Module n°3
Services réseaux et Sécurité

Auteur : Laboratoire Unix
0.1 – d octobre yyyy
Nombre de pages : 41

[Frame2]

Table des matières

1.DHCP / DNS

1.1.Présentation

Le DNS (Domain Name System) fournit un mécanisme qui permet de convertir les noms d'hôtes de réseaux ou autres alias de courrier électroniques en adresses IP. Par exemple, le passage de www.google.com en 216.239.51.100. Pour ce faire, il divise le nom en différents groupes logiques. Chaque zone est délimitée par un point : com est le réseau auquel appartient le sous réseau google. www est le nom d'hôte de la machine qui possède le serveur web, dans le sous réseau google.

Le serveur DNS permet bien entendu la résolution inverse qui permet de convertir l'IP en nom.

Nous ne souhaitons pas mettre en place un serveur DNS pour Internet, bien que cela soit possible, il faut payer une adresse IP publique alors que type de service est disponible gratuitement sur le net (www.dyndns.com). La mise en place d'un serveur DNS dans un LAN personnel ou dans celui d'une entreprise permet aux utilisateurs et à vous même de ne plus avoir à taper et à retenir les IP des différentes machines du réseau, les noms sont beaucoup plus pratiques.

1.2.Sécurité

Il est fortement conseillé d'utiliser une version récente de bind (9.x) car de nombreuses failles de sécurités sont corrigées à chaque mise à jour. La sécurité est importante, car mal configuré, le DNS permettrait à quelqu'un de mettre à jour votre serveur pour se faire passer pour une machine de votre réseau.

Pour ces raisons, nous n'allons autoriser l'update du DNS que par le serveur DHCP. Chaque client du domaine fournira son nom d'hôte (hostname) au serveur DHCP lorsqu'il prendra une IP. Le serveur DHCP fera alors une demande d'update du DNS pour ajouter le nouvel enregistrement DNS.

Attention: Les clients windows spécifient automatiquement leurs hostname au DHCP lorsqu'ils prennent une IP. Ce n'est pas le cas de tout les linux, tout dépend de votre distribution, par exemple avec dhcpcd il faut spécifier l'option -h hostname:

```
# dhcpcd -h blabla eth0
```

On passe ici **blabla** comme hostname au serveur DHCP.

Pour lancer - stopper - relancer named et dhcpd, l'utilisation des scripts pour init.d est préférable:

```
# /etc/init.d/dhcpd start

# /etc/init.d/named start
```

Vous pouvez remplacer start par restart ou stop suivant ce que vous voulez faire.

1.3.Définitions

1.3.1.DHCP

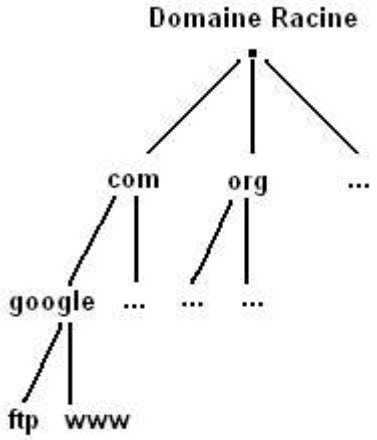
Le démon dhcpd implémente le Dynamic Host Configuration Protocol, qui permet d'attribuer des IP aux hôtes d'un réseau. Les hôtes récupèrent aussi d'autres informations liées à leur réseau comme l'adresse de la passerelle (pour établir la route par défaut), l'adresse du DNS et le nom du domaine.

Pour fonctionner le dhcpd lit le fichier de configuration **dhcpd.conf** qui contient les données nécessaire à la distribution d'adresses. Il se trouve la plupart du temps dans **/etc** mais il peut aussi être dans **/etc/dhcp/**, cela dépend de votre distribution. Ce fichier contiendra des déclarations de sous réseaux sur lesquels ils vont distribuer les IP, les plages d'adresses a réserver, le nom du domaine, l'adresse du routeur (passerelle), l'adresse du serveur DNS. Il est aussi possible de réserver une IP pour une certaine machine grâce à l'adresse mac.

1.3.2.DNS

Les données stockées dans le DNS sont identifiées par noms de domaine qui sont organisées comme un arbre.

Chaque noeud de l'arbre appelé domaine reçoit un label. Le nom de domaine du noeud est la concaténation de tous les labels des noeuds sur le chemin du domaine racine.



Le nom de domaine est séparé en parties appelées zones. Chaque zone part d'un noeud et va vers un noeud où commence une nouvelle zone. Les données pour chaque zone sont stockées dans le serveur de noms, qui répondra aux requêtes d'une zone en utilisant le protocole DNS.

Les données associées à chaque nom de domaine sont enregistrées sous forme de **Resource Record (RR)**.

Il existe quatre types de configuration de serveurs de nom :

- **Maître** - Stocke les enregistrements de zone originaux pour un certain espace de nom et répond aux questions d'autres serveurs de noms qui cherchent des réponses concernant cet espace de nom.
- **Esclave** - Répond aussi aux requêtes d'autres serveurs de noms concernant les espaces de nom pour lesquels il est considéré comme faisant autorité. Les serveurs de noms esclaves reçoivent leurs informations d'espace de noms des serveurs de noms maîtres par l'intermédiaire d'une zone de transfert, dans laquelle l'esclave envoie au maître une requête dite NOTIFY pour une certaine zone. Le maître répond en fournissant les informations, si l'esclave est autorisé à recevoir le transfert.
- **Caching-only** - Offre des services de résolution nom vers IP mais ne fonctionne pas dans n'importe quelle zone. Les réponses pour toutes les résolutions sont en général placées en cache dans une base de données stockée en mémoire pour une période établie, le plus souvent spécifiée par l'enregistrement de zone importé, ce qui permet d'obtenir une résolution plus rapide pour d'autres clients DNS après la première résolution.
- **Forwarding** - Fait suivre des requêtes pour résolution à une liste spécifique de serveurs de noms. Si aucun des serveurs de noms spécifiés ne peut effectuer la résolution, le processus s'arrête et la résolution a échoué.

1.4.Fichiers de configuration

1.4.1.named.conf

Ce fichier contient les informations globales du serveur comme par exemple sur quelle interface le démon va écouter.

Il contient aussi toutes les déclarations de zones. Le nom de la zone est important, puisqu'il constitue la valeur par défaut assignée à la directive **$ORIGIN** utilisée dans le fichier zone.

Par exemple, si cette déclaration zone définit l'espace de nom pour **domain.home**, il faut utiliser **domain.home** en tant que nom de zone pour qu'il soit placé à la fin des noms d'hôtes utilisés dans le fichier de zone.

Avant de pouvoir attaquer la configuration proprement dite, certains mots clés vont nous être utile dans la déclaration des zones:

- **allow-query** : Spécifie les clients qui sont autorisés à requérir des informations à propos de cette zone. Par défaut toutes les requêtes d'informations sont autorisées.

- **allow-transfer** : Spécifie les serveurs esclaves qui sont autorisés à requérir un transfert des informations de la zone. Par défaut toutes les requêtes de transfert sont autorisées.

- **allow-update** : Spécifie les hôtes qui sont autorisés à mettre à jour dynamiquement des informations dans leur zone. Par défaut aucune requête de mise à jour dynamique n'est autorisée (sécurité oblige).

Nous ne permettrons la mise à jour des zones uniquement par le DHCP, c'est à dire uniquement la machine qui dispose du DHCP/DNS.

- **file** : Spécifie le nom du fichier qui contient les données de configuration de la zone dans le répertoire de fonctionnement de named (par défaut **/var/named** ou **/var/bind/**).

- **masters** : Utilisée si la zone est définie comme de type esclave. L'option masters indique au named d'un esclave les adresses IP d'où il est possible de requérir des informations de zone.

- **notify** : Détermine si named envoie une notification aux serveurs esclaves quand une zone est mise à jour. Par défaut le choix est yes (oui), mais vous pouvez régler sur no, pour empêcher que les esclaves en soient

notifiés, ou explicit, pour n'envoyer de notification qu'aux serveurs de la liste also-notify.

- **type** : Définit le type de zone. Les types suivants peuvent être utilisés :

- **forward** : Dit au serveur de noms de faire suivre toutes les requêtes d'informations à propos de cette zone vers d'autres serveurs de noms.

- **hint** : Un type spécial de zone qui est utilisé pour orienter vers les serveurs de noms racines, servant à résoudre des requêtes lorsqu'une zone n'est pas connue par ailleurs. Normalement vous n'aurez pas besoin de configurer une zone d'indication au-delà du **/etc/named.conf** par défaut.

- **master** : Désigne le serveur de noms présent comme faisant autorité pour cette zone. Une zone devrait être configurée comme de type master si vous possédez les fichiers de configuration de la zone sur le présent système.

- **slave** : Désigne le serveur de noms présent comme serveur esclave pour cette zone, disant à named de requérir les fichiers de configuration de la zone depuis l'adresse IP du serveur de noms maître pour cette zone.

Maintenant voyons les mots clés des fichiers de zones que nous avons spécifiés par le mot clé **file**.

1.4.2.Fichiers de zones

Les fichiers de zones, qui contiennent des informations sur un espace de nom particulier, sont stockés dans le répertoire de fonctionnement de named (**/var/named** ou **/var/bind**).

Chaque fichier de zone contient des enregistrements de ressources. Les enregistrements de ressources définissent les paramètres de la zone, assignant une identité à des systèmes à l'intérieur de l'espace de nom de la zone.

Voici les différents champs que nous allons rencontrer dans nos fichiers de zones:

```
@ IN SOA <serveur-noms-primaire> <email-hôte> (
 <numéro-série>
 <temps-actualisation>
 <temps-nouvel essai>
 <temps-expiration>
 <TTL-minimum> )
```

- **SOA** : Enregistrement « Start Of Authority », qui proclame des informations importantes faisant autorité à propos des espaces de nom pour les serveurs de noms.

- Le symbole **@** place le nom de zone en tant qu'espace de nom défini par l'enregistrement de ressources **SOA**. Le serveur de noms primaire autorisé pour ce domaine est utilisé pour **serveurs-noms-primaire** et l'adresse e-mail de la personne à contacter à propos de cet espace de nom est substituée à **email-hôte**. Notez que l'**@** est remplacé par un point.

- **numéro-série** est incrémentée chaque fois que vous changez le fichier de zone afin que named sache qu'il doit recharger cette zone.

- **temps-actualisation** dit à tout serveur esclave combien de temps attendre avant de demander au serveur de noms maître si des changements ont été effectués dans la zone.

- **temps-nouvel** essai informe le serveur de noms esclave de l'intervalle de temps à attendre avant d'émettre une nouvelle requête de rafraîchissement, au cas où le serveur de noms maître ne répondrait pas. Si le serveur maître n'a pas répondu à une requête de rafraîchissement avant que la valeur indiquée dans **temps-expiration**, le serveur esclave cesse de répondre aux requêtes au sujet de cet espace de nom.

- **TTL-minimum** demande que d'autres serveurs de noms placent en cache les informations pour cette zone pendant au moins cette période (en secondes).

Dans BIND, tous les temps sont exprimés en secondes. Toutefois, vous pouvez aussi utiliser des abréviations pour des unités de temps autres que des secondes, comme les minutes (M), les heures (H), les jours (D) et les semaines (W).

Les secondes comparées à d'autres unités de temps:
En secondes - En autres unités de temps (M = Minute, H = Heures, D = Day, W = Week)

```
60 - 1M
```

```
1800 - 30M
3600 - 1H
10800 - 3H
21600 - 6H
43200 - 12H
86400 - 1D
259200 - 3D
604800 - 1W
31536000 - 365D
```

- **NS** : Enregistrement de serveur de noms (NameServer) qui annonce les serveurs de noms faisant autorité pour une zone.

- **MX** : Enregistrement Mail eXchange, qui dit où doit se diriger le courrier envoyé à un nom d'espace particulier contrôlé par cette zone.

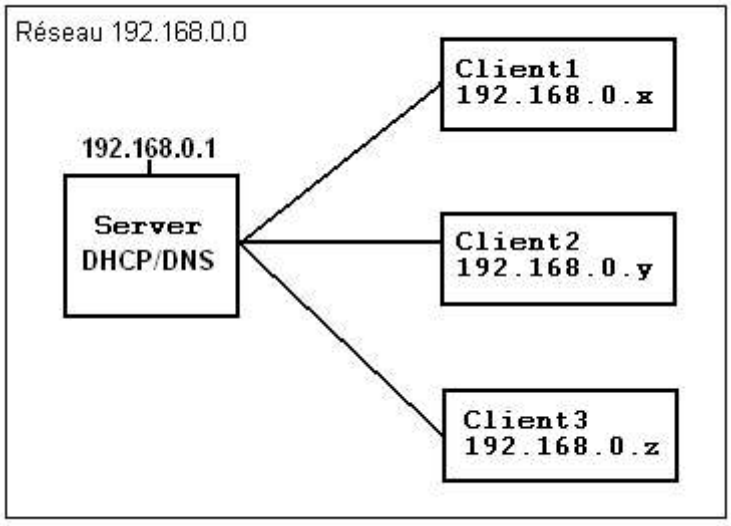
- **A** : Enregistrement d'Adresse qui spécifie une adresse IP à assigner à un nom.

- **PTR** : Enregistrement PoinTeR record, conçu pour orienter vers une autre partie de l'espace de nom. Les enregistrements PTR servent surtout à la résolution inverse des noms, puisqu'ils réorientent les adresses IP vers un nom particulier.

Voila, nous possédons tous les mots-clés nécessaire à la réalisation de notre exemple, passons maintenant à la pratique.

1.5.Un exemple pratique

Dans cet exemple nous disposons d'un réseau 192.168.0.0 dans lequel nous avons 4 machines: Un serveur DHCP/DNS qui à comme nom d'hôte « **server** » et trois clients qui se verront attribuer leur IP sur le réseau par le DHCP, ils fourniront leur nom d'hôte lorsqu'ils prendront une IP et le DHCP fera une demande de mise à jour au DNS.



1.5.1.dhcpd.conf

```
# dhcpd.conf
  ddns-update-style interim;          #Spécifie le type de mise à jour du DNS (Il
existe
                                      #2 modes de mise à jour: le ad-hoc qui est
                                      #maintenant déclaré obsolète et l'interim
que
                                      #nous utilisons

ignore client-updates;                      #On bloque la mise à jour du DNS
directement par
                                      #les clients

log-facility local7;                  ##Optionnel

key "rndc-key"                        #On déclare une clé pour les transactions
entre
{                                     #DHCP et DNS pour plus de sécurité
   algorithm hmac-md5;
   secret "RcGBoblablablabla";        #Cette clé est générée automatiquement par
                                      #certaines distributions lors de
l'installation.
};

zone domain.home. {                         #Nous déclarons ici la zone sur
laquelle le DHCP
                                      #pourra demander une mise à jour
   primary 192.168.0.1;               #on spécifie le DNS primaire
   key rndc-key;                      #et la clé à utilisé pour les transactions
}

zone 0.168.192.in-addr.arpa. {                  #La zone de reverse
   primary 192.168.0.1;
```

```
  key rndc-key;
}

#Ici commence la configuration proprement dite du DHCP

subnet 192.168.0.0 netmask 255.255.255.0 {        #On déclare notre sous
réseau

  authoritative;                                          #Permet de spécifier
que le server est
                                                          #"maître" du sous réseau

  range 192.168.0.10 192.168.0.20;                #La plage d'adresse à
distribuer

  option domain-name-servers 192.168.0.1;                 #L'adresse du DNS

  option domain-name "domain.home";               #Le nom du domaine

  option routers 192.168.0.1;                             #La passerelle par
défaut

  option broadcast-address 192.168.0.255;         #L'adresse de broadcast

  default-lease-time 600;                         #Le temps des baux par
défaut
  max-lease-time 7200;                            #et maximum
}

#host unemachine {                                #Pour réserver une IP en
#  hardware ethernet 00:XX:XX:XX:XX:X;                   #fonction d'une
adresse MAC
#    fixed-address 192.168.0.5;
#    }
```

1.5.2.named.conf

```
#named.conf

options {
        directory "/var/bind";                  #Le répertoire d'exécution de bind
        pid-file "/var/run/named/named.pid";
};

key "rndc-key"
{
        algorithm hmac-md5;
        secret "RcGBoeTblablabla";
};
```

```
zone "." IN {                                   #Cette zone est définie par défaut,
elle
                                                #permet la résolution vers les
adresses
                                                #internet
        type hint;
        file "named.ca";                                #Ce fichier est déjà créé par
défaut
};

zone "localhost" IN {                           #La zone pour résoudre
localhost
        type master;
        file "pri/localhost";
        allow-update { none; };                 #Cette zone n'a pas a être
modifiée
        notify no;
};

zone "127.in-addr.arpa" IN {                    #La zone de reverse du
localhost
        type master;
        file "pri/127";
        allow-update { none; };
        notify no;
};

zone "domain.home"                              #La zone de notre domaine
{
        type master;
        file "pri/domain.home";
allow-update { 192.168.0.1;key rndc-key; };#On autorise la mise à jour par le
serveur
                                                #lui même avec la clé
};

zone "0.168.192.in-addr.arpa"                   #La zone de reverse de notre
domaine
{
        type master;
        file "pri/192.168.0";
allow-update { 192.168.0.1;key rndc-key; };
};
```

Le mot clé IN signifie INternet et est utilisé dans 99% des cas.

1.5.3.Fichiers de zone

Les fichiers **localhost** et **127** sont générés lors de l'installation, la configuration par défaut nous convient.

Attention **:** Les noms de machines doivent se terminer par un point ; par exemple, **« server.domain.home. »** désigne **« server.domain.home »** alors que **« server.domain.home »** désigne **« server.domain.maison.domain.maison »**.

- **fichier localhost:**

```
$TTL 1W
@       IN      SOA     ns.localhost. root.localhost.  (
                              2002081601 ; Serial
                              28800      ; Refresh
                              14400      ; Retry
                              604800     ; Expire - 1 week
                              86400 )    ; Minimum

                    IN      NS      server.domain.home.
```

- **fichier 127 (qui représente la zone inverse de localhost):**

```
$TTL 1W
@                   1D IN SOA       localhost. root.localhost. (
                                2002081601      ; serial
                                3H              ; refresh
                                15M             ; retry
                                1W              ; expiry
                                1D )            ; minimum

                    1D IN NS        localhost.
*                   1D IN PTR       localhost.
```

- **fichier domain.home:**

```
$ORIGIN .
$TTL 86400      ; 1 jour
domain.home             IN SOA  server.domain.home. email.domain.home. (
                                2001044761 ; Numéro de série
                                86400      ; refresh (1 jour)
                                21600      ; retry (6 heures)
                                3600000    ; expire (5 semaines 6 jours 16 heures)
                                3600       ; minimum (1 heure)
                                )
                        NS      server.domain.home.                     #notre serveur
de noms

server.domain.home.             A       192.168.0.1             #l'adresse de notre
serveur
```

- **fichier 192.168.0 (la zone reverse de domain.home):**

```
$ORIGIN .
$TTL 86400      ; 1 jour
0.168.192.in-addr.arpa  IN SOA  server.domain.home. email.domain.home. (
                        2001044074 ; serial
                        28800      ; refresh (8 heures)
                        14400      ; retry (4 heures)
                        3600000    ; expire (5 semaines 6 jours 16 heures)
                        86400      ; minimum (1 jour)
                        )
                    NS      server.domain.home.
$ORIGIN 0.168.192.in-addr.arpa.
$TTL 3600       ; 1 hour
1                   PTR     server.domain.home.
```

1.5.4.Test de fonctionnement

Tout d'abord vérifiez que les démons tournent en faisant par exemple un

```
# ps ax
```

Si ce n'est pas le cas, cela est sûrement du à une erreur dans un des fichiers de configuration.

Editez les fichiers **/var/log/daemons.log** ou **/var/log/messages** pour voir ce qui ne va pas.

Une fois que les démons tournent, mettez les clients en DHCP puis montez l'interface réseau.

Toujours dans **/var/log/daemons.log** vous devez obtenir une ligne du type:

```
server named[4101]: client 192.168.0.1#1025: updating zone
'domain.home/IN': adding an RR
```

Pour vérifier notre domaine, utilisons l'utilitaire dig:

```
# dig domain.home
; <<>> DiG 9.2.2 <<>> domain.home
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62723
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1,
ADDITIONAL: 0

;; QUESTION SECTION:
```

```
;domain.home.                  IN      A

;; AUTHORITY SECTION:
domain.home.            3600    IN      SOA     server.domain.home.
email.domain.home.
 2001044761 86400 21600 3600000 3600

;; Query time: 3 msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
;; WHEN: Thu Apr 10 15:33:04 2003
;; MSG SIZE  rcvd: 77
```

Le serveur fait donc bien autorité pour la zone domain.home.

Un ping du nom d'hôte « client2 » depuis la machine client1 permet d'être certain du bon fonctionnement du DNS dynamique.

1.5.5.Liens

Pour plus d'informations vous pouvez consulter le site d'ISC qui fourni BIND et le DHCP, ainsi qu'un guide de référence pour BIND 9: **http://www.isc.org/**

Les RFC qui concernent le protocole DNS: 1033, 1034, 1035 et 2136.

2.SAMBA

2.1.Pré requis

Théoriques

- Connaissances de bases en TCP/IP (Fin Prépa1)
- Connaissances de bases en Linux et Windows (Fin Prépa 1)

Pratiques

- Des ordinateurs reliés entre eux au sein d'un réseau TCP/IP fonctionnel.
- Une distribution contenant les packages SAMBA, ou, à défaut Linux + les packages SAMBA.

2.2.Un peu d'histoire

Samba a été créé par Andrew Tridgell (actuellement chef de file de l'équipe de développement du projet Samba) en 1991. Il travaillait sur le développement d'un programme de gestion de fichiers basé sur le protocole propriétaire SMB (Server Message Block) implémenté par Microsoft et IBM. Le nom de Samba à été trouvé grâce à la commande UNIX suivante:

```
grep -i 's.*m.*b' /usr/dict/words

↙ résultat: salmonberry samba sawtimber scramble
```

En d'autre termes SAMBA est l'implémentation libre (sous liscence Gnu/Gpl) du protocole de communication SMB. Concrètement samba vous permet de :

- Partager des répertoires
- Partager des systèmes de fichiers distribués
- Partager une imprimante sur le serveur Unix avec des clients Unix/Windows
- Jouer le rôle de contrôleur de domaine 2000/NT.
- Fourni un serveur WINS (Windows Internet Name Service)

Tout le travail a été fait en Reverse Engineering, c'est à dire en observant le fonctionnement du serveur sous Windows, en analysant les transactions sur le réseau, en analysant les dumps des exécutables Windows.

Voici une citation extraite d'un article sur les débuts du projet Samba:

« Several megabytes of NT-security archives, random whitepapers, RFCs, the CIFS spec, the Samba stuff, a few MS knowledge-base articles, strings extracted from binaries, and packet dumps have been dutifully waded through during the information-gathering stages of this project, and there are *still* many missing pieces... While often tedious, at least the way has been generously littered with occurrences of clapping hand to forehead and muttering "crikey, what are they thinking?" »

--Hobbit, CIFS: Common Insecurities Fail Scrutiny

Depuis peu (version 3.0), il supporte les authentifications via un PDC Windows 2000/2003 avec Active Directory ou même un annuaire LDAP (OpenLDAP).

2.3.Récupérer des packages SAMBA.

Vérifiez si les packages SAMBA sont bien installés sous RedHat :

```
# rpm -qa | grep samba
samba-2.2.3a-6
samba-common-2.2.3a-6
samba-client-2.2.3a-6
```

Si vous obtenez quelque chose de similaire, c'est que tout est déjà prêt, passez à la section suivante.
Sinon il faut installer les packages. Selon la distribution que vous utilisez la procédure diffère légèrement. Sous RedHat il vous faudra récupérer les rpm sur le site de samba (www.samba.org) ou utiliser l'utilitaire apt disponible sur le site www.freshrpms.net :

```
# apt-get install samba
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
  samba-common
The following NEW packages will be installed:
  samba samba-common
0 packages upgraded, 2 newly installed, 0 removed and 78
not upgraded.
Need to get 4996kB of archives.
After unpacking 14.7MB of additional disk space will be
used.
Do you want to continue? [Y/n]
Get:1  http://ayo.freshrpms.net  redhat/7.3/i386/updates
samba-common 2.2.7-3.7.3 [2419kB]
Get:2  http://ayo.freshrpms.net  redhat/7.3/i386/updates
samba 2.2.7-3.7.3 [2577kB]
Fetched 3231kB in 18s (177kB/s)
Executing RPM (-Uvh)...
Preparing...
########################################### [100%]
                                              1:samba-common
########################################### [ 50%]
    2:samba
########################################### [100%]
```

Sous Gentoo pour vérifier si samba est installé :

```
emerge -s samba
```

Si le champs Latest Version Installed renvoie « not installed », le programme n'est pas installé.
Pour installer samba :

```
emerge samba
```

2.4.Les démons de samba

La suite SAMBA est composé d'une paire de 'demon' UNIX qui fournissent des ressources partagées :

smbd

Smbd est un 'demon' qui prend en charge le partage de fichiers, d'imprimantes et l'authentification des clients SMB.

nmbd

Nmbd est un 'demon' qui supporte les services NetBIOS et WINS (l'implémentation Microsoft de NBNS : NetBios Name Server).

Pour démarrer ces deux démons on pourra utiliser la fonction 'service' sous RedHat :

```
service smb start
```

Ou plus généralement :

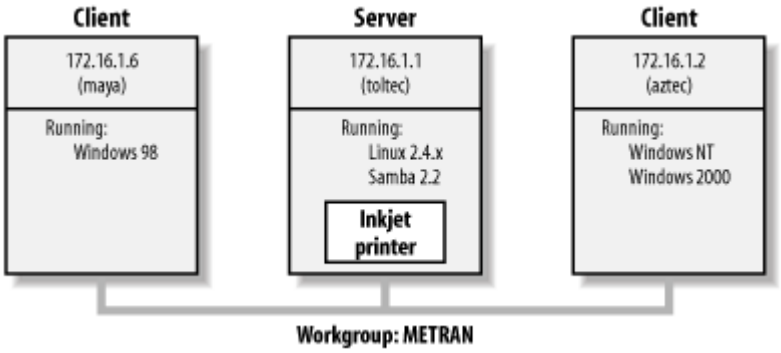
```
# /etc/init.d/smb start
Starting                   SMB                    services:
[  OK  ]
Starting                   NMB                    services:
[  OK  ]
```

Les deux démons nécessaires au fonctionnement de SAMBA viennent de se lancer.
Pour vérifier leur état :

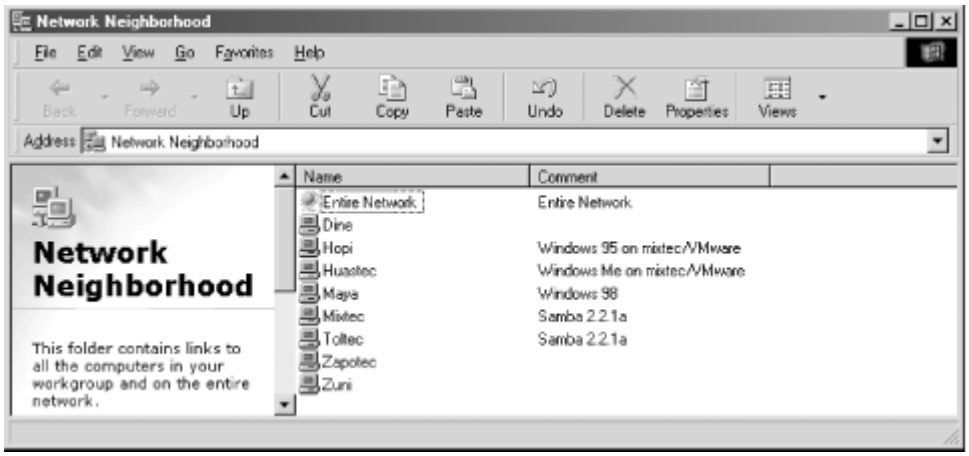
```
# /etc/init.d/smb status
smbd (pid 1504) is running...
nmbd (pid 1509) is running...
```

2.5. Une configuration réseau type, propice à l'utilisation de Samba

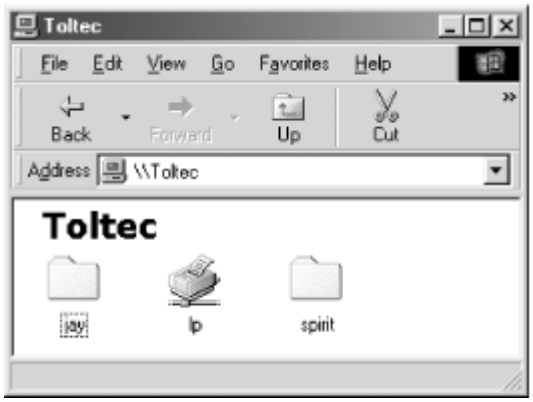
Le cas classique regroupe des clients windows NT/2000/98 et un serveur Unix . On considère que l'imprimante est reliée au serveur Unix et que les clients Windows souhaitent accéder à tout type de ressource partagée (partage de fichiers, imprimantes etc..) :



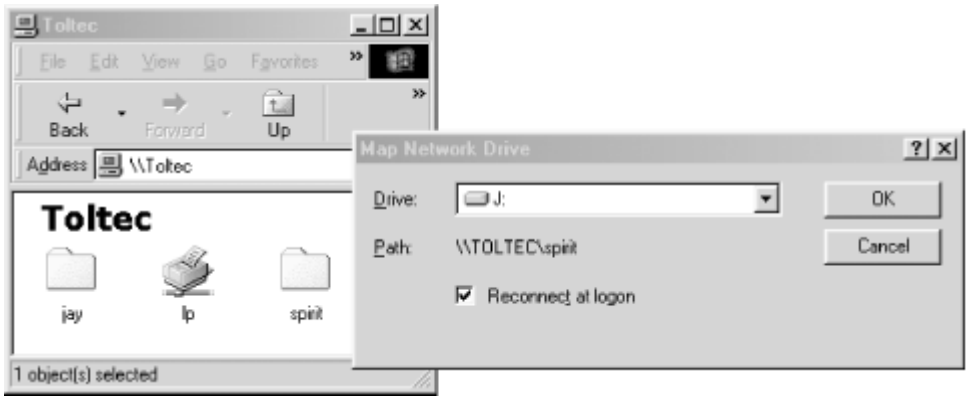
Si tout est bien configuré nous pourrons obtenir un affichage semblable à celui-ci :



Nous voyons que toltec (le serveur samba) est bien présent dans l'explorateur de proximité d'une machine windows sur le réseau (par exemple maya). Voyons ce qui ce passe si l'on double clique sur toltec :



Tous les partages sont bien reconnus. Ici Jay est un répertoire partagé, lp, une imprimante partagée et spirit un système de fichier partagé. On peut sans aucun problème sous Windows mappé un de ces partages à un lecteur réseau :



A partir du moment ou un lecteur distant est créé, on pourra sans aucun problème installer des programmes sur le lecteur distant et les exécuter comme en locale.

L'installation de l'imprimante réseau devrait également se passer sans difficulté :

Du côté du serveur Unix la configuration des deux 'demon' smbd et nmbd passe par l'édition de l'unique fichier smb.conf. Pour afficher des statistiques sur l'état des démons on pourra utiliser la commande smbstatus :

```
 # smbstatus
 Processing section "[homes]"
 Processing section "[printers]"
 Processing section "[spirit]"

 Samba version 2.2.6
 Service      uid    gid    pid     machine
 ----------------------------------------
 spirit       jay    jay    7735    maya     (172.16.1.6)
Sun Aug 12 12:17:14 2002
 spirit       jay    jay    7779    aztec    (172.16.1.2)
Sun Aug 12 12:49:11 2002
 jay          jay    jay    7735    maya     (172.16.1.6)
Sun Aug 12 12:56:19 2002

 Locked files:
 Pid    DenyMode   R/W        Oplock     Name
 --------------------------------------------------
 7735   DENY_WRITE RDONLY     NONE       /u/RegClean.exe
Sun Aug 12 13:01:22 2002

 Share mode memory usage (bytes):
    1048368(99%) free + 136(0%) used + 72(0%) overhead =
1048576(100%) total
```

Ce résultat n'est la qu'un titre indicatif, nous expliquerons cela plus en détail par la suite. Remarquez juste que les informations sont classées par catégories et nous fournissent des informations sur, les noms des partages, les noms des entités qui utilisent les différents partages et sur l'état du serveur.

2.6.Le protocole NetBIOS

La communication réseau via le protocole SMB et assez différente d'une communication classique TCP/IP utilisée par des protocoles comme FTP ou Telnet. Nous allons commencer par voir les concepts de base du protocole SMB puis nous verrons quelques implémentations que Microsoft en a fait, enfin on terminera par exposer les situations qui sont ou non propices à l'utilisation du serveur Samba.

2.6.1.Comprendre NetBIOS

En 1984, IBM créa une simple API (Application Programming Interface) pour mettre en réseau ses ordinateurs : *Network Basic Input/Output System* (NetBIOS). L'API Netbios permet aux applications de se connecter et de partager des données entre elles de manière rudimentaire.

Considérez l'API NetBIOS comme une extension de l'API BIOS qui contient du code de bas niveau permettant d'effectuer les opérations sur le système de fichier local. Au début NetBIOS à été mis en œuvre sur les réseaux de type TokenRing, puis sur des réseaux IPX de Novell.

Le protocole TCP utilise des nombres pour représenter les adresses des machines (192.168.0.2) tandis que NetBIOS n'utilise que des noms. Certains problèmes ce sont posés lorsque l'on essaya de faire cohabiter les deux protocoles, c'est à cet effet que les RFC 1001 et 1002 (documents de standardisation) furent publiés et décrivent comment NetBIOS doit fonctionner sur un réseau de type TCP/UDP. Ce protocole est communément appelé NBT (NB over Tcp).

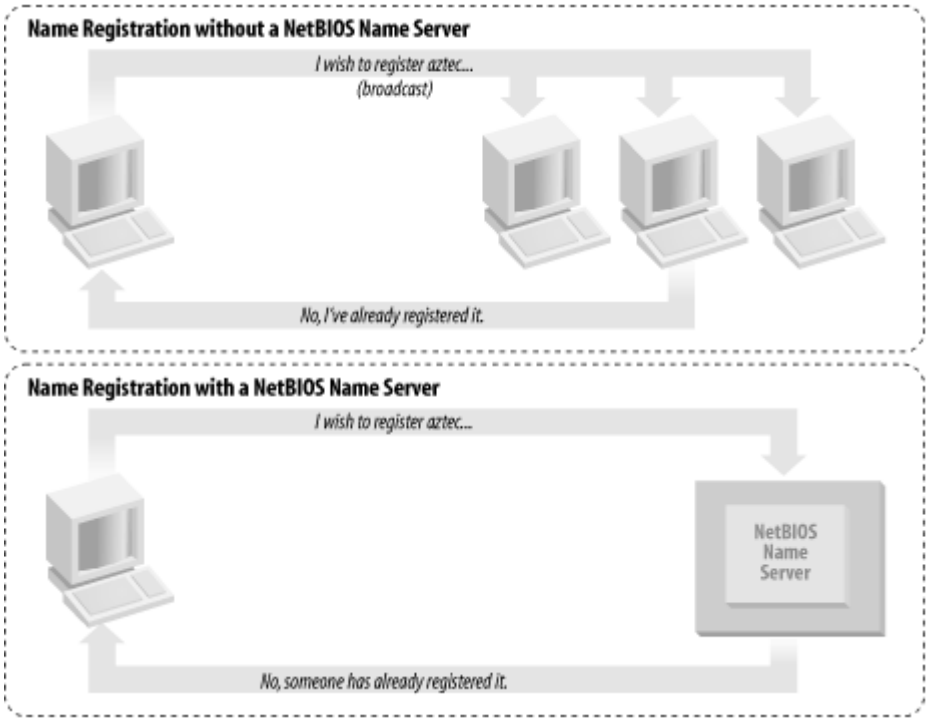
Ces documents continuent aujourd'hui à régir toutes les implémentations y compris celles de Microsoft ou même Samba.

2.6.2.Récupérer un nom NetBIOS

Dans le monde de NetBIOS, lorsqu'un ordinateur se connecte, il effectue une requête d'enregistrement de son nom NetBIOS. Cependant il est impossible que deux ordinateurs du même groupe de travail aient le même nom. Deux approches permettent de s'assurer que ce cas ne se produise jamais :

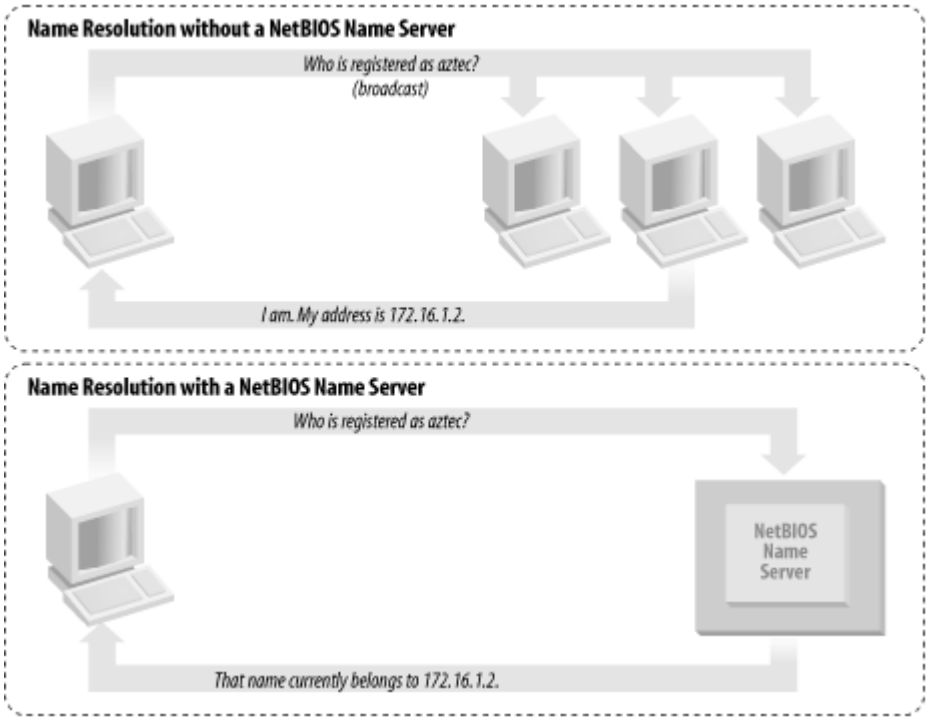
- Utiliser un NBNS pour garder une trace de toutes les machines qui ont enregistré un nom NetBIOS sur le réseau
- Autoriser chaque ordinateur du réseau à défendre son enregistrement dans le cas ou un autre ordinateur essaye d'enregistrer le même nom.

Voici l'illustration de l'echec d'une requête d'enregistrement avec ou sans NBNS :



Comme nous l'avons vu avant, il doit y avoir un moyen de résoudre un nom NetBIOS à partir d'une adresse IP. Les deux différentes approches possibles sont :

- Chaque ordinateur renvoie son, IP lorsqu'il 'entend' une requête de type broadcast de son nom NetBIOS.
- Utiliser un NBNS pour résoudre les noms NetBIOS vers des adresses IP.



Il semble logique d'affirmer que la mise en place d'un NBNS permet de grandement accroître les performances du réseau en évitant un déluge de broadcast (communication en Point to Point) inutile pour le simple enregistrement de nom.

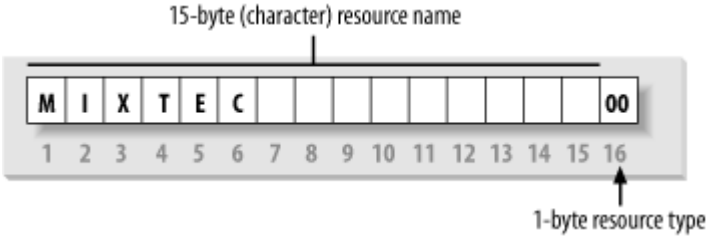
Pour savoir comment une machine windows va réagir pour l'enregistrement et la résolution de nom on pourra tapper ipconfig /all dans un terminal MS-DOS et rechercher la ligne contenant Node-Type.
Voici les valeurs que vous pourrez trouver :

b-node	Utilise la méthode de broadcast pour l'enregistrement et la résolution.
p-node	Utilise la méthode d'enregistrement et de résolution en Point-to-Point (NBNS)
m-node (mixed)	Utilise le broadcast pour la résolution et l'enregistrement puis informe un serbeur NBNS de ses résultats
h-node (hybrid)	Utilise un NBNS s'il existe, dans le cas contraire il effectue un broadcast.

2.6.3.

2.6.4.Type et nom des ressources

Le protocole NetBIOS permet non seulement à une machine de s'identifier au sein d'un réseau par son nom mais aussi d'expliciter le type de service qu'elle peut éventuellement fournir. Par exemple la station mixtec (de notre configuration type) peut indiquer qu'elle n'est pas juste une station de travail mais est aussi un serveur de fichiers et peut recevoir des message de Windows Messenger. Ces informations peuvent être fournies en ajoutant un 16ème byte à la fin du nom de machine (qui ne peut contenir que 15 caractères).



Pour connaître toutes les informations disponible sur un ordinateur utilisant le protocole NetBIOS sur le réseau on pourra saisir :

```
C:\nbtstat -a <machine>

Par example :
C:\>nbtstat -a toltec

        NetBIOS Remote Machine Name Table
   Name               Type         Status
---------------------------------------------
TOLTEC          <00>  UNIQUE      Registered
TOLTEC          <03>  UNIQUE      Registered
TOLTEC          <20>  UNIQUE      Registered
...
```

Voici la signification de la plupart des codes ressources :

Nom de la ressource	Valeur Hexadécimale du 16ème byte
Standard Workstation Service	00
Messenger Service	03
RAS Server Service	06
Domain Master Browser Service (associated with primary domain controller)	1B
Master Browser name	1D
NetDDE Service	1F
Fileserver (including printer server)	20
RAS Client Service	21
Network Monitor Agent	BE
Network Monitor Utility	BF

2.7.Introduction au protocole SMB

Nous allons dans cette section décrire plus en détail le protocole SMB lui-même. D'un point de vue 'haut niveau' le protocole SMB est relativement simple. Il inclus un jeu de commandes nécessaire aux diverses opération relatives aux fichiers et aux imprimantes partagées. Par exemple on pourra :

- Ouvrir et fermer des fichiers
- Créer et supprimer des fichiers et des répertoires
- Lire et écrire des fichiers
- Chercher des fichiers
- Gérer les files d'impressions

Chaque opération peut être encodé en un message SMB et transmis depuis et vers un serveur. Ces commandes sont au format Server Message Blocks.

2.7.1.Le format SMB

Le protocole SMB est dans 99% du temps un protocole de requête/réponse. En effet, dans la majeure partie du temps, un client envoie une requête SMB au serveur et celui-ci renvoi une réponse SMB au client.
Un message SMB n'est pas si complexe. Regardons sa structure interne d'un peu plus près. On peut la séparer en deux grandes partie : une entête de taille fixe et une chaîne de commande fixe de longueur très variable en fonction du message.
Voici les différents champs d'une entête de message SMB :

Field	Size (bytes)	Description
0xFF 'SMB'	1	Protocol identifier
COM	1	Command code, from 0x00 to 0xFF
RCLS	1	Error class
REH	1	Reserved
ERR	2	Error code
REB	1	Reserved
RES	14	Reserved
TID	2	TID; a unique ID for a resource in use by the client
PID	2	Caller process ID
UID	2	User identifier
MID	2	Multiplex identifier; used to route requests inside a process

Le champ COM identifie la commande à exécuter. Toutes les commandes ne doivent ne remplissent pas tous les champs à chaque requête. Prenons par exemple la première demande de connexion d'un client vers un serveur, le client ne pourra fournir un identificateur d'arborescence (TID), la valeur de ce champ sera alors NULL. D'autres champs peuvent être égal à 0 lorsqu'ils ne sont pas utilisés.

Juste après l'entête se trouve un nombre variable de bytes qui constitue une commande ou une réponse SMB. Chaque commande comme Open File (COM=SMBopen) ou Get Print (COM=SMBsplretq) possède son propre jeu de paramètres et de données. Comme pour l'entête tous les champs n'ont pas forcément à être remplis.

Field	Size (bytes)	Description
WCT	1	Word count
VWV	Variable	Parameter words (size given by WCT)
BCC	2	Parameter byte count
DATA	Variable	Data (size given by BCC)

Pour plus d'informations sur chaque commande SMB référez vous à *CIFS Technical Reference* : http://www.snia.org/tech_activities/CIFS.

De part son évolution le jeu de commandes SMB se trouve de plus en plus enrichi. Cependant il garde une compatibilité ascendante, ainsi des entités utilisant différentes versions du protocole pourrons communiquer entre elles.
Voici à titre indicatif l'ensemble des 'dialectes' du protocole SMB

Protocol name	ID string	Used by
Core	PC NETWORK PROGRAM 1.0	
Core Plus	MICROSOFT NETWORKS 1.03	
LAN Manager 1.0	LANMAN1.0	
LAN Manager 2.0	LM1.2X002	
LAN Manager 2.1	LANMAN2.1	
NT LAN Manager 1.0	NT LM 0.12	Windows NT 4.0
Samba's NT LM 0.12	Samba	Samba
Common Internet File System	CIFS 1.0	Windows 2000/XP

2.7.2.Une connection SMB simple

Le client et le serveur doivent accomplir trois étapes afin d'établir une connexion à une ressource.

- Etablir une session NetBIOS
- Déterminer le dialecte du protocole à employer
- Définir les paramètres de session et établir la connexion à la ressource.

Chaque étape peut être observée grâce au programme d'écoute réseau tcpdump légèrement modifié et disponible sur le site de Samba dans le répertoire `samba/ftp/tcpdump-smb`.

Le programme s'utilise comme d'habitude, pensez juste à ajouter l'option –s 1500 pour vous assurez de récupérer l'intégralité du paquet et non pas quelques bytes inutilisables.

Voici plus en détail chaque étape scrutée avec notre tcpdump :

Etablissement de la session NetBIOS

Lorsqu'un utilisateur fait une requête d'accès à un disque réseau ou envoie une impression dans la file d'attente d'une imprimante partagée, NetBIOS se charge de la connectivité au niveau de la couche session du modèle OSI. Le résultat est un canal bidirectionnel entre le client et le serveur. Deux messages suffisent à l'établissement d'une connexion entre les deux entités :

1. Le client envoie une requête d'ouverture de session, voici le rapport de tcpdump :

```
>>> NBT Packet
NBT Session Request
Flags=0x81000044
Destination=TOLTEC      NameType=0x20 (Server)
Source=MAYA             NameType=0x00 (Workstation)
```

2. Le serveur répond en accordant une session au client :

```
>>> NBT Packet
NBT Session Granted
Flags=0x82000000
```

A partir de là un canal de communication est ouvert entre le client et le serveur.

Détermination du dialecte à employer et établissement de la connexion

Le client envoie maintenant un message au serveur afin de négocier le dialecte du protocole SMB à employer. La commande encapsulée dans le message est SMBnegprot, le client envoie au serveur la liste des dialectes qu'il comprend et vice versa :

1. Envoi de la requête client :

```
>>> NBT Packet
NBT Session Packet
Flags=0x0
Length=154

SMB PACKET: SMBnegprot (REQUEST)
SMB Command   =  0x72
Error class   =  0x0
Error code    =  0
Flags1        =  0x0
Flags2        =  0x0
Tree ID       =  0
Proc ID       =  5315
UID           =  0
MID           =  257
Word Count    =  0
Dialect=PC NETWORK PROGRAM 1.0
Dialect=MICROSOFT NETWORKS 3.0
Dialect=DOS LM1.2X002
Dialect=DOS LANMAN2.1
Dialect=Windows for Workgroups 3.1a
Dialect=NT LM 0.12
```

2. Réponse du serveur :

Le serveur répond en fournissant l'index de la valeur supportée (0xFF si aucun protocole n'est valide).

Attention : l'indexation commence à 0 !

```
>>> NBT Packet
NBT Session Packet
Flags=0x0
Length=84

SMB PACKET: SMBnegprot (REPLY)
SMB Command   =  0x72
```

```
Error class    =  0x0
Error code     =  0
Flags1         =  0x80
Flags2         =  0x1
Tree ID        =  0
Proc ID        =  5315
UID            =  0
MID            =  257
Word Count     =  17
NT1 Protocol
DialectIndex=5
[...]
```

Dans cet example le serveur choisi donc le dialect NT LM 0.12 qui d'après notre tableau précedent correspond au standard Windows NT 4.0.

3. Définition des paramètres de session et établissement de la connexion à la ressource

Enfin pendant la dernière étape, les paramètres de session et de connexion sont échangés grâce à la commande SMBSesssetupX.
Parmis les paramètres on trouve :

- Le nom du compte et le mot de passe (quand il y en à un)
- Le nom du groupe de travail (workgroup)
- La taille maximum des données qui peuvent être transférées
- Le nombre de requêtes en attente dans la file

Le résultat de tcpdump est le suivant :

```
>>> NBT Packet
NBT Session Packet
Flags=0x0
Length=150

SMB PACKET: SMBsesssetupX (REQUEST)
SMB Command    =  0x73
Error class    =  0x0
Error code     =  0
Flags1         =  0x10
Flags2         =  0x0
Tree ID        =  0
Proc ID        =  5315
UID            =  1
MID            =  257
Word Count     =  13
Com2=0x75
Res1=0x0
Off2=120
MaxBuffer=2920
MaxMpx=50
VcNumber=0
SessionKey=0x1380
CaseInsensitivePasswordLength=24
CaseSensitivePasswordLength=0
Res=0x0
Capabilities=0x1
Pass1&Pass2&Account&Domain&OS&LanMan=
  JAY METRAN Windows 4.0 Windows 4.0

SMB PACKET: SMBtconX (REQUEST) (CHAINED)
smbvwv[]=
Com2=0xFF
Off2=0
Flags=0x2
PassLen=1
```

```
Passwd&Path&Device=
smb_bcc=23
smb_buf[]=\\TOLTEC\SPIRIT
```

Il n'est pas nécessaire de comprendre l'intégralité du résultat, on remarquera juste que l'hexadécimal contenu dans le champs Com2 de la commande SMBsesssetupX représente le code de la commande SMBTconX, qui elle-même recherche le nom de la ressource dans la variable smb_buf. Dans cet exemple cette variable contient la chaîne `\\TOLTEC\SPIRIT` qui est le chemin absolu d'un partage situé sur toltec. Le TID (Tree ID ou identifiant d'arborescence) est toujours égal à 0 à ce moment de la transaction.
Enfin le serveur envoie un TID au client lui indiquant que l'utilisateur à bien été autorisé à accéder à la ressource désirée, et qu'il peut l'utiliser :

```
>>> NBT Packet
NBT Session Packet
Flags=0x0
Length=85

SMB PACKET: SMBsesssetupX (REPLY)
SMB Command   =  0x73
Error class   =  0x0
Error code    =  0
Flags1        =  0x80
Flags2        =  0x1
Tree ID       =  1
Proc ID       =  5315
UID           =  100
MID           =  257
Word Count    =  3
Com2=0x75
Off2=68
Action=0x1
[000] Unix Samba 2.2.6
[010] METRAN

SMB PACKET: SMBtconX (REPLY) (CHAINED)
smbvwv[]=
Com2=0xFF
Off2=0
smbbuf[]=
ServiceType=A:
```

Le champs ServiceType contient le champ « A » ce qui indique qu'il s'agit d'un partage de fichier.
Les champs possibles sont :

- "A" fiichier ou disque
- "LPT1" sortie en file (impression)
- "COMM" modem ou imprimante directement connectés
- "IPC" tube nommé

A présent qu'un TID à été assigné, le client peut utiliser la ressource comme il l'aurait fait localement.

2.8.Les nouveautés depuis Samba 2.2 ?

Dans la version 2.2, Samba possède un support de réseau Microsoft plus avancé permettant une meilleure intégration au sein d'un domaine Windows NT. En plus de cela Samba 2.2 supporte quelques technologies introduites dans Windows 2000.

- **PDC pour clients de type Windows 2000/XP**

Jusque là Samba pouvais servir de PDC (Primary Domain Controler) à un ensemble de machines Windows 95/98/Me et NT, à partie de la 2.2, le serveur Samba peut également agir comme PDC au sein d'un réseau Windows 2000/Xp.

- **Support du Microsoft Dfs**

Microsoft Dfs permet de regrouper aux yeux des utilisateurs un ensemble de ressources situées sur des serveurs différents dans un même répertoire sur un serveur. Ceci simplifie beaucoup la vie des utilisateurs. Ce protocole est supporté à partie de Samba 2.2 .

- **Support d'impression Windows NT/2000/XP**

Support du protocole d'impression RPC.

- **ACLs**

Samba peut comprendre et donc traduire en conséquence les ACL UNIX et Windows Nt/2000/Xp.

- **Intégration de Winbind**

Winbind permet aux utilisateurs dont les login sont stockés sur un serveur Windows de s'authentifier sur un système Unix. Ceci permet un uniformisation de l'environnement de logon qui est bien plus facilement administrable que deux systèmes que l'on doit régulièrement synchroniser.

- **Extensions CIFS Unix**

Les extensions CIFS d'UNIX permettent à samba de supporter le système d'attributs de fichiers Unix et ainsi de servir de remplacement à un serveur NFS. L'avantage étant que l'authentification se fait par utilisateur et non pas par Ip, ce qui garanti une meilleure sécurité..

2.9.Quoi de neuf dans Samba 3.0?

La principale caractéristique de Samba 3.0 est le support de l'authentification Kerberos 5 et LDAP, ce qui est requis pour agir en tant que client dans un environnement ActiveDirectory. On notera également le support de l'Unicode qui simplifie de manière significative l'internationalisation des langues.

Voici en guise de tableau récapitulatif l'ensemble des fonctionnalités fournies par Samba à ce jour :

Rôle	Prise en charge ?
File server	Yes
Printer server	Yes
Microsoft Dfs server	Yes
Primary domain controller	Yes

Backup domain controller	No
Active Directory domain controller	No
Windows 95/98/Me authentication	Yes
Windows NT/2000/XP authentication	Yes
Local master browser	Yes
Local backup browser	Yes
Domain master browser	Yes
Primary WINS server	Yes
Secondary WINS server	No

2.10.Le jeu de commandes samba (non exhaustif)

nmbd
Serveur de nom simple qui joue le rôle d'un serveur WINS. Ce 'demon ' fourni en plus une liste accessible dans la section « Voisinage Réseau » de Windows.

smbd
Gère les ressources partagées entre le serveur Samba et ses clients. Il offre des services de partage d'imprimantes et de fichiers à des clients SMB sur un ou plusieurs réseaux. Il est en plus responsable de l'authentification des utilisateurs.

winbindd
Apparu à partir de Samba 2.2, ce 'demon' permet d'authoriser des utilisateurs dont le compte est stocké dans une base Windows NT/2000.

La distribution Samba contient un jeu de petites commandes Unix :

findsmb
Un programme qui recherche des ordinateurs du réseau local qui répondent au protocole SMB.

net
Un nouveau programme distribué avec Samba 3.0 qui peut être utilisé pour effectuer de l'administration à distance sur un serveur.

nmblookup
Un programme qui permet de trouver l'IP d'un ordinateur dont on connais le nom NetBIOS.

pdbedit

Un nouveau programme fourni dans Samba 3.0 facilitant la gestion des comptes contenus dans des bases de données SAM.

smbcacls
Un programme permettant de définir ou scruter les ACLs sur un système de fichier Windows NT

smbclient
Un client Unix ftp-like qui peut être utilisé pour se connecter à un partage SMB et d'agir dessus.

smbgroupedit
Une commande permettant de mapper des groupes NT avec des groupes UNIX (depuis Samba 3.0)

smbmount/smbumount
Commandes permettant de monter/démonter un partage SMB distant sur un système de fichier local.

smbpasswd
Programme permettant à l'administrateur de changer le mot de passe utilisé par Samba.

smbsh
Un outil qui fonctionne comme un shell et qui permet l'acces a un système de fichier SMB distant grâce aux commandes usuelles unix.

smbspool
Permet d'envoyer des travaux d'impression à une imprimante partagée.

testparm
Ce programme vous permet de tester votre fichier de configuration Samba.

wbinfo
Utilitaire permettant d'interroger le 'demon' winbindd.

2.11.Démarrage

2.11.1.En utilisant init Sys V

Pour lancer SAMBA à chaque démarrage de Linux, un script de démarrage a été créé dans /etc/rc.d/init.d. Des liens vers ce script ont été placés dans les dossiers / etc/rc.d/rc*.d, afin de démarrer ou d'arrêter SAMBA en fonction du runlevel.

SAMBA **ne doit pas** fonctionner dans les "runlevels" 0, 1, 2 et 6 (en effet dans ces "runlevels" le réseau n'est pas activé).
SAMBA **doit** fonctionner dans les "runlevels" 3, 4 et 5.

```
 # chkconfig --list smb
 smb             0:off   1:off   2:off   3:on    4:on
5:on    6:off
```

Pour modifier l'état du service (démarré ou arrêté) dans les différent runlevels :

```
 # chkconfig --level 345 smb on
```

Vous pouvez aussi effacer et créer les liens manuellement :

```
# cd /etc/rc3.d
# ln -s ../init.d/smb S91smb
```

2.11.2.En utilisant inetd

Editez le fichier /etc/service et vérifiez qu'il contient les lignes :

```
netbios-ns 137/tcp     # NETBIOS Name Service
netbios-ns 137/udp
netbios-dgm      138/tcp     #    NETBIOS    Datagram
Service
netbios-dgm      138/udp
netbios-ssn      139/tcp     #    NETBIOS    session
service
netbios-ssn      139/udp
```

ainsi que le fichier /etc/inetd.conf qui doit contenir :

```
netbios-ssn  stream  tcp  nowait  root  /usr/sbin/smbd
smbd
netbios-ns   dgram   udp  wait    root  /usr/sbin/nmbd
nmbd
```

puis vous redémarrez le super démon inetd par :

```
/etc/rc.d/init.d/inet restart
```

Avec cette méthode, les daemons smbd et nmbd ne sont pas lancés au démarrage de linux, c'est le super serveur inetd qui se charge de les lancer lorsqu'un client se connecte aux ports correspondants.

N'utilisez inetd uniquement pour des serveurs ayant une FAIBLE CHARGE.

2.11.3.En utilisant xinetd :

xinetd est le successeur de inetd.

Créez deux fichiers appelés `netbios-ssn` et `netbios-ns` dans le répertoire `/etc/xinetd.d`. Ci-dessous un exemple de configuration, à adapter à votre cas :

```
# cat /etc/xinetd.d/netbios-ssn
service netbios-ssn
{
   socket_type = stream
   protocol = tcp
   wait = no
   user = root
   server = /usr/sbin/smbd
   disable = no
}
# cat /etc/xinetd.d/netbios-ns
service netbios-ns
{
```

```
    socket_type = dgram
    protocol = udp
    wait = no
    user = root
    server = /usr/sbin/nmbd
    disable = no
}
```

Puis pour prendre en compte les modifications, relancez xinetd :

```
# /etc/rc.d/init.d/xinetd restart
```

Mêmes remarques qu'avec inetd…

Avec cette méthode, les daemons smbd et nmbd ne sont pas lancés au démarrage de linux, c'est le super serveur xinetd qui se charge de les lancer lorsqu'un client se connecte aux ports correspondants.

N'utilisez xinetd uniquement pour des serveurs ayant une FAIBLE CHARGE.

2.12.Gestion des utilisateurs

On se limitera ici à une gestion basique des utilisateurs.

Samba gère sa propre liste d'utilisateurs (!= liste des utilisateurs Unix). Il faut donc ajouter chaque utilisateur Unix la liste d'utilisateurs Samba.

Pour simplifier, à chaque utilisateur Samba va correspondre un utilisateur Unix.

Il faut maintenant activer les comptes que l'on souhaite utiliser en leur attribuant un mot de passe :

```
# smbpasswd -a my_user
New SMB password:
Retype new SMB password:
Added user my_user.
```

2.13.Le fichier smb.conf

Le fichier smb.conf permet de définir la configuration de Samba. Il est vérifié toutes les 60 secondes et les modifications sont automatiquement appliquées.
Il est constitué de sections et de paramètres. Chaque section pouvant contenir plusieurs paramètres.

Exemple :

```
[nom_de_section]
        nom_de_parametre = valeur_de_parametre
        read only = No
        browseable = No
```

Chaque section définit un partage de fichier ou d'imprimante, cependant, il existe 3 sections spéciales :

- 'global' : indique les paramètres du serveur applicables à toutes les autres sections

- 'homes' : permet de partager les répertoires personnels des utilisateurs en une seule section
- 'printers' : permet de partager toutes les imprimantes en une seule section

2.13.1.La section 'global'

Voici les principaux paramètres utilisés par la section 'global' :

- netbios name = nom netbios du serveur
- server string = nom du serveur affiché
- interfaces = interfaces sur lesquelles le serveur écoute
- encrypt passwords = obligatoirement Yes
- log file = emplacement du fichier de log
- printing = type de serveur d'impression, généralement lprng ou cups
- guest account = nom d'utilisateur unix correspondant au compte invité

Exemple :

```
[global]

        netbios name = kiwi
        workgroup = LABO-UNIX
        server string = jewom-samba-server
        log file = /var/log/samba/%m.log
        lock directory = /var/lock/samba
        guest account = nobody
        printing = lprng
        encrypt passwords = yes
        smb passwd file = /etc/samba/smbpasswd
```

2.13.2.Le partage de fichier

Il faut créer une section par répertoire partagé (excepté pour les homes).

Les principaux paramètres sont :

- path = chemin vers le répertoire partagé (peut être ignoré pour les homes)
- read only = lecture seule, yes ou no
- comment = description du patage
- browseable = apparait-il dans l'explorateur réseau , yes ou no
- guest ok = autoriser un utilisateur non authentifié à se connecter ou pas

Exemple :

```
[tmp]
        comment = Dossier pour fichiers temporaires
        path = /tmp
        read only = no
        public = yes
        guest ok = yes
```

2.13.3.Le partage d'imprimante

Il faut créer une section par imprimante partagée.

Exemple :

```
[printers]
        comment = All Printers
        path = /var/spool/samba
        printable = Yes
        browseable = No
```

pour plus d'info : http://www.samba.org/samba/ftp/docs/htmldocs/

2.14.Installation de SWAT

SAMBA est livré avec un outil de configuration assez puissant : SWAT. Son rôle est l'édition du fichier de configuration de SAMBA : `/etc/smb.conf` et le contrôle de SAMBA. Il fonctionne via un browser de page WEB (par exemple mozilla).

```
# apt-cache search swat
samba-swat - The Samba SMB server configuration program.
[root@localhost xinetd.d]# apt-get install samba-swat
Reading Package Lists... Done
Building Dependency Tree... Done
The following NEW packages will be installed:
  samba-swat
0 packages upgraded, 1 newly installed, 0 removed and 78
not upgraded.
Need to get 2061kB of archives.
After unpacking 4791kB of additional disk space will be
used.
Get:1  http://ayo.freshrpms.net  redhat/7.3/i386/updates
samba-swat 2.2.7-3.7.3 [2061kB]
Fetched 2061kB in 11s (177kB/s)
Executing RPM (-Uvh)...
Preparing...
########################################### [100%]
                                                1:samba-swat
########################################### [100%]
```

2.14.1.Utilisation de SWAT avec inetd

Modifiez le fichier `/etc/inetd.conf` de façon que celui-ci contienne la ligne :

```
swat        stream    tcp       nowait.400        root   /usr/sbin/swat
swat
```

Il est judicieux de modifier cette ligne pour utiliser `tcpd` de façon à n'autoriser les accès au serveur SAMBA que depuis certains endroits (en modifiant les fichiers `/etc/hosts.allow` et `/etc/hosts.deny`), vous pouvez par exemple remplacer la ligne précédente par :

```
swat  stream    tcp     nowait.400      root    /usr/sbin/tcpd /
usr/sbin/swat
```

Modifier aussi `/etc/services` de façon que celui-ci contienne la ligne :

```
swat    901/tcp
```

Maintenant que SWAT est correctement configuré, il faut redémarrer le super daemon `inetd` pour recenser ses services :

```
# /etc/rc.d/init.d/inetd restart
```

2.14.2.Utilisation de SWAT avec xinetd

Modifier `/etc/xinetd.d/swat` pour activer le service.
2 étapes à suivre :

Modification de `/etc/xinetd.d/swat` :

```
# cat /etc/xinetd.d/swat
service swat
{
   port = 901
   socket_type = stream
   wait = no
   only_from = 127.0.0.1
   user = root
   server = /usr/sbin/swat
   log_on_failure += USERID
   disable = no
}
```

Redémarrer `xinetd` :

```
# /etc/rc.d/init.d/xinetd restart
```

2.14.3.Générer et modifier smb.conf avec SWAT

Pour utiliser SWAT, lancez un navigateur et dans la zone adresse tapez :
http://root@localhost:901

Lorsqu'il vous le sera demandé, identifiez vous en tant que root. SWAT est maintenant lancé. Ce que vous voyez est la page d'accueil de SWAT, elle s'appelle Home (il s'agit en fait de renvois vers les manpages relatives à SAMBA).

L'utilisation de SWAT étant intuitive, elle ne sera pas détaillée ici…

2.15.Samba en tant que PDC

2.15.1.Modification dans smb.conf

Premièrement voici les modifications que vous devez effectuer dans la section global du fichier smb.conf.

```
[global]

    ;Le nom du serveur
    netbios name = toltec
    ;Le nom du groupe de travail
    workgroup = METRAN
    encrypt passwords = yes


    domain master = yes
    local master = yes
    preferred master = yes
    os level = 65
```

```
     security = user
     domain logons = yes

        ;Dit à Samba où il doit placer les profiles
itinérants de
     ;Windows NT/2000/XP
     logon path = \\%L\profiles\%u\%m
     logon script = logon.bat

     logon drive = H:

       ; Permet de spécifier la location des profiles
itinérants de
     ; Windows 95/98/Me
     logon home = \\%L\%u\.win_profile\%m

     time server = yes

     ; instead of jay, use the names of all users in the
Windows NT/2000/XP
     ; Administrators group who log on to the domain
     domain admin group = root jay

      ; Ce script fonctionne sour RedHat mais peut mal
tourner sous
     ;d'autres OS.
     add user script = /usr/sbin/useradd -d /dev/null -g
100 -s /bin/false -M %u
```

Voici les nouveaux partages à placer après la section global :

```
 [netlogon]
     path = /usr/local/samba/lib/netlogon
     writable = no
     browsable = no

 [profiles]
      ; you might wish to use a different directory for
your
     ; Windows NT/2000/XP roaming profiles
     path = /home/samba-ntprof
     browsable = no
     writable = yes
     create mask = 0600
     directory mask = 0700

 [homes]
     read only = no
     browsable = no
     guest ok = no
     map archive = yes
```

Le partage `[profiles]` est nécessaire à l'utilisation de profiles itinérants de Windows NT/2000/XP. Le chemin pointe sur un répertoire du serveur Samba où seront stockés les profiles, le client doit donc être capable de lire et d'écrire des données. Les directives `create mask` et `directory mask` permettent de s'assurer que seuls les utilisateurs autorisés puisse lire et écrire dans ce répertoire et personne d'autre.
Le partage `[homes]` lui est nécessaire à nos définitions de logon drive et logon home.

Vous pouvez dès à présent lancer testparm pour vérifier l'exactitude de la syntaxe de votre fichier smb.conf.

2.15.2.Création des répertoires sur le serveur samba

Les partages `[netlogon]` et `[profiles]` définis dans notre smb.conf référencent des répertoires sur le serveur Samba, il est donc nécessaire de créer ces derniers avec les bonnes permissions bien sûr !

```
# mkdir /usr/local/samba/lib/netlogon
# chmod 775 /usr/local/samba/lib/netlogon
# mkdir /home/samba-ntprof
# chmod 777 /home/samba-ntprof
```

Les noms des répertoires sont totalement arbitraires et choisis à titre d'exemple.

2.15.3.Redémarrer le serveur Samba

Il ne reste plus qu'à redémarrer le serveur Samba et tous les changements seront pris en compte. Pour cela vous pouvez tout simplement saisir :

```
# /etc/rc.d/init.d/smb restart
```

Le serveur devrait être maintenant opérationel et prêt à accepter des requêtes de connexion provenant du domaine.

2.15.4.Ajouter des comptes pour les ordinateurs

Afin d'interagir dans un domaine donné, un système Windows NT/2000/Xp doit être membre de ce domaine. L'apartenance d'une machine à un domaine s'implémente en utilisant des « comptes machine ». Ceux-ci sont, comme nous le verrons, assez semblables à des comptes utilisateurs et permettent au contrôleur de domaine de reconnaître les machines pouvant s'authentifier sur le domaine. Si le PDC est un Windows2000/Xp, ces comptes machine sont stockés dans la base de donnée SAM.

2.16.Plus de précisions sur quelques commandes utiles

2.16.1.smbclient

Le programme `smbclient` permet d'établir des connections avec un serveur SAMBA.
Pour savoir quelles sont les ressources disponible sur un serveur faites :
smbclient -L serveur-samba
En ce qui concerne le mot de passe qui va vous être demandé, vous pouvez mettre n'importe quoi : il ne sera pas vérifié ! Cette connection semble avoir lieu en tant qu'invité.
Pour vous connecter à la ressource `ressource-samba` du serveur `serveur-samba` tapez :
smbclient \\\\serveur-samba\\resource-samba -U utilisateur
ou
smbclient "\\serveur-samba\resource-samba" -U utilisateur
Ne précisez pas d'utilisateur si vous lancez cette commande avec les droits nécessaires.
Après vous être connecté à cette ressource, vous pouvez envoyer des fichier par `put`, en récupérer par `get`. Si la ressource est une imprimante, vous pouvez imprimer en utilisant `print`.

2.16.2.testparm

Le programme testparm sert à vérifier la validité des entrées du fichier /etc/smb.conf. Lorsque vous éditez à la main ce fichier, n'hésitez pas à l'utiliser : il vous dira les erreurs que vous avez peut-être commises en l'éditant. La syntaxe d'appelle est simple :
testparm
S'il vous dit que tout est Ok, vous pouvez relancer SAMBA pour prendre vos modifications en compte.
Ce programme sert aussi à vérifier les deux entrées hosts allow et hosts deny, si vous voulez savoir si la connection à une ressource particulière sera accepté ou pas par SAMBA depuis la machine machine il suffit de taper :
testparm /etc/samba/smb.conf machine
Et vous saurez tout.

2.17.Quelques adresses utiles :

Je vous les livre en vrac :
Vous pouvez toujours vous procurer la dernière version de SAMBA sur le site principal : http://www.samba.org.

Il existe plusieurs utilitaires pour configurer SAMBA. Pour les mots de passes allez voir du côté de gsmb. Pour configurer les ressources partagées par SAMBA, allez voir KSamba.
En remplacement de SWAT, vous pouvez utiliser linuxconf. (si linuxconf est installé sur votre machine et si vous avez autorisé l'accès à linuxconf en réseau, cliquez là)
Le livre Using SAMBA. C'est un livre en Open Content (comme ça on peut être content)! (**Note** : le livre est inclu dans SAMBA à partir de la version 2.0.7)
Le site personnel de G. Blanchet qui explique comment configurer d'un autre point de vue un serveur SAMBA.