

# Administration réseau

## Gestion des utilisateurs (NIS)

A. Guermouche

# Plan

1. introduction
2. Fonctionnement
3. Configuration de NIS
4. Conclusion

# Plan

1. introduction

2. Fonctionnement

3. Configuration de NIS

4. Conclusion

# 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 (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

1. introduction

**2. Fonctionnement**

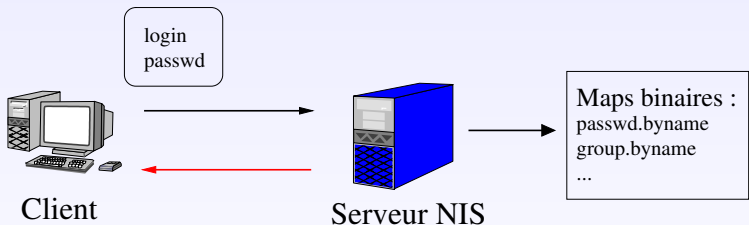
3. Configuration de NIS

4. Conclusion

# 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.

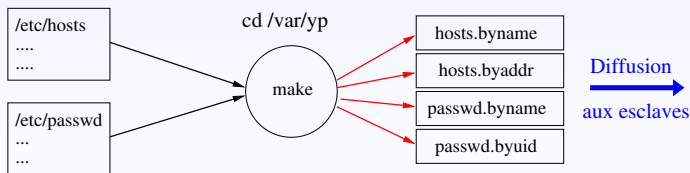


# Plan

1. introduction
2. Fonctionnement
- 3. Configuration de NIS**
4. Conclusion

# 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 server (ajout d'un utilisateur, changement de mot de passe,...), il faut régénérer la map associée et éventuellement 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 désigner 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 machine.

# 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

1. introduction
2. Fonctionnement
3. Configuration de NIS
4. Conclusion

# 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.