Serveur Linux : PROXY

Mise en place d'un serveur proxy sous Linux

Bouron Dimitri

10/06/2014

Ce document sert de démonstration concise pour l'installation, la configuration d'un serveur proxy sous Linux utilisant squid3.

Table des matières

I. Installation et configuration de base

Une fois l'installe de Ubuntu faite, il faut se connecter avec l'identifiant de connexion créé lors de l'installation de l'OS. Une fois que nous avons réussi à nous connecter nous allons devoir activer l'utilisateur root pour les configurations à venir en utilisant la commande **$ sudo passwd root**. Le mot de passe que nous donnerons à root sera : P@ssword. Puis on se connectera avec les identifiants de ce nouvel utilisateur avec la commande **$ su root**. Voir ci-dessous (attention, l'image ne correspond pas à ce serveur) :

```
mangetsu@serveur-WEB-base:~$ sudo passwd root
[sudo] password for mangetsu:
Entrez le nouveau mot de passe UNIX :
Retapez le nouveau mot de passe UNIX :
passwd: password updated successfully
mangetsu@serveur-WEB-base:~$ su root
Mot de passe :
root@serveur-WEB-base:/home/mangetsu#
```

À partir de maintenant, nous considérerons que toutes les commandes qui suivront sont faites à partir du compte root, même si l'utilisateur simple peut faire la modification pour des raisons de facilités.

A. Apt-get update

On met rapidement la base à jour avec la commande **# apt-get update**.

B. Interfaces

Nous utilisons un serveur, proxy (ou mandataire) pour être exacte, ce qui implique de lui attribuer une adresse IP fixe, nous allons donc attribuer l'adresse fixe directement sur notre serveur proxy. Pour cela nous allons utiliser la commande **# nano /etc/network/interfaces** et nous modiferons le fichier en remplaçant la ligne « iface eth0 dhcp » comme suit :

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
        address 192.168.1.150
        netmask 255.255.255.0
        network 192.168.1.0
        broadcast 192.168.1.255
        gateway 192.168.1.1
        dns-nameservers 192.168.1.1
```

Les adresses IP sont à modifier selon la situation naturellement.

Pour éviter des erreurs, on va faire appel au DHCP (si l'on en dispose d'un) pour obtenir la configuration IP excepté l'adresse et le masque, pour cela le fichier sera comme suit :

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
        address 192.168.1.150
        netmask 255.255.255.0

iface eth0 inet dhcp
```

Pour prendre en compte les modifications, on force la reconfiguration réseau avec la commande **# /etc/init.d/networking restart** et on utilise la commande **# ifconfig** pour vérifier que le changement de valeur est pris en compte.

```
root@serveur-PROXY-base:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:96:bf:a7
          inet adr:192.168.1.39  Bcast:192.168.1.255  Masque:255.255.255.0
          adr inet6: fe80::a00:27ff:fe96:bfa7/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Packets reçus:2275 erreurs:0 :0 overruns:0 frame:0
          TX packets:51 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:198599 (198.5 KB) Octets transmis:3777 (3.7 KB)
```

Après reconfiguration :

```
root@serveur-PROXY-base:~# /etc/init.d/networking restart
 * Running /etc/init.d/networking restart is deprecated because it may not enabl
e again some interfaces
 * Reconfiguring network interfaces...
ssh stop/waiting
ssh start/running, process 2277
ssh stop/waiting
ssh start/running, process 2354
                                                                        [ OK ]
root@serveur-PROXY-base:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:96:bf:a7
          inet adr:192.168.1.150  Bcast:192.168.1.255  Masque:255.255.255.0
          adr inet6: fe80::a00:27ff:fe96:bfa7/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Packets reçus:2358 erreurs:0 :0 overruns:0 frame:0
          TX packets:584 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:173141 (173.1 KB) Octets transmis:33633 (33.6 KB)
```

C. Mise en place de openssh-server

1) Installation du paquet

On va installation openssh-server pour pouvoir se connecter en ssh sur le serveur. On utilise la commande **# apt-get install openssh-server**.

```
root@serveur-PROXY-base:~# apt-get install openssh-server
```

2) Configuration de openssh-server

Nous allons maintenant configurer openssh-server correctement, ou plutôt sécuriser l'accès ssh un minimum pour l'instant en modifier le port d'écoute ainsi que la restriction d'utilisation de root pour se connecter en ssh. Pour cela on utilise la commande **# nano /etc/ssh/sshd_config**, puis on va remplacer le numéro de port (22) à la ligne 5 par le port souhaité mais en faisant attention à utiliser un port correcte et non utilisé. Il faut donc un port inférieur à 65535 et éviter les ports utilisés (pour en savoir plus sur les ports utilisés le plus souvent par les autres services et donc les éviter il faut suivre ce lien). Ce n'est pas tout, on va modifier la valeur du PermitRootLogin à la ligne 27 en remplaçant la valeur actuelle (yes) par la valeur no. Voir ci-dessous :

```
# What ports, IPs and protocols we listen for
Port 12543_
```

```
# Authentication:
LoginGraceTime 120
PermitRootLogin no
StrictModes yes
```

Une fois fait et sauvegardé, on utilisera la commande **# /etc/init.d/ssh restart** pour forcer la prise en compte des modifications.

D. Mise en place de squid3

1) Installation du paquet

Nous allons maintenant installer le paquet pour le service mandataire. On utilisera le service Squid (ou Squid3). Pour installer le paquet nous utiliserons la commande **# apt-get install squid**.

On installe soit Squid, soit Squid3 (on ne choisit pas). Selon le service, les noms des fichiers et répertoires pourront changer (squid ou squid3). Dans le cas qui va suivre on installe Squid3 :

```
root@serveur-PROXY-base:~# apt-get install squid
```

2) Configuration de squid3

On va maintenant configurer Squid3, mais avant ça on va renommer le fichier de configuration (on ne va pas s'en servir mais mieux vaut le conserver). On fait comme suit :

```
root@serveur-PROXY-base:~# cd /etc/squid3/
root@serveur-PROXY-base:/etc/squid3# mv squid.conf squid.conf.old
```

On va créer et modifier le fichier de configuration avec la commande **# nano /etc/squid3/squid.conf**.

```
root@serveur-PROXY-base:~# nano /etc/squid3/squid.conf
```

```
 GNU nano 2.2.6         Fichier : /etc/squid3/squid.conf

visible_hostname serveur-PROXY-base

http_port 192.168.1.150:3128

cache_dir ufs /var/spool/squid3 100 16 256


#################################### ACL ########################################

acl all src all # ACL pour autoriser/refuser tous les réseaux (Source = all)
acl lan src 192.168.1.0/24 # ACL pour autoriser/refuser le réseau 192.168.1.0
acl Safe_ports port 80 # Port HTTP = Port 'sûre'
acl Safe_ports port 443 # Port HTTPS = Port 'sûre'

#################################################################################


# Désactive tous les protocoles sauf les ports sûres
http_access deny !Safe_ports

# Désactive l'accès pour tous les réseaux sauf les clients de l'ACL "lan"
http_access deny !lan

# Port d'écoute du proxy, à préciser dans les navigateurs
http_port 3128
```

Une fois fait et sauvegardé, on utilise la commande **# /etc/init.d/squid3 restart** pour forcer la reconfiguration de Squid3.

```
root@serveur-PROXY-base:~# /etc/init.d/squid3 restart
Rather than invoking init scripts through /etc/init.d, use the service(8)
utility, e.g. service squid3 restart

Since the script you are attempting to invoke has been converted to an
Upstart job, you may also use the stop(8) and then start(8) utilities,
e.g. stop squid3 ; start squid3. The restart(8) utility is also available.
squid3 start/running, process 2383
```

Le proxy est désormais fonctionnel.

3) Compléments de configuration

Nous allons voir certains compléments de configuration pour le proxy (bloquer les noms de domaine, une authentification, bloquer des extensions).

a) Bloquer des noms de domaines

Nous allons voir comment configurer le proxy pour pouvoir bloquer des noms de domaine. Vous trouverez un grand nombre d'exemples de noms de domaine à bloquer selon des catégories en suivant ce lien.

```
# Déclarer un fichier qui contient les domaines à bloquer
acl deny_domain url_regex -i "/etc/squid3/denydomain"

# Refuser les domaines déclarés par l'ACL deny_domain
http_access deny deny_domain_
```

Il suffit ensuite de créer le fichier déclaré dans l'ACL et de saisir tous les domaines à bloquer (un domaine par ligne)

b) Authentification

Nous allons voir comment configurer le proxy pour exiger une authentification de la part des utilisateurs, si l'authentification n'aboutit pas l'utilisateur ne pourra pas accéder à Internet.

On va utiliser une méthode d'authentification simple (htpasswd), pour cela on installe le paquet apache2-utils avec la commande **# apt-get install apache2-utils**.

```
root@serveur-PROXY-base:~# apt-get install apache2-utils
```

Pour créer un utilisateur, on utilise la commande htpasswd, la commande va renseigner dans un fichier le nom d'utilisateur et son mot de passe (en crypté). Si le fichier contenant les utilisateurs n'existe pas encore alors on utilise la commande suivante :

 # htpasswd –cb /etc/squid3/utilisateurs *identifiant mot_de_passe*

Si le fichier existe on utilise la commande suivante :

 # htpasswd –b /etc/squid3/utilisateurs *identifiant mot_de_passe*

On remplacement *identifiant* et *mot_de_passe* par les valeurs souhaitées.

Si on utilise la première commande alors que le fichier existe déjà alors celui-ci sera écrasé puis recréé.

```
# On déclare le programme qui gère l'authentification
auth_param basic program /usr/lib/squid3/ncsa_auth /etc/squid3/utilisateurs

# ACL qui fait que le proxy demandera une authentification
acl utilisateurs proxy_auth REQUIRED

# Refuser l'accès à tous les utilisateurs sauf ceux du fichier utilisateurs
http_access deny !utilisateurs
```

c) Bloquer des extensions

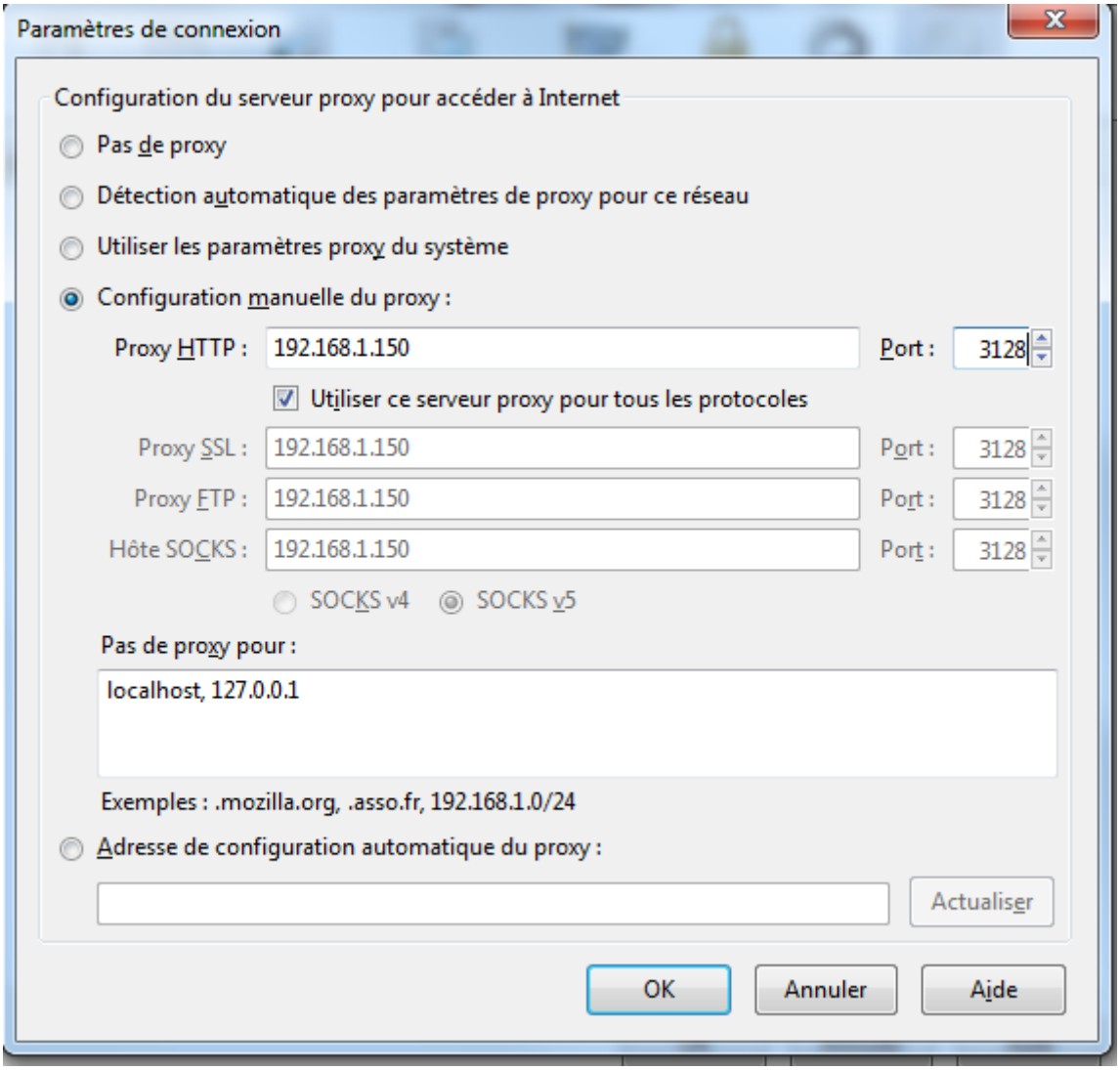
On va voir maintenant comment bloquer des extensions spécifiques. Toujours dans le fichier de configuration, on utilisera les lignes suivantes :

```
# ACL pour bloquer les fichiers AVI
acl extension_avi url_regex -i \.avi$

# Bloquer les fichiers AVI
http_access deny extension_avi
```

E. Configuration des clients

Nous allons maintenant voir comment configurer les clients.



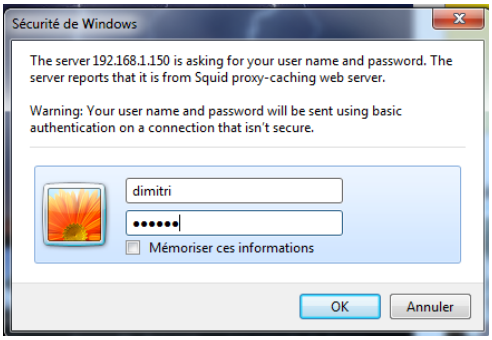
On configure le proxy manuellement, on saisit l'adresse IP du Proxy, son port (3128 par défaut).

Il faut que tous les protocoles utilise le proxy, dans le cas de mozilla il suffit de cocher la case « *Utiliser ce serveur proxy pour tous les protocoles* ».

Il est possible qu'il soit nécessaire de redémarrer le navigateur.

Dans le cas où une authentification utilisateur est requise, votre navigateur doit vous demander d'une certaine manière une authentification.

- Internet Explorer



- Mozilla Firefox