RÉSEAUX: TP STATION DISKLESS

Pxe-Dhcp-Tftp
Application à la configuration d'une station Diskless

1. Introduction

Une station "diskless" est un ordinateur ne possédant pas d'unités de stockage permanentes. Lors de la mise sous tension de la machine, cette dernière va chercher sur le réseau une image du noyau du système d'explotation qui va lui permettre de démarrer (image de boot). Avant de pouvoir récupérer cette image, il faut d'abord que la station puisse communiquer sur le réseau, ce qui suppose qu'elle possède une adresse IP, et qu'elle connaisse les différents paramètres nécessaires à toute station connectée (masque de réseau, routeur, dns, ...). La station ne possédant pas de support de stockage, elle doit aller trouver ces informations sur le réseau à chaque fois qu'elle démarre. C'est le protocole Dhcp (Dynamic Host Configuration Protocol) qui permet à une station cliente de récupérer ses paramètres réseaux et une image de boot en s'adressant à un serveur Dhcp.

Le but de ce TP est de vous faire comprendre les différents mécanismes et protocoles qui entrent en jeu dans la problématique du boot via le réseau d'une machine en configurant pas à pas un serveur Dhcp et une station diskless.

2. PXE - Pxelinux

Lors de la mise sous tension d'une machine, le premier composant logiciel s'exécutant est le Bios (Basic Input/Output System). Son travail consiste entre autre à scanner l'ensemble des périphériques bootables d'une machine (lecteur de disquettes, disques durs, CD-Rom,...) jusqu'à trouver sur le secteur d'amorçage de l'un d'entre eux un programme particulier nommé *bootstrap loader* (ou encore *boot loader*, *chargeur*, *gestionnaire d'amorçage*). Ce programme (qui ne fait au plus que quelques centaines de kilo-octets) n'a qu'une seule tâche à réaliser: lire sur le périphérique l'image du noyau du système d'exploitation, la charger en mémoire et passer la main au noyau. Il existe de nombreux *boot loader*: *Lilo*, *Grub*, ceux de DOS/Windows, ...

2.1. **PXE.** Lorsqu'une carte ethernet respecte la norme PXE ceci signifie qu'elle fait partie des périphériques scannés par le Bios lors de la recherche d'un *boot loader*. Cependant, dans ce cas le boot loader ne se trouve pas directement stocké sur la carte. Si le Bios sollicite la carte pour démarrer la machine, le protocole PXE émettra une requête Dhcp sur le réseau pour récupérer l'adresse IP de la machine ainsi qu'un *boot loader* qui à son tour se chargera de placer en mémoire l'image du noyau d'un système d'exploitation (placée elle aussi sur le réseau si la machine ne possède pas de support de stockage). Encore faut-il que ce boot loader soit capable de récupérer sur le réseau le noyau du système. Ce n'est pas le cas de *Lilo* par exemple.

2.2. **Pxelinux.** Pxelinux est un *boot loader* qui a été spécifiquement développé pour charger un noyau Linux via le réseau tout en héritant des informations précédemment obtenues par PXE.

Dans le cadre de ce TP, nous supposons que nous travaillons avec des stations clientes qui n'ont AUCUN support de stockage (donc pas de lecteur de disquette). L'idée est alors la suivante: combiner PXE et Pxelinux afin de pouvoir démarrer un système via le réseau.

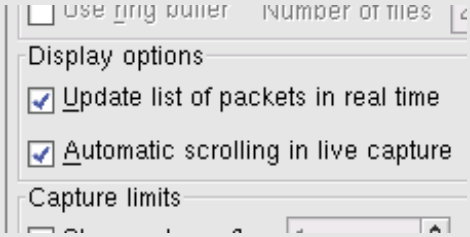
3. Mise en place des machines virtuelles

Vous allez utiliser deux machines virtuelles : l'une d'elle jouera le rôle du serveur et l'autre du poste client diskless. Dans `/home/partage/RESEAU/VBOX/` vous trouverez les scripts `VBoxInitserver` et `VBoxInitclient` qui se chargeront de créer automatiquement vos machines.

Date: Décembre 2009.

4. DHCP

Exercice 1. Démarrez votre station diskless à l'aide de l'interface graphique proposée par la commande `VirtualBox`. Récupérez l'adresse MAC de votre station diskless.
Démarrez votre serveur virtuel. Une fois un prompt obtenu exécutez `startx` pour obtenir une interface graphique. Le clic droit de la souris vous permet d'obtenir un menu à partir duquel vos pouvez lancer un terminal. Exécutez `startnet` pour configurer le réseau. Lancez `wireshark` et faites une capture interactive des paquets. Pour cela, cochez les options *Update list of packets in real time* et *Automatic scrolling in live capture* dans la fenêtre des options de capture.



Fermez la fenêtre des options de capture. Redémarrez votre station diskless et observez les trames. Déduisez-en:

. sur quel protocole de transport se base DHCP ?
. quel est le port sollicité sur le serveur ?
. comment procède votre station diskless pour communiquer sur le réseau ?
. quelles informations sollicitent votre station ?

Le protocole PXE a donc fait une requête DHCP sur le réseau pour obtenir une adresse IP, pour lui répondre il faut mettre en place un serveur DHCP. Ceci nécessite la création du fichier `/etc/dhcpd.conf` qui permet de spécifier entre autres l'association *adresse MAC-adresse IP*. La structure d'un fichier de configuration minimaliste est la suivante (à vous de remplacer les parties écrites en majuscules):

```
not authoritative;
ddns-update-style             none;

subnet ADRESSE_DE_RESEAU netmask MASQUE_DE_RESEAU {
    }


group  {

    default-lease-time            21600;
    max-lease-time                21600;
    use-host-decl-names           on;

    option domain-name            "METTRE ICI LE NOM DU DOMAINE";
    option domain-name-servers    ADRESSE IP DU DNS;
    option broadcast-address      ADRESSE DE BROADCAST DU RESEAU;
    option routers                ADRESSE DU ROUTEUR;
    server-name                   "METTRE ICI LE NOM DU SERVEUR DHCP";


    host METTRE ICI NOM DE LA STATION DISKLESS {
        hardware ethernet     ADRESSE MAC DE LA STATION DISKLESS;
        fixed-address         ADRESSE IP DE LA STATION DISKLESS;
        next-server           ADRESSE IP DU SERVEUR;
    }

}
```

Quelques explications (pous plus de détails voir le man de `dhcpd.conf`):

- `not authoritative` signifie que le serveur ne doit renvoyer aucune réponse s'il reçoit une requête DHCP qui ne le concerne pas,
- `ddns-update-style none` signifie que lorqu'un nouveau poste client se voit attribuer une adresse IP on n'effectue pas de requête vers le serveur DNS pour le mettre à jour.
- la section `subnet` précise le sous réseau pris en compte par le serveur DHCP.
- dans la section `group` on place les paramètres communs à un ensemble de machines.
- les options `default-lease-time` et `max-lease-time` permettent de fixer la durée du bail en secondes (période de validité de l'adresse IP).
- le flag `use-host-decl-names` signifie que le nom mentionné après la déclaration `host` sera le nom attribué à la station cliente.
- la déclaration `next-server` permet d'indiquer l'adresse IP du serveur TFTP contenant les images à télécharger. Dans votre cas, il s'agit de l'adresse IP de votre serveur DHCP.
- les autres options parlent d'elles-mêmes.

Le réseau sur lequel vous travaillez a les caractéristiques suivantes:

. réseau de classe C,
. adresse du DNS et du routeur: `192.168.1.1`,
. nom du domaine: `univ-tln.fr`.

Exercice 2. Lancer le serveur DHCP en tapant `/etc/rc.d/init.d/dhcpd start`. Redémarrez votre station cliente et observez les trames avec wireshark jusqu'à ce qu'un message d'erreur apparaisse sur votre station cliente (arrêtez alors wireshark).

- Comment votre serveur répond-il à la machine qui n'a pas d'adresse IP ?
- Quel protocole de transport utilise-t-il ?
- Sur quel port renvoie-t-il ses informations ?

En regardant la réponse du serveur DHCP (`DHCP Offer`) vous remarquez que parmi les infos contenues dans `Bootstrap Protocol` il est indiqué `Boot file name not given`. Sur votre station cliente vous avez l'erreur `PXE...: No Boot file name received`. En effet le protocole `PXE` s'attend à ce que le serveur DHCP lui renvoie en plus de l'adresse IP de la machine le nom d'un fichier à charger en mémoire pour démarrer la machine (le *boot loader*). Si cette information n'est pas disponible, PXE va relancer une nouvelle requête DHCP jusqu'à ce qu'un serveur lui fournisse toutes les informations dont il a besoin. Etant donné que vous ne disposez que d'un seul serveur DHCP, la requête échoue.

Exercice 3. Editez le fichier `dhcpd.conf` et rajoutez dans la section `host` la ligne `filename "/pxelinux.0"` qui spécifie le nom du fichier contenant le code du *boot loader* pxelinux. Le `/` représente ici la racine du serveur TFTP (cf. paragraphe suivant) et non pas la racine de votre système de fichiers.
Relancez votre serveur DHCP en exécutant `/etc/rc.d/init.d/dhcpd restart` et redémarrez votre station cliente. Observez alors à l'aide de wireshark que le protocole DHCP se déroule en 4 phases. Expliquez ces 4 phases.

Quel nouveau protocole entre en jeu pour le transfert du fichier `pxelinux.0` ? Sur quel protocole de transport se base-t-il et sur quel port du serveur devrait-il être en écoute ?

Remarquez qu'à ce stade même si le fichier n'a pu être récupéré le protcole DHCP s'est entièrement déroulé et vous devez voir apparaître sur la station cliente, son adresse IP et d'autres informations.

5. TFTP

Le protocole TFTP (Trivial File Transfer Protocol) est un protocole permettant de transférer des fichiers d'une machine vers une autre (si on connaît déjà leurs noms et emplacements) sans avoir besoin de s'identifier auparavant. Il constitue donc un trou de sécurité énorme et est désactivé par défaut sur toutes les machines (sinon il suffirait simplement de faire: `tftp machine_a_attaquer`, `get /etc/passwd` pour récupérer le fichier des mots de passe d'une machine où vous n'avez pas de compte.) Le démon TFTP est contrôlé par `xinetd`. Vous trouverez dans `/etc/xinetd.d` le fichier de configuration de `tftp`.

Exercice 4. Editez ce fichier et modifiez l'option qui permet d'activer ce service. Profitez-en pour identifier l'option qui spécifie le répertoire par défaut où doivent être placés les fichiers accessibles en tftp. Ce répertoire est considéré comme le répertoire racine du serveur tftp. Placez y `pxelinux.0`. Pour activer le démon TFTP, il suffit de relancer `xinetd` : `/etc/rc.d/init.d/xinetd restart`.

Redémarrez vore poste client, vous devriez voir apparaître un message vous indiquant qu'un fichier dans `pxelinux.cfg` n'a pas pu être trouvé. Ceci signifie que le code de `pxelinux.0` s'est bien chargé mais que la configuration de ce dernier est incomplète.

Il faut à présent configurer `pxelinux` pour qu'il envoie via TFTP le noyau Linux à votre station cliente.

Exercice 5. Créez à la racine de votre serveur TFTP le répertoire `pxelinux.cfg`. Le boot loader `pxelinux.0` recherche dans ce répertoire un fichier dont le nom est `01-`*adresse_mac_client*. Créez ce fichier et placez à l'intérieur les lignes suivantes ;
```
label linux
kernel vmlinuz-f8
```
Puis redémarrez votre station cliente.

Vous venez de préciser au boot loader qu'il doit charger le noyau dont le nom est `vmlinuz-f8` mais pour que cela fonctionne il faut placer ce dernier à la racine de votre serveur TFTP.

Exercice 6. Copiez le fichier `/boot/vmlinuz-f8` à la racine de votre serveur TFTP et relancez votre poste client.

Que se passe-t-il ? Après avoir fait quelques tests matériels (notamment déterminer le type du processeur, la taille de la mémoire, le nombre de disques durs présents et les différentes partitions), le noyau Linux cherche à monter le *root file system* sur le répertoire `/` afin de lancer l'exécution de `/sbin/init` qui est le processus chargé de lancer tous les programmes nécessaires au fonctionnement du système. Normalement le *root file system* correspond à une partition d'un disque dur (`/dev/hda1` par exemple). Notre machine ne possédant pas de support de stockage nous devons indiquer au noyau que le *root file system* doit être monté à partir du réseau. C'est pourquoi pour l'instant vous avez sur votre poste client l'erreur: `Kernel Panic: unable to mount root fs`. Attention il ne s'agit pas ici de télécharger via le résau le *root file system* (comme vous l'avez fait pour le *boot loader* et le noyau) mais de spécifier au noyau que le répertoire `/` doit être associé à un système de fichiers se trouvant ailleurs sur le réseau. C'est l'objet du protocole NFS. Dans le cadre de ce TP nous n'allons pas poursuivre dans cette voie. Nous allons voir à présent comment il est possible à cette étape d'obtenir un shell en téléchargeant en mémoire vive un système minimaliste.

6. Une phase de boot en 2 temps

Le noyau Linux offre la possibilité de travailler sur des disques mémoire (ou *ramdisk*). Ceci revient tout simplement à considérer une partie de la mémoire centrale (RAM) comme un disque dur où l'on pourra stocker des fichiers. Il est possible de passer en paramètre au noyau l'image d'un système de fichiers qu'il va automatiquement placé en mémoire vive au moment de son démarrage avant d'essayer de monter le `root file system`. Pour celà il faut créer un fichier `initrd.img` qui contiendra l'image compressée du système de fichiers à placer en mémoire. Ce système de fichier une fois décompressé en mémoire sera associé à la racine `/` par le noyau et ce dernier lancera l'exécution de `/init` (script qui doit obligatoirement se trouver dans le système de fichiers).

6.1. **Création du fichier `initrd.img`.** Pour créer ce fichier il faut dans un premier temps créer un répertoire temporaire dans lequel vous placerez tous les fichiers devant constituer le système de fichiers à placer en mémoire (notamment le script `init`). Une fois ceci terminé, la création de l'image se fait en exécutant **dans votre répertoire temporaire** la commande `cpio` qui permet de créer une archive. Cette archive devra être ensuite compressée avec `gzip` et copier à la racine de votre serveur TFTP.
```
find . | cpio -o -c | gzip > ../initrd.img
```

7. Vers la fin du TP

Il s'agit de créer une image vous permettant de lancer un shell. Une fois que vous aurez créé l'image `initrd.img` adéquate (cf. exercice suivant), il faudra éditer votre fichier de configuration dans `pxelinux.cfg` pour y rajouter

la ligne `append initrd=initrd.img`

Exercice 7. Créez l'image `initrd.img` permettant de lancer `tcsh`. Notamment pour pouvoir lancer `/bin/tcsh`,le répertoire `/` étant associé à votre image, ceci suppose qu'il y a dans cette image un répertoire `bin` avec l'exécutable `tcsh`. Vous devez donc au moins créer dans cette dernière un répertoire `bin` et y copier l'exécutable `tcsh`. Ceci n'est pas suffisant. En effet pour s'exécuter `tcsh` a besoin de librairies. Exécutez `ldd /bin/tcsh` pour déterminer les librairies nécessaires. Il faut placer ces dernières dans votre image `initrd` dans un répertoire `lib`. Ne cherchez pas la librairie `linux-gate`, il s'agit d'une libaririe virtuelle incluse dans le code de `tcsh`. Pour trouver les autres éléments à placer dans votre image `initrd` procédez par essai-erreur en redémarrant votre station diskless et en observant les messages d'erreur jusqu'à obtenir le prompt du shell. N'oubliez pas de créer le script `init` qui doit lancer `/bin/tcsh`. Un dernier indice: le `tcsh` lit et écrit par défaut sur un périhpérique nommé console.
ATTENTION: si vous devez recopier des fichiers se trouvant dans `/dev`, utilisez **toujours** l'option `-a` de `cp`.