http://www.blanche-de-peuterey.com/Un-reseau-sous-linux-avec-Debian-squid-et-squidguard

Un réseau sous linux avec Debian, squid et squidguard

- Articles -

Date de mise en ligne : mercredi 22 septembre 2010

Configurer un serveur sous Linux, afin de réaliser un réseau domestique ou de petite entreprise, avec un filtre parental, le tout réalisé grâce à des logiciels gratuits.

Mise à jour de l'article

j'ai écrit cet article en septembre 2010. J'avais utilisé à l'époque une distribution Xubuntu, légère, car je ne voulais pas tout faire en ligne de commandes.

Depuis, il s'est passé pas mal de choses :

1. J'ai quitté Xubuntu pour Debian, beaucoup plus stable et opérationnel pour un serveur. Et les lignes de commandes, ce n'est pas si compliqué que cela.

2. Le disque dur de mon serveur a planté, et j'ai du tout réinstaller. J'en profite pour faire une mise à jour de cet article.

Introduction

J'ai beaucoup galéré, n'ayons pas peur des mots. Cela m'a pris un certain temps pour parvenir à ce que je voulais : réaliser la configuration d'un serveur pour un centre culturel. Permettre à chaque utilisateur d'aller sur le net, mettre en place un « filtre parental », capable de bloquer des pages indésirables et de faire un filtrage horaire, éventuellement installer un intranet.

Pas mal de recherche sur Internet, et une fois que tout fonctionne, l'envie de faire un tutoriel pas à pas, pour que je puisse refaire ce que j'ai fais, et que cela puisse profiter à d'autres. Je ne prétends pas être un spécialiste, j'ai une connaissance de base de linux (en ligne de commande, cela va de soit), connaissance minimale indispensable à tous ceux qui voudraient se lancer dans la même aventure.

La configuration matérielle et logicielle

- Un serveur (HP Prolian en l'occurrence)
- deux cartes réseaux (et si la carte réseau native n'est pas bien gérée par Linux, n'hésitez-pas à en acheter une à la Fnac, pour 7 Euros... c'est ce que j'ai fais)
- un hub
- des ordinateurs reliés au réseau (un ordinateur dans chaque chambre ou bureau)

Des utilisateurs pas spécialistes du tout, partisans du « plug and play » : je branche mon ordinateur, et je veux naviguer sur le web.

Le serveur servira de serveur DHCP : c'est lui qui va donner les adresses IP aux différents ordinateurs du réseau.

L'installation de la distribution Linux

J'ai choisi de travailler avec un LiveCD d'une distribution Debian 8. L'installation se fait en mode texte, et se passe bien. Y compris pour le formatage du disque dur, la création des partitions, etc.

Par la suite, toutes les instructions sont données comme si vous étiez connecté en root. Ce qui n'est pas recommandé, qui plus est ce qui n'est pas possible si vous vous connectez en ssh.

Donc, ajoutez votre user au groupe sudo

```
adduser user sudo
```

Ensuite, surtout si vous vous connectez en ssh, utilisez sudo pour toutes les instructions.

Comment partager la connexion internet ?

C'est là que les choses sérieuses commencent. Je me contente de traduire un excellent article que vous trouverez ici, intitulé : Comment partager une connexion internet http://ubuntuforums.org/showthread.php?t=713874

[REM] : j'ai lu pas mal de textes. J'ai essayé d'appliquer pas mal de tutoriels, sans succès. Il y avait toujours quelque chose qui ne marchait pas. Celui-ci est le seul qui aborde le sujet simplement, sans situation particulière, et que j'ai suivi à la lettre, avec succès.

Il est indispensable **de bien connaître quelle carte de votre serveur est connectée au Net**, et quelle carte est connectée au réseau. Sans cela, vous êtes mort.

Pour connaître la carte connectée à Internet et celle connectée au réseau interne, juste après l'installation de votre distribution Linux, faites dans un terminal :

```
sudo ifconfig
```

Le résultat vous donnera des informations sur chaque carte. La carte dont toutes les informations sont à zéro est la carte connectée au réseau, la carte qui a une adresse est connectée à Internet.

Je traduis maintenant l'article de référence :

Dans l'exemple du texte de référence, la carte connectée au réseau est eth0, et celle connectée au net est eth1.

La carte connectée au réseau interne a besoin d'une adresse statique. Pour cela, éditer le fichier de configuration du réseau, et attribuez à eth0 une adresse statique :

```
sudo vi /etc/network/interfaces
```

ajoutez les lignes suivantes :

```
auto eth1
 iface eth1 inet dhcp

 auto eth0
 iface eth0 inet static
 address 192.168.0.1
 netmask 255.255.255.0
 broadcast 192.168.0.1
 network 192.168.0.0
 gateway 192.168.0.1
```

Remarques :

1) j'ai utilisé vi comme éditeur de texte. Un peu déroutant au début, il y a quelques commandes simples à connaître. Par la suite, lorsque mon réseau fonctionnera, j'installerai un serveur ssh, et je pourrai administrer mon serveur depuis mon ordinateur, de façon beaucoup plus confortable qu'avec vi. **Vous pouvez également utiliser nano sur Debian**.

2) à vous de donner à votre réseau les valeurs que vous voulez. J'ai pris des valeurs plus que standard.

3) Attention : en anglais, « address » s'écrit avec 2 « d » ...

Les modifications introduites dans le fichier /etc/network/interfaces n'interviennent qu'après un redémarrage du serveur.

Activer « l'IP forwarding ».

Pour que les adresses IP extérieures soient communiqués au réseau intérieur, il faut autoriser l'IP forwarding (désactivé par défaut dans Ubuntu). Pour cela :

```
sudo vi /etc/sysctl.conf
```

recherchez les lignes suivantes :

```
#net.ipv4.ip_forward=1
 #net.ipv6.conf.all.forwarding=1
```

Supprimez le # pour rendre ces lignes actives

Ajoutez également es lignes suivantes :

```
net.ipv4.conf.default.forwarding=1
 net.ipv4.conf.all.forwarding=1
```

Redémarrez votre système.

Configurez les iptables

Pour permettre aux paquets de passer à travers votre routeur, nous devons ajouter quelques instructions. Pour de plus amples informations, je vous renvoie à différents documents que l'on peut trouver sur le net au sujet du natage d'adresses.

Pour cela, éditez le fichier rc.local :

```
sudo vi /etc/rc.local
```

Ajoutez les lignes suivantes :

```
/sbin/iptables -P FORWARD ACCEPT
 /sbin/iptables --table nat -A POSTROUTING -o eth1 -j MASQUERADE
```

Configuration du serveur DHCP

Pour que les ordinateurs du réseau puissent avoir une adresse IP automatiquement, nous allons installer sur notre machine-serveur un serveur DHCP. Voir là encore sur le net toutes les infos sur ce qu'est un serveur DHCP

Telle qu'elle est décrite dans le tuto de référence en anglais, et comme l'auteur le dit lui-même « The start will fail, but that is nothing to worry about. » C'est à dire : « Une fois installé, il y aura un échec au démarrage du serveur, mais il ne faut pas s'en inquiéter » (sous-entendu : nous allons régler le problème)

Donc,

```
sudo apt-get install isc-dhcp-server
```

ATTENTION : avant le serveur dhcp s'appelait dhcp3-server. Il a changé de nom et les dossiers d'installation et de configuration aussi !

L'installation se fait, et comme prévu, le serveur ne démarre pas. Pour aller plus loin, vous avez besoin d'un peu d'information, entre autre de connaître les serveurs DNS que vous utilisez (ceux de votre fournisseur d'accès)

Pour cela, tapez la commande :

```
cat /etc/resolv.conf
```

et prenez note des adresses des serveurs de noms de domaine. Nous allons appeler ces adresses Nameserver1 et Nameserver2.

Il faut maintenant configurer le serveur DHCP. Pour cela :

```
sudo vi /etc/dhcp/dhcpd.conf
```

Et recopiez dans le fichier de conf les lignes suivantes : (éventuellement adaptées à votre situation)

```
ddns-update-style none;
 option domain-name "example.org";
```

```
option domain-name-servers Nameserver1, Nameserver2;
option routers 192.168.0.1;
authoritative;
subnet 192.168.0.0 netmask 255.255.255.0 {
range 192.168.0.100 192.168.0.200;
}

default-lease-time 600;
max-lease-time 7200;
log-facility local7;
```

Attention aux " ;" en fin de ligne...

Regardez la ligne "range" : on définit ici la plage d'adresses de nos machines clientes. Adaptez ces données en fonction de vos besoins. Ici, je peux connecter 100 machines. (c'est un peu surdimensionné dans mon cas...)

La dernière chose à faire pour que le serveur dhcp fonctionne est de lui dire sur quelle carte réseau il doit écouter. Dans notre cas, la carte réseau d'écoute est eth0.

Pour cela,

```
sudo vi /etc/default/isc-dhcp-server
```

éditez la ligne (ou écrivez-là si elle n'existe pas) et marquez :

```
INTERFACES= "eth0"
```

Redémarrez l'ensemble, et normalement, CA MARCHE !

C'est à dire :

- chaque poste client du réseau a une adresse IP
- depuis chaque poste client du réseau, vous pouvez naviguez sur le net.

Si cela ne marche pas, n'allez pas plus loin... (il n'y a aucune raison que cela fonctionne plus tard par magie, d'autant plus que nous allons complexifier notre système)

Si tout fonctionne, prenez un café, faites une pose, respirez par les narines... et quand vous êtes prêt, passez à l'étape suivante.

Mise en place du filtre parental

Installez un serveur proxy

Maintenant que le réseau fonctionne, nous voulons un filtre parental, qui bloque les pages indésirables, et qui réalise également un filtrage horaire par poste.

C'est à dire : Avoir un filtrage qui empêche Jacques d'aller sur le net de telle heure à telle heure, mais qui permette néanmoins à Sophie d'y aller. Le filtre Squidguard réalise cela, le tout gratuitement... et avec une grande souplesse.

Quel est le principe ? Il s'agit de mettre en place un serveur proxy, par lequel tout le flux va passer. Sur ce proxy, il y a squidguard, qui va filtrer le flux. Si le contenu n'est pas acceptable, il est redirigé vers une adresse qui bloque l'affichage.

Installation de Squid

Pour fonctionner, Squidguard a besoin d'un serveur proxy. J'imagine qu'il en existe plusieurs, nous allons utiliser l'un des plus connus, Squid.

Attention, la version de squid a changé, ainsi que les noms des répertoires.

Pour cela,

```
sudo apt-get install squid3
```

Configuration des ACL

ACL=Access Control List

Une fois Squid installé, il démarre normalement sans problème. Il vous faut maintenant le configurer, pour qu'il prenne en compte votre réseau. Si vous ne faites rien, par défaut, Squid bloquera la navigation depuis vos machines clients.

Pour cela,

```
sudo vi /etc/squid3/squid.conf
```

Je vous conseille d'alléger le fichier de configuration et de supprimer les lignes inutiles (toutes celles qui sont en commentaires, avec le « # » devant). Quoi qu'il en soit, vous devez créer une ACL qui autorise l'accès à votre réseau ; Vous obtenez alors un fichier de conf qui ressemble à cela :

```
acl localnet src 172.16.0.0/12  # RFC1918 possible internal network
 acl localnet src 192.168.0.0/16 # RFC1918 possible internal network

 acl SSL_ports port 443          # https
 acl SSL_ports port 563          # snews
 acl SSL_ports port 873          # rsync
 acl Safe_ports port 80          # http
 acl Safe_ports port 21          # ftp
 acl Safe_ports port 443         # https
 acl Safe_ports port 70          # gopher
 acl Safe_ports port 210         # wais
 acl Safe_ports port 1025-65535  # unregistered ports
```

```
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl Safe_ports port 631         # cups
acl Safe_ports port 873         # rsync
acl Safe_ports port 901         # SWAT
acl lan_veymont src 192.168.0.0/255.255.255.0
acl purge method PURGE
acl CONNECT method CONNECT
http_access allow manager localhost
http_access deny manager
http_access allow purge localhost
http_access deny purge
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localhost
http_access allow lan_veymont
http_access deny all
icp_access allow localnet
icp_access deny all

http_port 3128
hierarchy_stoplist cgi-bin ?
access_log /var/log/squid3/access.log squid
refresh_pattern ^ftp:           1440    20%     10080
refresh_pattern ^gopher:        1440    0%      1440
refresh_pattern -i (/cgi-bin/|\?) 0     0%      0
refresh_pattern (Release|Package(.gz)*)$        0       20%     2880
refresh_pattern .               0       20%     4320
acl shoutcast rep_header X-HTTP09-First-Line ^ICY.[0-9]
upgrade_http0.9 deny shoutcast
acl apache rep_header Server ^Apache
broken_vary_encoding allow apache
extension_methods REPORT MERGE MKACTIVITY CHECKOUT
hosts_file /etc/hosts
coredump_dir /var/spool/squid3
```

Vous voyez bien la ligne

```
acl lan_veymont src 192.168.0.0/255.255.255.0
```

dans laquelle je définis mon réseau local, et j'autorise ce réseau local par la ligne

```
http_access allow lan_veymont
```

Redémarrez Squid, et testez le réseau depuis une machine cliente. A ce stade, votre navigateur doit avoir un proxy pour sortir. C'est ce que l'on veut : que le flux internet passe par ce proxy.

[REM]

Arrivé à ce stade, je me suis trouvé confronté au problème suivant : depuis une machine cliente, si je mettais le proxy dans mon navigateur, je pouvais sortir sur le net. Cela signifiait que le proxy fonctionnait.

Mais : si je ne mettais pas le proxy dans mon navigateur, je pouvais également sortir sur le net... Ce qui ne convenait pas... Puisque je veux forcer les machines clientes à passer par le proxy et par le filtre parental, il ne faut pas qu'elles puissent naviguer sur le net si le proxy n'est pas indiqué dans le navigateur.

Ce fut une rude recherche... La solution m'a été donnée dans le forum Ubuntu dans le topic Squid et réseau local.

Pour que le résultat soit permanent, j'ai écrit ces règles dans mon fichier rc.local dont je vous ai parlé plus haut :

```
/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --to 192.168.1.1:3128
 /sbin/iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 -j REDIRECT --to-port 3128
```

Ainsi, toutes les requêtes qui passent par le port 80 sont redirigées sur le port du squid.

Rendre le proxy transparent

Une fois que vous avez testé que votre serveur proxy fonctionne, c'est à dire que seuls les navigateurs ayant renseigné l'adresse du proxy peuvent sortir sur le net, on peut rendre le proxy transparent.

Mais il ne faut pas le rendre transparent tout de suite, histoire de bien tester son fonctionnement. Une fois que l'on est sur que le serveur proxy est bien réglé, on le rend transparent. C'est à dire que l'on n'aura plus besoin de renseigner l'adresse du proxy dans son navigateur.

Pour cela, dans le fichier squid.conf, sur la ligne où vous indiquez le port du proxy, ajoutez "transparent"

```
http_port 3128 transparent
```

Installation de SquidGuard

On peut suivre les indications données dans la doc du forum ubuntu. Mais elles ne sont plus à jour, puisque la version de squid a changé, ainsi que les répertoires. Je détaille l'installation de SquiGuard ci-dessous.

Dans l'ordre :

- Installer squidguard
- configurer squidguard, via squidguard.conf
- **Ajouter dans le fichier de conf de squid la ligne qui fait passer le flux par squidguard** (important...)
- Construire les bases de données
- créer un script qui met à jour ces bases de données

Installer SquidGuard

```
sudo apt-get install squidguard
```

Récupérer les listes noires

Je suis les indications données par le forum Ubuntu

Configurer SquidGard

Voir pour cela mon article sur le fichier squidguard.conf

Ajouter la redirection dans le fichier squid.conf

Ligne fondamentale à ne pas oublier, sinon squid n'ira pas tout seul faire passer le flux par squidGuard

```
url_rewrite_program /usr/bin/squidGuard -c /etc/squidguard/squidGuard.conf
```

Construire les bases de données

```
squidGuard -C all
```

On nous dit que cela peut prendre du temps... Certes. Mais seules les destinations renseignées dans votre fichier squidGuard.conf sont construites. Si vous avez gardez les mêmes destinations que celles indiquées dans mon fichier squidGuard.conf, il y en a 5, et donc cela ne prend pas trois heures. Si le système est bloqué, regardez dans les logs de SquidGuard, qui vous dira ce qui se passe. Généralement, il y a une erreur dans le fichier squidGuard.conf, et SquidGuard passe en "emergency mode"

Mise à jour hebdomadaire de la base de données

Là encore, j'ai suivi le tuto du Forum Ubuntu

Module Webmin de Squidguard

Contrairement à ce que je disait plus haut, j'ai également abandonné l'utilisation de webmin pour la gestion de SquidGuard, le module Webmin de SquidGuard a des erreurs et ne convient pas pour une version de squid supérieure ou égale à squid 2.6.

Et que fait-on du https?

Tout cela est bien gentil. Mais autant ce filtrage était bien utile en 2010, il devient très relatif en 2018, puisque presque tous les sites sont passés en https, et que le https n'est pas filtrable.

Pour aller plus loin

- Installer un proxy Squid et un filtrage avec SquidGuard sous Debian
- Contrôle parental, sur le Forum Ubuntu
- L'Internet rapide et permanent (un peu la bible dans le domaine)