Syslog

- syslogd: daemon chargé de gérer les journaux d'une machine
  - journaux: /var/log/*.log (en général)
- peut gérer les journaux d'hôtes distants
  - option « -r » à positionner explicitement
  - rfc 3164: BSD Syslog protocol
  - udp port 514
  - supporté par de nombreux type d'équipement réseau: un standard incontournable

Syslog

- sécurité:
  - pas d'authentification, de filtrage des sources,
  - pas de chiffrement des informations
  - udp: non connecté, pas d'assurance de délivrance

Syslog

- gestion des journaux:
  - gaffe classique: un disque plein à cause de journaux accumulés
  - outils de gestion des journaux : logrotate, newsyslog, ...: compresser, déplacer, effacer, ...

Syslog

- analyse des journaux:
  - pour détecter un problème et/ou en déterminer les causes **après coup**
  - pour alerter d'un problème **en cours**
  - des rapport d'analyse de journaux trop long ne sont pas (plus) lus. Il faut :
    - réagir rapidement aux choses graves
    - extraire les informations pertinentes de la masse d'information
  - Deux types d'outils
    - outils d'analyse de journaux: logcheck, logsurfer, swatch, sec, ...
    - via un ids: système de détection d'intrusion

syslog-ng:

- configuration plus souple
- classement des messages par leur contenu, par l'hôte d'origine
- meilleure redirection des messages sur le réseau
- possibilité de chroot
- peut utiliser UDP et TCP
- chiffrement et authentification du trafic réseau
- portable
- export des journaux vers un sgbd

configuration: syslog.conf

- facilité.niveau<tab>action
- facilité: type de service source

Action
fichier
terminal
pipe |
@machineDistante
utilisateur1,utilisateur2,...
*

Niveau	
emerg (panic)	Situations de panique.
alert	Situations urgentes.
crit	Situations critiques.
err (error)	Erreurs.
warning (warn)	Messages de WARNING.
notice	Messages divers.
info	Messages d'informations.
debug	Débogage.

Facilités	
kern	Le noyau.
user	Process des utilisateurs.
mail	Système de courrier.
daemon	Démons systèmes.
auth	Authentification.
lpr	Système de spooling d'imprimante.
news	Usenet.
uucp	UUCP.
cron	Démon cron.
mark	Messages generes a intervals réguliers.
local0-7	Huit niveaux de messages locaux.
syslog	Messages internes a syslogd.
authpriv	Messages privés auth.
*	Toutes les facilités sauf mark.

Syslog : demo

- lister le syslog d'un système existant
- lister un journal de /var/log, montrer les entrées "MARK" insérées par syslogd
- tester son comportement avec la commande logger
  - logger -p mail.crit "boîte au lettre en feu :-) »
  - logger -p news.err "pas de nouvelles, bonne nouvelle"
  - comparer l'effet avec le contenu de syslog.conf et notamment que le message est stocké si son niveau est supérieur ou égal à celui de la règle
- le modifier en y insérant une entrée
- tester l'entrée insérée avec logger

Bibliographie sur la supervision et sur syslog

- « unix, guide de l'administrateur » de Nemeth, Snyder & Al, Campus press
- « MISC No 22 » (revue): superviser sa sécurité
- Ntsyslog: http://ntsyslog.sourceforge.net/
- http://www.linux-kheops.com/line/html/line/line-dec1996/datas/syslog.htm
-

comptes utilisateurs: création

- uid
- modifier /etc/passwd & Co
- mot de passe
- dossier personnel
- fichier d'initialisation dans $HOME
- donner les bons droit au dossier perso (chgrp, chown)
- déclarer l'utilisateur dans les services usuels (mail, ...)
- tester le compte

comptes utilisateurs

- structure d'un fichier /etc/passwd
- passwd: pour changer son mot de passe
- shadows passwords: /etc/shadow
- commande d'administration :
  - dépend du système d'exploitation
  - exemples:
    - useradd/adduser
    - userdel

groupes

- /etc/group
- chaque utilisateur a un groupe initial (/etc/passwd) et des groupes secondaires (/etc/group)
- groups: liste les groupes de l'utilisateur
- groupes sous BSD:
  - l'utilisateur appartient à tous les groupes
  - création de dossier/fichier: groupe du dossier père
  - gestion des groupes: pw (création/suppression, ajout d'utilisateurs, ...)

groupes

- groupe sous SysV et Linux
  - l'utilisateur appartient à un instant donné à un seul groupe => newgrp pour changer de groupe
  - création de dossier/fichier: groupe du dossier père ou groupe de l'utilisateur (Linux, autorisé par SysV)
  - gestion des groupes
    - groupadd, groupmod, groupdel: ajout/suppression de groupes
    - usermod -G group,... login: ajoute login au(x) groupe(s)

planification de tâches: cron et atd

- cron: tâches planifiées régulières
- atd: exécution unique
- cron et arrêt systèmes/chgt d'heures
- commande crontab:
  - crontab -l : lister
  - crontab -r : supprimer
  - crontab -e : modifier
- dossier daily, monthly, ...: (dépend de l'OS)

format du fichier crontab

- régles communes:
  - # en début de ligne indique un commentaire
  - les champs sont séparés par des espaces
  - les espaces de la commandes sont laissés inchangés. commande exécutée par sh
  - dans la commande, % indique un saut de ligne
  - contenu des champs :
    - *, entier, entier-entier, des entiers/intervalles séparés par des virgules
- crontab utilisateur :

```
minute heure jourDuMois jourDeLaSemaine commande
```

- crontab sytème (souvent : /etc/crontab)

```
minute heure jourDuMois jourDeLaSemaine utilisateur
  commande
```

crontab: exemples

- commandes valides :

```
echo date courante: `date` >> /tmp/test
mutt -s "coucou Pascal" petit@shayol.org % coucou
  % courrier de test
find / -xdev -name core -atime +7 -exec /bin/rm
  -f {} \;
```

- spec de temps valides:

```
*0 * * * * : toutes les 10 mn
10 2 * * * : tous les jours à 2h10
0 23 * * 0 : tous les dimanches à 23h00
0 20-23,0-7,10,12,14,16,18 * * * : toutes les
  heures entre 20h00 et 7h00 puis toutes les deux
  heures
```

cron : sécurité

- contrôle d'accès :
  - cron.allow: seuls utilisateurs habilités à programmer des tâches
  - cron.deny: seuls utilisateur NON autorisés à programmer des tâches (suppose l'absence de cron.allow)
  - si ni cron.(allow|deny): seul root y a droit
- contrôle d'accès réalisé par la commande crontab
  - => les fichiers crontab doivent avoir les bons droits

Chiffrement : définitions

services offerts par le chiffrement:

- confidentialité
- intégrité: chiffrer une empreinte du message
- signature numérique
- authentification (ex.: ssh qui authentifie les machines)
- kerberos: authentification centralisée unique
- non répudiation: prouver qui a créé un message: utilisation de tiers de confiance, chiffrement à clef publique

Chiffrement: robustesse

- cryptanalyse: analyser une information chiffrée pour la déchiffre (dont des méthodes en force brute, ...)
- algo public
- la sécurité repose sur :
  - la non divulgation de la clef
  - la robustesse de l'algorithme
  - la taille de la clef (gare aux comparaisons entre algo différents)
  - l'utilisation de clefs différentes pour chiffrer des messages différents limite la quantité d'information à la disposition de l'attaquant

chiffrement: taille des clefs

- attaques en force brute: tenter une partie importante de l'espace des clefs
- temps dépend du nombre de clefs possibles et donc de la taille de la clef:
  - 10 bits : 1024 clefs possibles
  - 56 bits: 2^56=7 10^16
  - dépendance exponentielle en fonction de la taille de la clef: 1 bit de plus = 2 fois plus de temps
- la taille critique dépend de l'algo (et de sa vitesse, de ses faiblesses, ...)

algorithme de chiffrement

- chiffrement symétrique/asymétrique
  - symétrique: les algo classiques sont rapides
    - la même clef sert au chiffrement et au déchiffrement
    - souvent utilisé via une clef de session
      - clef de session: transmise via algo asymétrique (on parle d'enveloppe digitale)
      - session: chiffrée par un algo symétrique et la clef transmise
  - asymétrique: les algo classiques sont lents
    - couple de clef publique/clef privée
      - clef publique: peut être connue de tous
      - clef privée: tenue cachées
      - ce qui est chiffré avec l'une ne peut être déchiffré qu'avec l'autre

- Algorithmes de chiffrement symétrique:
  - DES (1976): standard américain (1977), clef de 56 bits sur des blocs de 64 bits. dépassé de nos jours.
  - triple DES (1978): variante via une triple application de DES permettant d'avoir des clefs entre 128 et 192 bits sur des blocs de 64 bits.
  - RC2, RC4, RC5 (1994) et RC6:
  - IDEA (1992): clef 128 bits sur des blocs de 64 bits
  - blowfish: clef 32 à 448 bits sur des blocs de 64 bits. Algo très analysé,considéré comme solide. utilisation libre.
  - AES (1998): clefs 128, 192 ou 256 bits sur blocs de 128 bits. standard américain. utilisation libre.

algorithmes classiques

- asymétriques:
  - RSA s'appuyant sur la factorisation de nombres premiers
  - Diffie-Hellman et El Gamal s'appuyant sur le calcul des logarithmiques discrets
  - des algorithmes nouveaux s'appuyant sur les courbes elliptiques

durée de vie des clefs

- dépend de sa taille
- dépend de son taux d'utilisation
- dépend du contexte d'utilisation
- hiérarchie de clef (clef maîtresse, clef de session par ex.)
- révocation de clef
- une utilisation intensive du chiffrement nécessite la mise en place d'une IGC (infrastructure de gestion de clef ou PKI – Public Key Infrastructure en anglais)

hachage/ empreinte

- principe:
  - une fonction non réversible H:
    - connaissant H(x), il est très difficile de trouver y tel que H(y)=H(x)
  - telle que deux empreintes différentes correspondent forcément à deux textes différents
  - la probabilité d'avoir deux empreintes identique est très faible

hachage: applications

- authentification des utilisateurs:
  - on stocke la version hachée du mot de passe
  - un grain de sel permet d'éviter que deux personnes qui ont le même mot de passe aient la même empreinte
- copie optimisée de fichiers
- vérification de l'intégrité de fichiers

Hachage: algo classiques

- MD4 (mdp windows NT & Co)
- MD5 (mdp unix): empreinte de 128 bits, considéré comme faible (collisions)
- sha-1: empreintes de 160 bits (solidité mise en doute actuellement)
- sha-2: empreintes de 256, 384 ou 512 bits au choix
- utilisation d'un algo de chiffrement: le mot de passe est transformé en clef pour chiffrer un texte connu. ex. connu: DES modifié itéré 25 fois pour les mots de passe unix.

Identification et authentification

- **identification**: définir l'identité de l'utilisateur
- **authentification**: permet de vérifier l'identité fournie (authentification simple vs authentification forte)
  - via un élément que l'utilisateur connait (mot de passe, ...)
  - via un élément que l'utilisateur possède (carte à puce, certificat, ...)
  - via biometrie

authentification

- élément clef pour assurer :
  - la confidentialité et l'intégrité des données via un contrôle d'accès: seules les personnes identifiées, authentifiées et habilités à le faire peuvent accèder/modifier les données
  - la non-répudiation et l'imputabilité (preuve d'une transaction, ...)
- Authentification unique (SSO: Single Sign On)
  - l'utilisateur s'authentifie une fois
  - il a accès à toutes les ressources du réseau
  - cf partie technique (keberos, ...)

Authentification de base sous unix

- authentification par login/mot de passe
- l'emprunte du mot de passe (+ un peu de sel): stockée dans /etc/passwd ou /etc/shadow ou ~
- Algo: des, md5, blowfish
- lorsqu'un utilisateur s'authentifie
  - on calcule l'emprunte (+ le sel) du mot de passe qu'il fournit
  - on compare le résultat à l'emprunte stockée

Authentification sous unix: PAM

- PAM: pluggable authentification modules
- mécanismes permettant d'intégrer des modes d'authentification variés via une interface unique
- via la configuration de PAM (et l'existence du module concerné), on peut faire supporter à de nombreux unix des systèmes d'authentifications variés (carte à puce, mot de passe jetables, annuaire LDAP, ...)

Authentification de base sous windows

- 2 algo de chiffrement: LanMAN (faible) et NTLM
  - pour rester compatible avec un parc ancien
  - mdp < 15 car. chiffrés en LanMAN et en NTLM
- attaques sur LanMAN: de quelques secondes à quelques heures pour trouver un mot de passe alphanumérique par force brute
- LanMAN:
  - désactivable sur les windows 2000sp2+
  - désactivé sur windows Vista

LanMAN: algorithme

- mot de passe tronqué à 14 caractères ou mis à 14 (bourrage avec des caractères nuls)
- mis en majuscule et coupé en deux parties de 7 caractères
- chaque partie:
  - utilisée comme clef de chiffrement DES à 56 bits pour chiffrer la chaîne « KGS!@#$% »
  - on concatène les deux résultats de 8 octets pour obtenir une emprunte LanMAN de 16 octets

attaque par force brute

- on calcule l'emprunte de tout ou partie de l'espace des mots de passe et on compare à l'emprunte stockée
- attaques utilisant des jeux d'empruntes totalement ou partiellement pré-calculées
- attaque par dictionnaires (+ modifications classiques)
- outils: lc4 (windows), john the ripper

exemples d'autres attaques

- espionnage du réseau:
  - pour récupérer les mots de passe en clair
- remplacement d'une machine par une autre:
  - l'utilisateur s'authentifie sur la machine du pirate en croyant s'authentifier sur un serveur
- compromission d'une machine
  - sur le serveur distante; on remplace les programmes de login et autres
  - sur le poste client: on met en place un keylogger (il en existe de compatibles avec les claviers virtuels)

SSH

- ssh est à la fois
  - un protocole
  - une commande
  - un ensembles d'outils dont il existe diverses version de diverses origines

SSH

- ssh permet de relier
  - des machines sûres et non compromises
  - à travers un réseau non sûr
    - but: éviter l'écoute passive ou active de la communication
  - l'ensemble des échanges est chiffré
  - les machines sont authentifiées

SSH

- authentification des machines
- chiffrement de session
- authentification des utilisateurs
- tunneling
- boîte à outil ssh

authentification des machines

- chaque machine a un couple clef privée/publique
- chaque machine doit avoir la clef publique de l'autre
- quand ce n'est pas le cas, cette clef peut être fournie par l'une des machines à l'autre qui la sauvera localement
  - dans ce cas, l'authentification de l'autre machine ne peut être garantie lors de cette première connexion
  - compromis pour faciliter l'adoption du protocole ssh face à la difficulté de diffuser les clefs de façon simple et sûre

Authentification des machine: processus

- les deux machines échangent des informations sur les protocoles de chiffrement qu'ils supportent (algo de chiffrement symétrique, à clef pub/priv, algo de hash, algo de signature de messages)
- le client génère une d'une clef de session pour algorithme symétrique
- il la transmet au serveur en la chiffrant avec la clef publique du serveur et indique l'algo de chiffrement utilisé
- le serveur envoie un message de confirmation chiffré avec le clef de session
- le reste de la communication est chiffrée avec la clef de session et l'algorithme de chiffrement symétrique choisi

Authentification des utilisateurs

- authentification par pam (mdp, one time password, ...)
- authentification par clef publique
  - l'utilisateur possède un couple clef privée/publique
  - la clef privée est sur la machine cliente protégée par une phrase d'accès
  - la clef publique est transférée par un moyen sur sur le serveur dans le fichier authorized_keys de l'utilisateur

authentification par clef publique

- l'utilisateur fournit la phrase d'accès à sa clef privée
- la machine client déchiffre la clef privée de l'utilisateur et l'utilise pour générer une signature qui est envoyée au serveur
- le serveur tente de valider cette signature à l'aide des clefs publiques présentes dans le fichier authorized_keys de l'utilisateur
- en cas de succès, l'accès est autorisé

processus du point de vue de l'utilisateur

- générer un couple clef publique/privée sur le poste client (ex.: ssh-keygen -t dsa. clef privée: id_dsa, publique: id_dsa.pub)
- tranférer la clef PUBLIQUE sur le serveur et l'ajouter au fichier contenant les clefs publiques de l'utilisateur (ex.: ~/.ssh/authorized_keys
- la connexion est ensuite possible sans mot de passe (si la stratégie de sécurité du serveur l'autorise)
- il est possible de placer des restrictions (IP d'origine, commande autorisée, ...) pour chaque clef présente dans le authorized_keys.

agents d'authentification: ssh-agent

- agent d'authentification ssh: mémorise les clefs en mémoire vive pour éviter à l'utilisateur de taper une clef à chaque utilisation
- principe: ssh-agent est le processus père (ou un ancètre) du processus qui réalise la connexion ssh
- en pratique:
  - ssh-agent est lancé au démarrage de la session graphique X
  - on lance à la main « ssh-agent bash » ou « ssh-agent xterm »

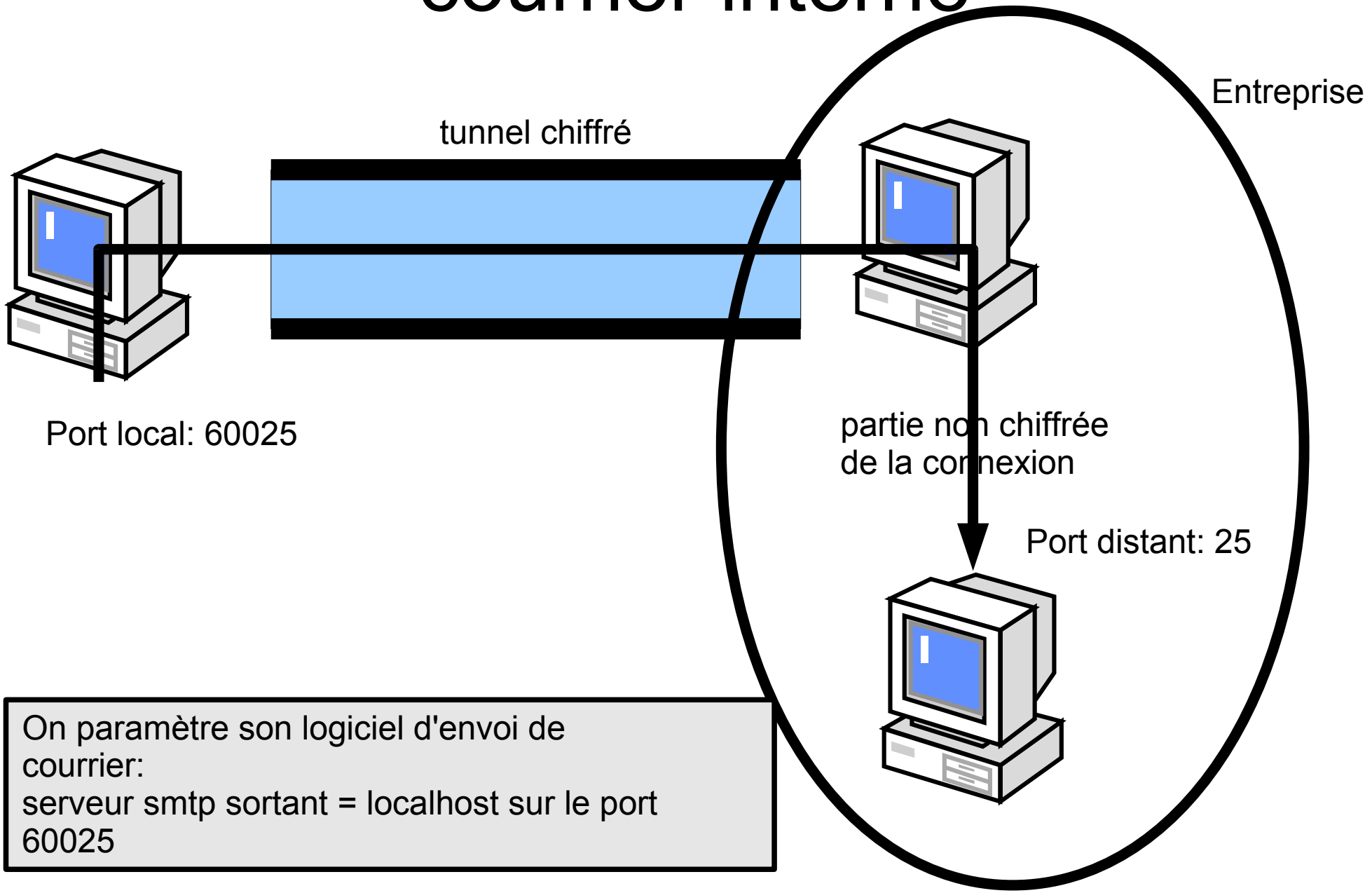
agents d'authentification: ssh-add

- ssh-add: commande utilisateur pour ajouter une clef en mémoire

tunnel SSH

- ssh permet de rediriger des connexions tcp effectuées sur un port donné du client vers un port donné d'une machine accessible depuis le serveur
- il permet de faire de même d'un port du serveur vers le client
- utilisation traditionnelle (option -X): redirection X11
- vpn du pauvre : accès à un intranet depuis internet

tunnel SSH: accès à un serveur de courrier interne



On paramètre son logiciel d'envoi de courrier:
serveur smtp sortant = localhost sur le port 60025