprimer les arguments de la ligne de commande et les options du fichier de configuration de sshd qui spécifient des durées à l'aide d'une séquence de la forme : time [qualifier] où time est une valeure entière positive et qualifier est l'un des suivants :

Tous les membres de la séquence sont additionnés pour obtenir la valeur totale de la durée.

Exemples de formats de durées :

```
600

600 secondes (10 minutes)

10m

10 minutes

1h30m

1 heure 30 minutes (90 minutes)
```

III) Fichiers

```
/etc/ssh/sshd_config
```

Contient les données de configuration de sshd Ce fichier ne doit être accessible en écriture qu’à root, mais il est recommandé (bien que pas nécessaire) qu’il soit accessible en lecture à tous.