IUT 'A' Paul SABATIER
Dépt Informatique

ASR => Administration Réseau : Adr4

TP2: Système d'Information Réseau (NIS)

Objectif :

Présentation du Système d'Information Réseau (**NIS**) et mise en œuvre de serveurs et de clients sur des machines fonctionnant sous le système **Linux Debian**.

---

1. Introduction

Administrer un environnement distribué de stations de travail Unix/Linux présente un problème majeur, à savoir le maintien de différentes copies séparées des fichiers de configurations comme les fichiers de mots de passe, les fichiers de groupes ou encore les fichiers des noms de machines.

L'utilisateur d'un parc de machines ne doit pas se soucier de savoir sur quelle machine il dispose d'un compte ni s'il va pouvoir atteindre une nouvelle machine sur le réseau.

On a vu que le système de noms de domaines (**DNS**) permettait de mettre en place un système performant de résolution de noms de machines. Le problème demeure donc pour la gestion des utilisateurs et de leurs mots de passe.

Le Système d'Information Réseau (**NIS**) traite ces problèmes. Il peut être vu comme un système distribué de bases de données se substituant aux nombreuses copies des fichiers de configuration communs dont la gestion est centralisée.

Au lieu de traiter les fichiers d'une machine (**/etc/passwd**, **/etc/group**…), l'administrateur devra maintenir une base de données pour chaque fichier sur un serveur central et les machines utilisant **NIS** retrouveront ces informations à partir de ces bases de données.

Ainsi un utilisateur peut avoir le même login sur plusieurs machines et lorsqu'il souhaite modifier son mot de passe par exemple, il suffit de modifier un fichier sur le serveur central et de propager ensuite cette modification au reste du réseau, au lieu de modifier le fichier des mots de passe sur l'ensemble des machines pouvant accueillir cet utilisateur.

Comme **NIS** maintient des vues consistantes de fichiers sur un réseau, il est adapté à des fichiers ne comportant pas d'informations propres à une machine. Par exemple, les fichiers **/etc/fstab** des **« file system »** et leurs points de montage, sont de sérieux candidats à leur gestion par **NIS** parce qu'ils sont différents sur chaque machine.

Enfin, l'utilisation de **NIS** ne se résume pas à la gestion de fichiers de configuration, mais à tout type de base de données.

2. Principes conceptuels

NIS est construit selon le modèle Client/Serveur. Un serveur **NIS** est une application qui gère les fichiers de données **NIS**, appelés des "**maps**". Les clients sont des applications qui demandent des informations extraites de ces maps.
Les serveurs sont de 2 types : maître ou esclave. Le serveur **maître** est seul propriétaire des données des maps. Les serveurs **esclaves** traitent les requêtes des clients mais ne peuvent pas modifier les **maps NIS**. Le serveur maître est responsable du maintien des maps et de leur distribution aux serveurs esclaves à chaque mise-à-jour.

Afin de permettre à un administrateur de définir plusieurs politiques pour différents systèmes, **NIS** fournit la notion de **domaine**. Un **domaine** est un ensemble de **maps NIS** (à ne pas confondre avec la notion de domaine du service DNS).

Un client **NIS** peut référencer une map d'un quelconque domaine. Toutefois, un client n'accède la plupart du temps qu'à un même ensemble de maps. D'où il est commun de confondre la notion de domaine avec le groupe de machines partageant le même ensemble de **maps NIS**. Les machines devant partager des informations communes de configuration sont regroupées dans un domaine. Chaque machine est alors associée à un domaine par défaut.

En fonction de l'organisation d'un réseau et du nombre de machines, un ou plusieurs domaines **NIS** peuvent être définis. Pour chaque domaine, un ou plusieurs serveurs esclaves peuvent être déployés.

La gestion **NIS** de base comprend alors la mise en œuvre de serveurs **NIS** ainsi que l'habilitation **NIS** des clients. La gestion des serveurs comporte 3 tâches :

- Installer un nouvel environnement **NIS**, en construisant les serveurs maître et esclave(s).
- Lancer le démon **ypserv**, qui permet à un système d'agir en tant que serveur.
- Définir de nouveaux serveurs lorsque le réseau croît.

Permettre aux machines d'être des clients **NIS** nécessite de :

- Modifier les fichiers d'administration des clients pour que ceux-ci tirent avantage de **NIS**.
- Lancer le démon **ypbind**, qui permet au client de formuler les requêtes **NIS**.

Le système **NIS** est matérialisé par deux programmes **RPC** (Remote Procedure Call) :
- **ypserv** pour les serveurs.
- **ypbind** pour les clients.

Le protocole **NIS** est supporté par TCP et UDP et aucun port n'est fixé par défaut. Certaines procédures utilisent TCP, celles qui se connectent à un serveur pour lui demander des informations, d'autres UDP, comme une requête pour obtenir un nom de serveur par exemple.
L'attribution des ports se fait dynamiquement et leur obtention se fait par interrogation du processus démon **portmap**. La commande :

rpcinfo –p permet de déterminer les services **RPC** actifs sur une machine.

Le fichier **/etc/rpc** contient la définition des services **RPC** disponibles sur un système (numéro de programme, numéro de version et procédure(s)). La commande:

rpcinfo –u demande l'identification d'un programme.

Un service **RPC** qui s'exécute, doit d'abord s'enregistrer auprès du **portmapper** actif sur le serveur. Le client invoque un service auprès du **portmapper** qui lui retourne alors le numéro de port où ce service peut être fourni.

3. Serveurs NIS

Comme les clients vont devoir récupérer la plupart des informations de configuration auprès de **NIS,** les serveurs doivent être hautement disponibles. Si un serveur **NIS** arrête de répondre ou répond trop lentement, le client essaie d'en trouver un autre, moins chargé. Cela plaide pour la définition d'au moins un serveur esclave pour un serveur maître et leur installation sur des machines fiables.
La commande **domainname** permet de fixer le nom du domaine par défaut ou de l'afficher.
La fixation d'un domaine étant un préalable indispensable à l'utilisation du **NIS**, cette commande est la première à être exécutée. Un nom de domaine est une chaîne d'au plus 64 caractères. Il n'y a pas de valeurs interdites dans l'absolu, néanmoins par convention, un nom doit commencer par une lettre et ne doit contenir que des lettres, des chiffres et/ou le caractère '**-**'.
Lors de leur création, les bases de données (maps) destinées au serveur **ypserv** seront placées dans le répertoire **/var/yp/*nom-de-domaine***.

La notion de domaine est assez rigide, ceux-ci sont connexes, il n'est donc pas possible d'avoir une machine appartenant à plusieurs domaines. Il est pourtant utile de pouvoir choisir uniquement quelques machines pour telle commande, et d'autres machines du même domaine pour une autre. De même, il est souhaitable de sélectionner des utilisateurs dans certains cas. C'est à cela que sert le fichier **/etc/netgroup**.
Celui-ci contient une liste de groupes, à raison d'un par ligne. Un groupe est constitué d'une liste de membres qui sont soit des groupes, soit des triplets de la forme (machine, utilisateur, domaine). Un champ vide indique qu'on fait référence à tous les objets concernés par celui-ci.

Les procédures **ypserv** peuvent être classées selon 3 catégories :

- celles qui permettent d'identifier un serveur et le domaine qu'il sert.
- celles qui permettent d'accéder aux informations des maps.
- les procédures de maintien de la cohérence entre le serveur maître et les serveurs esclaves en transférant des maps.

La commande de lancement de **ypserv** est : **/usr/sbin/ypserv**.

La configuration des serveurs **NIS** (maître ou esclaves), est réalisée par l'intermédiaire de la commande **ypinit** qui est en fait un shell-script. Lors de la création d'un serveur maître, **ypinit** demande les noms des serveurs esclaves à qui il doit transférer les maps. Ces dernières sont ensuite créées et placées dans le répertoire d'accueil.

Les fichiers d'administration sont ceux du serveur maître qui est censé être sur la machine sur laquelle s'exécute **ypinit**.
Ceci n'est pas forcément une bonne chose, par exemple si l'on souhaite que certains utilisateurs ne puissent pas se loger sur le serveur, le fichier **/etc/passwd** doit être expurgé des leurs comptes.

Pour un serveur esclave, **ypinit** transfère les maps à partir du serveur maître et les place dans le répertoire d'accueil. S'il s'agit d'un nouveau serveur, il faut créer une entrée dans la map *ypservers* afin qu'il puisse recevoir les mises-à-jour.
La syntaxe de la commande **ypinit** sur le serveur maître est la suivante :

/usr/lib/yp/ypinit –m

La syntaxe de la commande **ypinit** sur chaque serveur esclave est la suivante :

/usr/lib/yp/ypinit –s *serveur_maître*

Les **maps NIS** sont au format **DBM** qui est un format dérivé des bases de données ASCII. Les maps sont en fait des projections des fichiers utilisant des clés différentes.
Ainsi pour le fichier **/etc/passwd**, les projections sont *passwd.byname* et *passwd.byuid* et les clés d'accès sont respectivement le login et l'UID de l'utilisateur.

Des abréviations sont définies pour certaines projections, ainsi *group* correspond à *group.byname*. Les projections contiennent deux entrées spéciales. L'une, de clé *YP_LAST_MODIFIED*, est associée à la date de création de la projection. La seconde a comme clé *YP_MASTER_NAME* et référence le nom du serveur maître.

Le fichier **/etc/netgroup** a trois projections, la première (*netgroup*) est indexée par le nom du groupe, les deux autres respectivement par le nom de machine (*netgroup.byhost*) et par le nom d'utilisateur (*netgroup.byuser*).

La projection *ypservers* est un peu particulière : elle ne correspond à aucun fichier. Elle contient le nom des serveurs, esclaves et maître, et est utilisée principalement par ce dernier pour effectuer les mises-à-jour.

Les fichiers **/etc/passwd** et **/etc/group** peuvent être convertis dans le format **DBM** en utilisant le programme **makedbm** disponible dans les programmes du serveur. Cette commande est rarement appelée directement mais plutôt au travers de shell-scripts.

Les dépendances des projections NIS des fichiers de configuration sont maintenues dans le fichier **/var/yp/Makefile** du serveur maître. Chaque fois qu'un fichier de configuration est modifié, il faudra exécuter le "**make**" afin de mettre à jour la projection concernée par la mise-à-jour.

Les transferts de projections se font par l'intermédiaire des commandes **yppush** et **ypxfr**.
La commande **yppush** est exécutée sur le serveur maître et déclenche l'exécution de la commande **ypxfr** sur les serveurs esclaves. Lors de son exécution, **yppush** constitue d'abord une liste des serveurs esclaves grâce à la projection *ypservers*. La syntaxe est :

/usr/lib/yp/yppush [-v] [-d *domaine*] *projection*.

L'agent de transfert est réalisé par la commande **ypxfr** qui est activée par le démon **ypserv** du serveur esclave sur réception d'un appel à sa procédure de transfert.
Cette commande peut toutefois être lancée à la main pour récupérer une projection donnée. Normalement on réalise cela sur les serveurs esclave en utilisant des shell-scripts lancés par **cron** à intervalles réguliers.

Pour effectuer le transfert de la projection donnée en argument, **ypxfr** appelle une procédure spécifique du démon **ypserv** du serveur maître. Il reconstitue les projections, dans un fichier temporaire, en utilisant les fonctions **DBM**. En cas de réussite les anciennes projections sont écrasées.
La commande **ypinit –s** utilise **ypxfr** pour rapatrier les projections à partir du serveur maître.

Les commandes permettant d'identifier la machine sur laquelle tourne un serveur NIS actif, et aussi d'obtenir le contenu complet ou partiel d'une projection sont :

- **ypwhich** sans options affiche le serveur actif. L'option **–m** affiche la liste des projections et le nom du serveur maître pour celles-ci.

- **ypcat** affiche tout le contenu d'une projection. L'option **–k** permet d'afficher les clés.

- **ypmatch** est un complément de la commande **ypcat**.

4. Clients NIS

Une fois un ou plusieurs serveurs **NIS** activés, les clients **NIS** peuvent à leur tour être installés et lancés. Il est recommandé de ne le faire qu'une fois que les serveurs tournent. Afin de pouvoir utiliser **NIS** sur des machines clientes, il faut s'assurer de trois choses :

- Que les fichiers de configuration comportent le marqueur d'entrées **NIS** (signe '+') de sorte que les informations **NIS** puissent être rajoutées aux informations locales.
- Positionner le nom de domaine **NIS** sur le client dans le fichier **/etc/defaultdomain**.
- Lancer le démon **ypbind**, responsable de la localisation des serveurs **NIS** et du maintien des associations noms de domaines et serveurs. Sur certaines versions des plus récentes, l'adresse ou le nom du serveur maître sont conservées dans le répertoire **/var/yp/binding** afin de faciliter la recherche du serveur auquel s'associer. Il faut s'assurer que le répertoire **/var/yp** existe avant le lancement de **ypbind**.

Lorsque **NIS** est lancé sur un poste client, la référence aux fichiers de configuration est traitée de deux manières :

- La BD **NIS** remplace certains fichiers locaux. (Les copies locales sont ignorées)
- Certains fichiers sont augmentés par **NIS**. Ces fichiers sont consultés avant la formulation de requêtes **NIS**, comme c'est le cas pour **/etc/passwd**, ou **/etc/group**. Si le premier champ, le nom de login ou de groupe, est un '+' alors le contenu entier de la projection du fichier concerné est inséré à cet endroit. Évidemment il n'y a pas insertion physique mais poursuite de la recherche dans la projection au lieu du fichier. Si le premier champ est de la forme **+nom**, l'entrée concernant nom dans la projection est insérée. Enfin si ce champ est de la forme **+@groupe**, toutes les entrées concernées de groupe, dans le fichier **/etc/netgroup** (en fait sa projection) sont examinées, et les entrées retenues sont recherchées dans les projections des fichiers **passwd** et **group**.

Exercice :

Avec les utilitaires UML (User Mode Linux):

- Créer un réseau et lancer 2 machines nommées **serveurNIS** et **clientNIS** dont l' interface **eth0** est connectée à ce réseau.
- Affecter une @IP à l'interface eth0 de chaque machine et vérifier qu'elles communiquent
- Sur la machine **serveurNIS**, créer le groupe **admin**(GID=4000) et une dizaine de comptes utilisateurs dans ce groupe: **agent01** (UID=4001), **agent02** (UID=4002), etc...
- Sur la machine **serveurNIS**, configurer un serveur **maître NIS** pour le domaine **yp.tpNIS.**
- Sur la machine **clientNIS**, configurer un client **NIS** pour le domaine **yp.tpNIS.**
- Vérifier que les utilisateurs crées sur la machine **serveurNIS**, peuvent se connecter sur la machine **clientNIS** bien qu'il n'aient pas d'existence sur cette machine.
- Comme ils n'ont pas de répertoire de connexion sur la machine **clientNIS**, il sont connectés dans le répertoire racine (/). En s'inspirant du TP sur NFS, faire en sorte qu'ils accèdent à leur répertoire de connexion de la machine **serveurNIS**, quand ils se connectent sur la machine **clientNIS**.
- Sur la machine **serveurNIS**, supprimer le groupe **admin** et les utilisateurs **agent*xy*** des fichiers locaux. Configurer un client **NIS** et vérifier que les utilisateurs **agent*xy*** peuvent toujours se connecter.