

Création d'un serveur PXE personnalisé depuis une distribution du noyau Linux de la branche Debian

Prérequis

- Un système Linux (dans cet exemple, nous utiliserons Debian/Ubuntu)
- Droits d'administrateur (sudo)
- Une connexion réseau

Étape 1 : Installation de dnsmasq

1. Mettre à jour la liste des paquets :

```
sudo apt update
```

2. Installer dnsmasq :

```
sudo apt install dnsmasq
```

Étape 2 : Configuration de dnsmasq

1. Sauvegarder la configuration originale :

```
sudo cp /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
```

2. Créer un nouveau fichier de configuration :

```
sudo nano /etc/dnsmasq.conf
```

3. Ajouter les lignes suivantes :

```
port=69
dhcp-range=192.168.1.50,192.168.1.150,12h
dhcp-boot=pxelinux.0
enable-tftp
tftp-root=/tftpboot
```

S'assurer de remplacer `eth0` par le nom de votre interface réseau et ajustez la plage DHCP selon votre configuration réseau.

NB : Penser à vérifier que l'adresse ip de la carte réseau filaire (ethernet) soit bien en 192.168.1.49 par exemple via network-manager (outil graphique) ou via `/etc/network/interfaces` comme suit :

```
sudo nano /etc/network/interfaces
```

Avec les arguments suivants :

```
auto eth0 # remplacer eth0 par le nom de votre interface réseau filaire à chaque
occurrence
iface eth0 inet static
    address 192.168.1.49
    netmask 255.255.255.0
    gateway 192.168.1.1
```

Pour connaître le nom de l'interface réseau filaire, il est possible d'utiliser la commande `ifconfig` le résultat devrait sensiblement ressembler à celui-ci :

```
root@PXE:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.49 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::be24:11ff:fee9:9b8 prefixlen 64 scopeid 0x20<link>
    ether bc:24:11:e9:09:b8 txqueuelen 1000 (Ethernet)
    RX packets 1369771 bytes 111933946 (111.9 MB)
    RX errors 0 dropped 49007 overruns 0 frame 0
    TX packets 81393 bytes 4120418 (4.1 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Ici, `eth0` est le nom de l'interface réseau filaire.

Étape 3 : Préparation du répertoire TFTP

1. Créer le répertoire TFTP :

```
sudo mkdir -p /tftpboot
sudo chmod -R 777 /tftpboot
```

2. Installer syslinux pour obtenir les fichiers nécessaires au démarrage PXE :

```
sudo apt install syslinux pxelinux
```

3. Copier les fichiers nécessaires :

```
sudo cp /usr/lib/PXELINUX/pxelinux.0 /tftpboot/
sudo cp /usr/lib/syslinux/modules/bios/*.c32 /tftpboot/
```

Étape 4 : Configuration du menu PXE

1. Créer le répertoire de configuration PXE :

```
sudo mkdir -p /tftpboot/pxelinux.cfg
```

2. Créer un fichier de configuration par défaut :

```
sudo nano /tftpboot/pxelinux.cfg/default
```

3. Ajouter un menu de base :

```
DEFAULT menu.c32
PROMPT 0
MENU TITLE PXE Boot Menu
TIMEOUT 300

LABEL local
MENU LABEL Boot from local drive
LOCALBOOT 0

LABEL debian
MENU LABEL Install Debian
KERNEL debian-installer/amd64/linux
APPEND initrd=debian-installer/amd64/initrd.gz
```

Étape 5 : Ajout d'une image d'installation

Pour cet exemple, nous utiliserons Debian :

1. Télécharger les fichiers d'installation netboot :

```
cd /tftpbboot
sudo wget http://ftp.debian.org/debian/dists/stable/main/installer-
amd64/current/images/netboot/netboot.tar.gz
sudo tar xzf netboot.tar.gz
sudo rm netboot.tar.gz
```

Étape 6 : Redémarrage et vérification

1. Redémarrer le service dnsmasq :

```
sudo systemctl restart dnsmasq
```

2. Vérifier que le service fonctionne correctement :

```
sudo systemctl status dnsmasq
```

Test du serveur PXE

1. Configurer un ordinateur client pour démarrer via PXE (généralement dans les paramètres BIOS/UEFI).
2. Démarrer l'ordinateur client. Il devrait recevoir une adresse IP de dnsmasq et afficher le menu PXE.
3. Sélectionner l'option d'installation de Debian pour tester le démarrage PXE.

Étape 7 : Récupération d'un iso pour démarrer en live

1. Rendez-vous à l'adresse suivante : <https://store.lacapsule.org/ISO/TOOLBOX-amd64.iso>
2. Une fois le téléchargement terminé, monter l'iso. Dans notre cas, il se trouve dans les Téléchargements :

3. Créer un répertoire qui servira de point de montage pour l'image ISO :

```
sudo mkdir /mnt/iso
```

4. Utiliser la commande `mount` avec les options appropriées pour monter l'image ISO :

```
sudo mount -t iso9660 -o loop '/home/$USER/Téléchargements/TOOLBOX-amd64.iso' /mnt/iso
```

5. Une fois l'image ISO montée, on peut accéder à son contenu en naviguant vers le point de montage :

```
cd /mnt/iso  
ls -la
```

6. Dans cette image iso se trouvent 5 dossiers dont un caché, symbolisé par un point devant son nom :

```
total 16  
drwxr-xr-x 1 lacapsule lacapsule 2048 déc. 30 18:07 ./  
drwxr-x---+ 3 root root 4096 déc. 31 09:14 ../  
drwxr-xr-x 1 lacapsule lacapsule 2048 déc. 30 17:57 boot/  
drwxr-xr-x 1 lacapsule lacapsule 2048 déc. 30 18:07 .disk/  
drwxr-xr-x 1 lacapsule lacapsule 2048 déc. 30 17:57 efi/  
drwxr-xr-x 1 lacapsule lacapsule 2048 déc. 30 17:58 isolinux/  
drwxr-xr-x 1 lacapsule lacapsule 2048 déc. 30 18:01 live/
```

7. Créer un répertoire **TOOLBOX** à la racine du répertoire **tftpboot** qui contiendra les fichiers nécessaire au démarrage du système

```
sudo mkdir /tftpboot/TOOLBOX
```

8. Il faudra copier l'intégralité des répertoires contenus dans notre **ISO** vers notre répertoire **TOOLBOX**

```
sudo cp -r /mnt/iso/* /tftpboot/TOOLBOX
```

Nous pouvons à présent ajouter une entrée de menu dans notre fichier default afin de permettre un boot sur le système contenu dans l'iso.

9. Lorsque c'est terminé il faudra démonter l'ISO :

```
sudo umount /mnt/iso
```

Étape 8 : Configuration du partage NFS

1. Installer le serveur NFS :

```
sudo apt install nfs-kernel-server
```

2. Configurer le partage NFS :

```
sudo nano /etc/exports
```

3. Ajouter la ligne suivante :

```
/tftpbboot *(ro, sync, no_subtree_check)
```

4. Redémarrer le service NFS :

```
sudo systemctl restart nfs-kernel-server && sudo exportfs -a
```

On peut à présent envoyer de gros fichiers tel qu'un filesystem.squashfs via le réseau afin de démarrer dessus en live.

Étape 9 : Modification du menu PXE

1. Ouvrir le fichier de configuration du menu :

```
sudo nano /tftpbboot/pxelinux.cfg/default
```

2. Modifier les options pour inclure le partage NFS vers notre nouvelle distribution :

```
DEFAULT menu.c32
PROMPT 0
MENU TITLE PXE Boot Menu
TIMEOUT 300

LABEL local
MENU LABEL Boot from local drive
LOCALBOOT 0

LABEL debian
MENU LABEL Install Debian
KERNEL debian-installer/amd64/linux
APPEND initrd=debian-installer/amd64/initrd.gz

# Modification du menu de Debian à notre distribution TOOLBOX
LABEL TOOLBOX : Reconditionnement
KERNEL TOOLBOX/live/vmlinuz-5.4.0-72-generic
APPEND boot=live ksdevice=bootif rootfstype=nfs netboot=nfs
nfsroot=192.168.1.127:/tftpbboot/TOOLBOX initrd=TOOLBOX/live/initrd.img-5.4.0-72-
generic ip=dhcp nosplash --
TEXT HELP
Une distribution faite pour le reconditionnement informatique
ENDTEXT
```

Il sera nécessaire de modifier l'adresse IP **192.168.1.127** par celle de notre choix.

1. Pour savoir quelle IP nous devons utiliser, nous devons utiliser celle qui nous est retournée par la commande :

```
ip -4 addr show $(ip route show default | awk '/default/ {print $5}') | grep -oP '(?<=inet\s)\d+(\.\d+){3}'
```

Étape 10 : Configuration du routage

Pour permettre aux clients ayant des adresses IP en 10.x.x.x d'accéder au réseau 192.168.1.x, on doit configurer le routage sur notre serveur PXE :

1. Activer le forwarding IP :

```
sudo sysctl -w net.ipv4.ip_forward=1
```

2. Ajoutez une règle de NAT (Network Address Translation) :

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

3. S'assurer que ces modifications persistent après un redémarrage en les ajoutant à `/etc/sysctl.conf` et en sauvegardant les règles iptables.

Étape 11 : Configuration de l'interface réseau

Configurer l'interface réseau de notre serveur PXE pour qu'elle ait une adresse IP dans les deux plages :

1. Éditer le fichier `/etc/network/interfaces` :

```
sudo nano /etc/network/interfaces
```

1. Ajouter la configuration suivante :

```
auto eth0
iface eth0 inet static
    address 192.168.1.x # Adresse dans le réseau 192.168.1.x
    netmask 255.255.255.0
    gateway 192.168.1.1

iface eth0 inet static
    address 10.0.0.1 # Adresse dans le réseau 10.x.x.x
    netmask 255.255.255.0
```

Étape 12 : Redémarrage des services

Après avoir effectué ces modifications, redémarrer les services concernés :

```
sudo systemctl restart networking
sudo systemctl restart dnsmasq
```

Cette configuration permettra à notre serveur PXE de distribuer des adresses IP dans la plage 10.x.x.x tout en permettant l'accès au réseau 192.168.1.x. Il faut s'assurer d'adapter les adresses IP et les interfaces réseau en fonction de notre configuration spécifique

Étape 13 : Problème avec le service networking

Le message d'erreur indique que le service networking n'existe pas sur le système. Cela peut être dû à plusieurs raisons :

1. On utilise une version récente de Debian qui a migré vers systemd-networkd.
2. Le paquet ifupdown, qui fournit le service networking, n'est pas installé.
3. Le service a été désactivé ou supprimé manuellement.

Étape 14 : Solutions possibles

1. Utiliser systemd-networkd à la place :

```
sudo systemctl enable systemd-networkd
sudo systemctl start systemd-networkd
```

Étape 15 : Personnalisations possibles

Grâce au fichier default

Il est possible de personnaliser le menu du PXE en ajustant les couleurs à notre guise. Pour ce faire, il faudra ajouter des éléments au fichier `/tftpboot/pxelinux.cfg/default`.

⚠ Warning

Vous ne pouvez pas utiliser exactement le même système de personnalisation du menu avec menu.c32 et vesamenu.c32.

- menu.c32 utilise uniquement des codes ANSI pour définir les couleurs (par exemple 37;40 pour texte blanc sur fond noir).
- vesamenu.c32 utilise à la fois des codes ANSI et des codes hexadécimaux au format ARGB pour une personnalisation plus fine des couleurs.
- vesamenu.c32 offre des options de personnalisation supplémentaires, comme la possibilité d'ajouter une image de fond.

Pour rappel, l'utilisation d'une ou l'autre des solutions est déclarée en début de fichier comme suit :

```
# Interface visuelle:
PROMPT 0
DEFAULT vesamenu.c32
```

Exemple d'une personnalisation de menu.c32 :

```
menu color screen 37;40
menu color border 30;44
menu color title 1;36;44
menu color unsel 37;44
menu color hotkey 1;37;44
menu color sel 7;37;40
menu color hotsel 1;7;37;40
menu color disabled 1;30;44
menu color scrollbar 30;44
```

```

menu color tabmsg      31;40
menu color cmdmark     1;36;40
menu color cmdline     37;40
menu color pwdborder   30;47
menu color pwdheader   31;47
menu color pwdentry    30;47
menu color timeout_msg 37;40
menu color timeout     1;37;40
menu color help        37;40
menu color msg07       37;40

```

Exemple d'une personnalisation de vesamenu.c32 :

```

menu color screen      37;40      #000003 #000003 std
menu color border      30;44      #000003 #000003 std
menu color title       1;36;44    #2b2b2b #000003 std
menu color unsel       37;44      #ffffff #000003 std
menu color hotkey      1;37;44    #2b2b2b #000003 std
menu color sel         7;37;40    #000003 #ffffff all
menu color hotsel      1;7;37;40  #2b2b2b #000003 all
menu color disabled    1;30;44    #2b2b2b #000003 std
menu color scrollbar   30;44      #2b2b2b #000003 std
menu color tabmsg      31;40      #2b2b2b #000003 std
menu color cmdmark     1;36;40    #2b2b2b #000003 std
menu color cmdline     37;40      #2b2b2b #000003 std
menu color pwdborder   30;47      #2b2b2b #000003 std
menu color pwdheader   31;47      #2b2b2b #000003 std
menu color pwdentry    30;47      #2b2b2b #000003 std
menu color timeout_msg 37;40      #2b2b2b #000003 std
menu color timeout     1;37;40    #2b2b2b #000003 std
menu color help        37;40      #2b2b2b #000003 std
menu color msg07       37;40      #2b2b2b #000003 std

```

Structure de base

```

menu color [élément] [attributs_texte] [couleur_premier_plan]
[couleur_arrière_plan] [style]

```

Éléments du menu

- screen : Fond d'écran du menu
- border : Bordure du menu
- title : Titre du menu
- unsel : Éléments non sélectionnés
- hotkey : Touches de raccourci
- sel : Éléments sélectionnés
- hotsel : Touche de raccourci de l'élément sélectionné
- disabled : Éléments désactivés
- scrollbar : Barre de défilement
- tabmsg : Message de tabulation

- cmdmark : Marqueur de commande
- cmdline : Ligne de commande
- pwdborder : Bordure du champ de mot de passe
- pwdheader : En-tête du champ de mot de passe
- pwdentree : Saisie du mot de passe
- timeout_msg : Message de temporisation
- timeout : Compteur de temporisation
- help : Texte d'aide
- msg07 : Message spécifique (peut varier)

Attributs de texte

Format : [luminosité];[couleur_texte];[couleur_fond]

- Luminosité : 0 (normal), 1 (brillant)
- Couleur texte/fond : 30-37 (texte), 40-47 (fond)
 - 30/40 : Noir
 - 31/41 : Rouge
 - 32/42 : Vert
 - 33/43 : Jaune
 - 34/44 : Bleu
 - 35/45 : Magenta
 - 36/46 : Cyan
 - 37/47 : Blanc

Couleurs hexadécimales

Format : #RRGGBB

- RR : Rouge (00-FF)
- GG : Vert (00-FF)
- BB : Bleu (00-FF)

Styles

- std : Standard
- all : Applique le style à tout l'élément

Exemple commenté

```
textmenu color screen 37;40 #000003 #000003 std
# Fond d'écran : texte blanc sur fond noir, couleurs très sombres

menu color border 30;44 #000003 #000003 std
# Bordure : texte noir sur fond bleu, couleurs très sombres

menu color title 1;36;44 #2b2b2b #000003 std
# Titre : texte cyan brillant sur fond bleu, gris foncé sur noir

menu color sel 7;37;40 #000003 #ffffff all
# Élément sélectionné : texte blanc inversé sur fond noir, noir sur blanc
```

Conseils de personnalisation

1. Choisir une palette de couleurs cohérente pour une meilleure lisibilité.
2. Utiliser des contrastes pour mettre en évidence les éléments importants.
3. Tester différentes combinaisons pour trouver ce qui convient le mieux à notre environnement.
4. Ne pas hésiter à ajuster les couleurs hexadécimales pour des nuances plus précises.
5. Garder à l'esprit l'accessibilité pour les utilisateurs ayant des problèmes de vision.

Grâce à GRUB2

GRUB2 offre une alternative à PXELINUX pour configurer un environnement de démarrage PXE avec un menu personnalisé.

Il sera nécessaire d'installer GRUB2 sur le serveur comme suit :

```
sudo apt install grub2-tools-efi grub2-pc-modules
```

Il faudra aussi configurer le serveur TFTP en utilisant la commande `grub2-mknetdir` pour créer le répertoire de démarrage réseau GRUB2 dans la racine TFTP.

Pour un système BIOS :

```
grub2-mknetdir --net-directory=/tftpboot --subdir=/boot/grub -d
/usr/lib/grub/i386-pc
```

Pour un système UEFI :

```
grub2-mknetdir --net-directory=/tftpboot --subdir=/boot/grub -d
/usr/lib/grub/x86_64-efi
```

Dans ces commandes :

- `/tftpboot` est le répertoire racine du serveur TFTP.
- `/boot/grub` est le sous-répertoire où les fichiers GRUB2 seront placés.
- Le paramètre `-d` spécifie le répertoire contenant les modules GRUB2 pour l'architecture cible.

Il sera aussi impératif de modifier la configuration de dnsmasq en ouvrant le fichier dans un éditeur :

```
sudo nano /etc/dnsmasq.conf
```

Insérer les lignes suivantes :

```
dhcp-match=set:efi-x86_64,option:client-arch,7  
dhcp-boot=tag:efi-x86_64,boot/grub2/x86_64-efi/core.efi  
dhcp-boot=tag:!efi-x86_64,boot/grub2/i386-pc/core.0
```

Pour finir, il est possible de personnaliser totalement son menu comme dans l'exemple suivant :

[Créer son propre thème de menu au démarrage grâce à GRUB2](#)

<https://github.com/N0r3f/usb-multiboot>