Cours 7 : Gestion des utilisateurs NIS
Accés aux fichiers distants NFS et SMB

Rabii El Ghorfi

Plan

Plan

Objectifs

- Centraliser les connexions sur un réseau local
  - Se connecter à un serveur de fichier sous un compte centralisé
  - Ne pas définir de compte machine par machine
- Réseau homogène Linux
  - Connexion et authentification grâce au service NIS
  - Accès aux répertoires partagés grâce à NFS
  - Pour utiliser des stations Windows : serveur SAMBA
- Serveur NIS
  - Au moins un par réseau
  - Plusieurs : soit un par domaine NIS soit serveurs coopératifs (un maître et des esclaves)

NIS

NIS (Network Information System):

- introduit par SUN en 1985 (Yellow Pages (yp) à l'origine)
- n'est pas un standard, mais est très largement utilisé
- une base de donnée distribuée qui permet le partage d'informations système (login, mot de passe, . . .)

Objectifs:

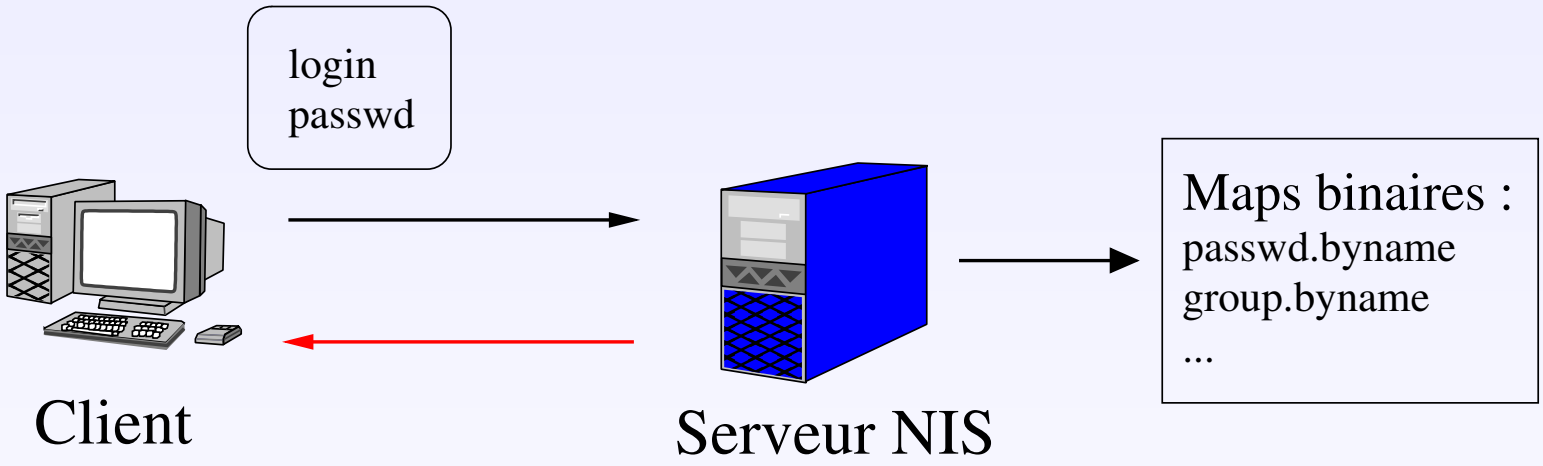
- simplifier la gestion des comptes, des mots de passe et les tâches d'administration dans le monde UNIX
- il suffit de créer un utilisateur sur le serveur NIS pour que chaque machine client NIS ait accès aux informations de *login* de cet utilisateur

Plan

Fonctionnement

NIS maintient une base de donnée au format DBM sur un domaine NIS

Exemple de fonctionnement :



- Au moins un serveur NIS par réseau
- Possibilité d'en avoir Plusieurs : soit un par domaine NIS soit serveurs coopératifs (un maître et des esclaves)

Architecture

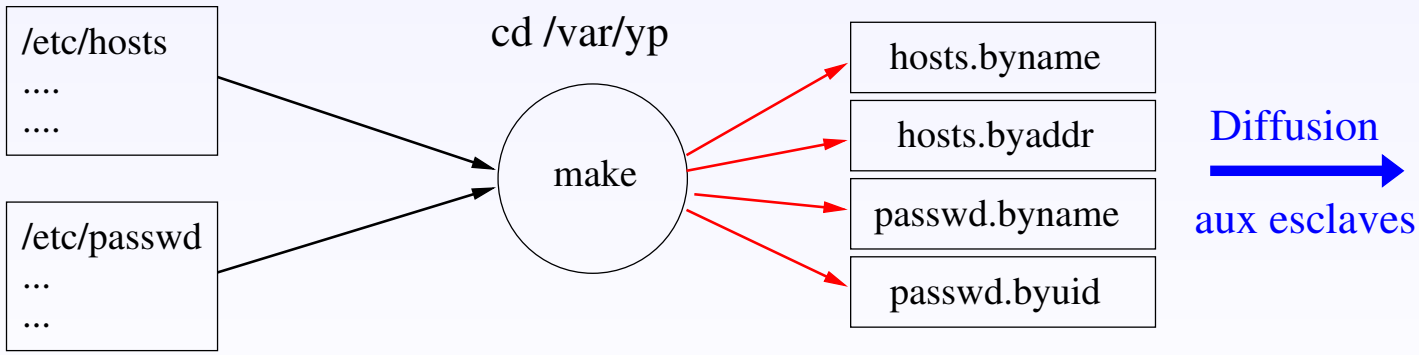
Achitecture : Découpage en domaines.

- modèle Client/Serveur au dessus des SUN-RPC
- un domaine NIS contient :
  - un serveur NIS maître qui maintient les *“maps”* (informations contenues dans la base)
  - aucun, un ou plusieurs serveurs NIS esclaves :
    - permet de décharger le seveur NIS principal et d’être plus résistant aux pannes
    - le maître réplique ses informations vers les serveurs secondaires
    - seul le maître peut modifier une map
    - les esclaves diffusent les maps sans pouvoir les modifier (diminue les problèmes de cohérence)
- des clients NIS qui peuvent interroger les serveurs qu’ils soient maître ou esclaves.

Plan

En pratique . . .

- Les maps sont stockées dans le répertoire : `/var/yp/nom_de_domaine`
- Quand le fichier source d'une map est modifié sur le serveur (ajout d'un utilisateur, changement de mot de passe,. . .), il faut regénérer la map associée et enventuellement propager les modifications aux serveurs NIS esclaves
- Chaque map stocke des couples clé/valeur



En pratique . . .

- La commande `ypcat` permet de voir le contenu d'une map depuis n'importe quel client
- Au niveau d'Un client NIS :
  - il est nécessaire de se lier (binding) à un serveur pour pouvoir l'interroger
    - deux méthodes : diffusion (traiter la première réponse) ou désignation explicite d'un serveur
    - nom de domaine positionné à l'aide de la commande `domainname` ou dans le fichier `/etc/defaultdomain`, ...
    - Le demon `ypbind` doit tourner pour rechercher régulièrement le serveur approprié.
  - `ypwhich` permet de connaître le nom du serveur NIS
  - `ypset` permet de positionner le nom du serveur (pour désignier explicitement un serveur NIS)
  - possibilité de configuration du nom du serveur NIS correspondant à un nom de domaine dans le fichier `/etc/yp.conf` :

```
domain nom_de_domaine
ypserver nom_de_serveur
```

Configuration du client

2 méthodes :

`nsswitch.conf` :

- Détermine l'ordre de recherche pour l'authentification des utilisateurs.

```
hosts:     files dns nis
networks: nis files
```

pour les entrées passwd, group et shadow : 2 solutions

```
passwd:     files nis
```

ou

```
passwd:     compat
```

`/etc/passwd` :

- Ajouter +:::::: à la fin de `/etc/passwd` au niveau du client.

Remarque :

Il suffit de remplacer "+" par "-" pour exclure les utilisateurs NIS d'une marchine.

Configuration du serveur

- Un serveur NIS esclave doit faire tourner :
  - `ypserv` pour répondre aux requêtes de ses clients NIS
  - `ypbind` s'il est lui-même un client NIS (n'est pas obligatoire)
- Un serveur NIS maître doit faire tourner :
  - `ypserv` pour répondre aux requêtes de ses clients NIS
  - `ypbind` s'il est lui-même un client NIS (n'est pas obligatoire)
  - `ypxfrd` pour répondre aux demandes de mise à jour des maps de la part des serveurs esclaves
  - `rpc.yppasswd` pour assurer les demandes de changement de mot de passe (`passwd`)

Plan

NIS: évolutions

Défauts de NIS :

- Pas d'authentification des clients NIS : il suffit de connaître le nom de domaine pour interroger le serveur et connaître le contenu des maps.
- les maps sont transmises dans leur intégralité même en cas de faible modification de leur contenu.
- pas adapté aux WAN (broadcast,...)

NIS+ : Un successeur éphémère sans succès qui a été abandonné au profit de LDAP.

NIS: évolutions

Défauts de NIS :

- Pas d'authentification des clients NIS : il suffit de connaître le nom de domaine pour interroger le serveur et connaître le contenu des maps.
- les maps sont transmises dans leur intégralité même en cas de faible modification de leur contenu.
- pas adapté aux WAN (broadcast,...)

NIS+ : Un successeur éphémère sans succès qui a été abandonné au profit de LDAP.

Cependant, NIS continue à être largement utilisé.

Partie 2 : Accès aux fichiers distants NFS et SMB

Plan

1. Introduction
2. NFS
3. SAMBA

Plan

Accès aux fichiers distants

Différences avec le transfert de fichier :

- l'accès aux fichiers distants est complètement transparent pour l'utilisateur
- tout se passe comme si le système de fichier distant était local
- l'utilisateur peut éditer le fichier, le modifier, . . . ; les modifications seront répercutées sur le système fichier distant

Les deux principaux protocoles :

NFS. Network File System (Unix/Sun-RPC)

SMB. Sserver Message Block (issu du monde Microsoft)

Plan

NFS : Network File System

Présenté par SUN en 1985 pour permettre à ces stations sans disque d'accéder à un système de gestion de fichiers distants (RFC 1904).
Utilise les appels de procédures distantes Sun-RPC (qui sont issues des travaux sur NFS)

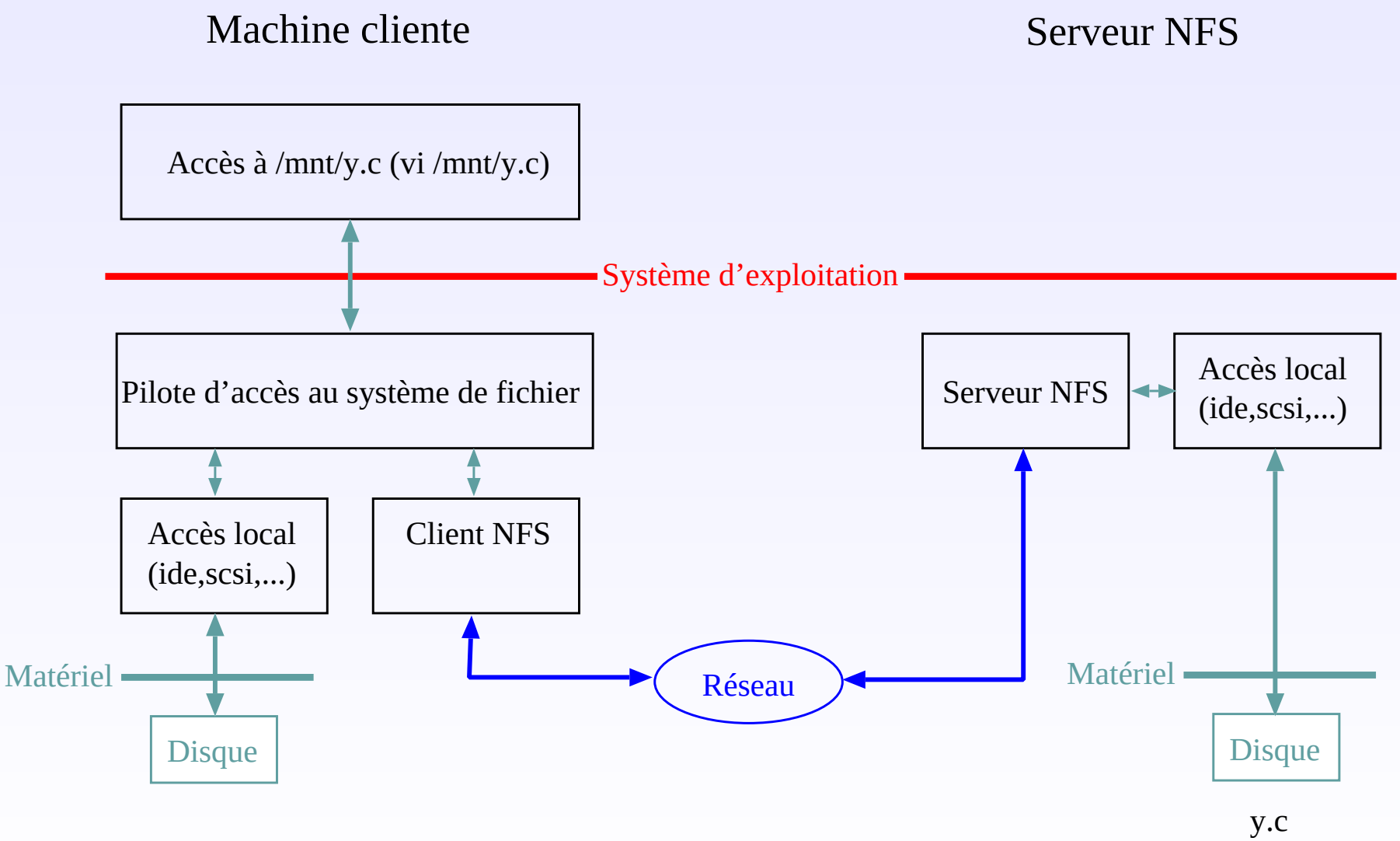
- à priori, les clients et serveur NFS devraient être des processus utilisateur s'exécutant au-dessus de RPC/XDR/UDP/IP.
- en fait, le client et le serveur NFS s'exécutent dans le noyau
  - le client pour rendre transparent l'accès à un fichier via NFS
  - le serveur pour des raisons d'efficacité

NFS : Network File System

Présenté par SUN en 1985 pour permettre à ces stations sans disque d'accéder à un système de gestion de fichiers distants (RFC 1904).
Utilise les appels de procédures distantes Sun-RPC (qui sont issues des travaux sur NFS)

- à priori, les clients et serveur NFS devraient être des processus utilisateur s'exécutant au-dessus de RPC/XDR/UDP/IP.
- en fait, le client et le serveur NFS s'exécutent dans le noyau
  - le client pour rendre transparent l'accès à un fichier via NFS
  - le serveur pour des raisons d'efficacité

Principe de fonctionnement



Les éléments d'accès aux fichiers

- Manipule des descripteurs, chemins, déplacements

Sytème d'exploitation

- Manipule des Files, Dentries, Inodes, déplacements
- Masque les différences à l'application (API uniforme, . . .)

Matériel

- Manipule des blocs
- Matériel-dépendant

Le choix entre NFS, EXT3,. . . se fait lors de l'ouverture du fichier.

NFS et les RPCs

NFS repose sur les RPC (Remote Procedure Calls)

- Utilisation du portmapper (programme portmap de Linux)
- Portmapper = conversion des n° de prog RPC en numéro de port

Déroulement d'une RPC :

- Serveur RPC :
  - → Indique à portmap le port qu'il utilise et les numéros de programme RPC qu'il gère
- Envoi d'une requête RPC par un client :
  - → Il contacte portmap du serveur pour connaître le numéro de port du programme souhaité
  - → Il envoie les données au port correspondant

Un serveur sans état

Dans un accès à un système de fichier local :

- Les accès reposent sur un pointeur de fichier maintenu au niveau du système d'exploitation

NFS est basé sur une connexion réseau :

$\Rightarrow$ Probabilité d'une panne importante

Solution :

- Le serveur NFS ne conserve aucune information sur les accès/opérations effectuées (fichiers ouverts, accès précédents au fichier,...)
- Le système d'exploitation du client NFS se charge de maintenir les informations concernant les fichiers

Intérêts :

- Simplifie le redémarrage du serveur en cas de crash.

Sécurité

Principe d'authentification :

- $(\texttt{uid}, \texttt{gid})_{\texttt{local}}$ "mappé" sur $(\texttt{uid}, \texttt{gid})_{\texttt{distant}}$
  $\rightarrow$ équivalence entre les droits locaux et les droits distants
- Problème pour `root` :
  - Quels droits possède le *root* d'une machine cliente sur les fichiers exportés par un serveur NFS?
  - par défaut `root` (coté client) correspond a l'utilisateur `nobody` (coté serveur) pour des raisons de sécurité (sinon il faut mettre l'option `no_root_squash` dans `/etc/exports`)

Règle de non transitivité :

- Si A exporte `/home` à B; Si B monte `A:/home` dans `/home2` et exporte `/home2` à C alors C n'aura pas accès au `/home` de A

Liens symboliques :

- Les liens symboliques relatifs sont interprétés par rapport au système de fichier du client.

NFS en pratique (1/2)

Démons importants utilisés par NFS :

- **portmap.** Gestion des connexions des applications utilisant le mécanisme de RPC.
- **nfsd.** Authentification + Création, recherche, lecture et écriture de fichiers
- **mountd.** Montage des systèmes exportés (mount et umount)
- **statd.** Surveillance des nœuds du réseau (redémarrages. . .)
- **lockd.** Section critique (lock les fichiers utilisés)

NFS en pratique (2/2)

coté client. le fichier `/etc/fstab` doit contenir le chemin vers le point de montage et le chemin sur le serveur NFS.

```
192.168.0.1:/home /nfs nfs defaults,noauto
0 0
```

coté serveur. le fichier `/etc/exports` contient le chemin vers les dossiers à exporter ainsi
que la liste les machines autorisées à y accéder. `/home 192.168.1.0/255.255.255.0(rw,no_root_squash)`

Après chaque modification de `/etc/exports` il est nécessaire :

- soit d'exécuter `exportfs` pour transmettre les modifications au serveur nfs
- soit de relancer le serveur NFS

Plan

SMB : Server Message Block

- Protocole de Microsoft et Intel permettant le partage de ressources (disques, imprimantes, . . .) à travers un réseau (1987)
- SMB est prévu pour être utilisé au dessus de l'interface NetBIOS
  - Utilisation des noms NetBIOS (15 caractères + 1 pour le type)
  - Utilisation du mécanisme de datagramme de NetBIOS par *broadcast* comme service de nommage (nom → MAC, pas d'adresse de niveau 3)

<table>
<tr><td colspan="3">Application</td></tr>
<tr><td colspan="3">SMB</td></tr>
<tr><td colspan="3">NetBIOS</td></tr>
<tr><td>TCP/IP</td><td>NetBEUI</td><td>IPX/SPX</td></tr>
<tr><td>802.x</td><td>PPP</td><td>...</td></tr>
</table>

SMB (1/2)

- Chaque machine client ou serveur possède un nom sur 15 caractères
- SMB ajoute un 16ème caractère pour distinguer les serveurs de fichiers, les clients, les imprimantes, ...
- Notion de domaine
  - un ensemble d'utilisateurs (avec nom et mot de passe) et de serveurs (avec des droits d'accès)
  - un *primary domain server* contient la base de données des utilisateurs et de leur mot de passe
- Un serveur une ou plusieurs ressources
  - fichiers, imprimantes, ...
  - à chaque triplet (domaine, serveur, ressource) correspond un nom unique : `\\serveur\ressource`

SMB (2/2)

Deux niveaux de protection :

- au niveau de chaque utilisateur : basé sur le nom des utilisateurs, permet de gérer l'accès aux ressources voire aux éléments d'une ressource
- au niveau de chaque ressource : un mot de passe commun à tous les utilisateurs est associé à une ressource pour y autoriser l'accès

Résolution de noms : 4 méthodes utilisées

broadcast. résolution par diffusion d'une requête dans le réseau

lmhost. résolution en utilisant des associations prédéfinies entre noms NetBIOS et IP

host. utilisation de DNS

wins. utilisation d'un serveur WINS (*Windows Internet Name Server*). À chaque machine est associé un serveur WINS à qui elle envoie ses requêtes et auprès duquel elle s'enregistre.

SAMBA (1/2)

Samba : Implémentation de SMB sous UNIX qui permet le partage de ressources entre les mondes UNIX et Windows
Samba permet de :

- partager un disque UNIX pour des machines Windows
- accéder à un disque Windows depuis une machine UNIX
- partager une imprimante UNIX pour des machines Windows
- utiliser une imprimante Windows à partir d'un hôte Linux.

Le serveur Samba sur la machine Unix émule un domaine SMB

SAMBA (2/2)

Serveur Samba :

- configuration via le fichier `/etc/smb.conf`
- travail partagé par deux démons :
  - `smbd`. pour le service “serveur”
  - `nmbd`. pour le service résolution des noms NetBIOS

Client :

`smbpasswd`. permet de changer le mot de passe d’un utilisateur SMB

`smbclient`. permet d’interroger un serveur Samba depuis UNIX `smbclient //host/ressource` permet l’accès à la ressource

Possibilité de monter une partition Windows distante à l’aide de Samba → utiliser le système de fichier `smbfs`

Exemple : (extrait du fichier `/etc/fstab`)

```
//serveur/ressource /commun smbfs defaults 0 0
```