

Serveur Proxy Squid.

Présentation :

Un proxy est un composant logiciel informatique qui joue le rôle d'intermédiaire en se plaçant entre deux hôtes pour faciliter ou surveiller leurs échanges donc il sert à mettre en cache des éléments et à filtrer des données.

Par extension, on appelle aussi proxy un matériel comme un serveur mis en place pour assurer le fonctionnement de tels services. Il peut être en mode serveur ou en mode transparent

Objectif :

Installer et configurer un serveur proxy.

Pré requis :

- ➔ Un ordinateur sur Linux (ici, on utilisera la Debian 8.2)
- ➔ Avoir une connexion internet
- ➔ Avoir une IP fixe pour le serveur
- ➔ Ma machine s'appelle squid et son @IP est 192.168.1.137/24

Sommaire :

- I. Installation de Squid
- II. Configuration de base
- III. Les contrôles d'accès
- IV. Authentification des utilisateurs
- V. SquidGuard
- VI. Analyseur de log Lightsquid
- VII. Configuration d'un navigateur via un script
- VIII. Annexes

I. Installation de Squid

On va installer le paquet d'installation de Squid :

```
apt-get install squid3
```

On remarque que le port d'écoute par défaut de Squid est 3128 :

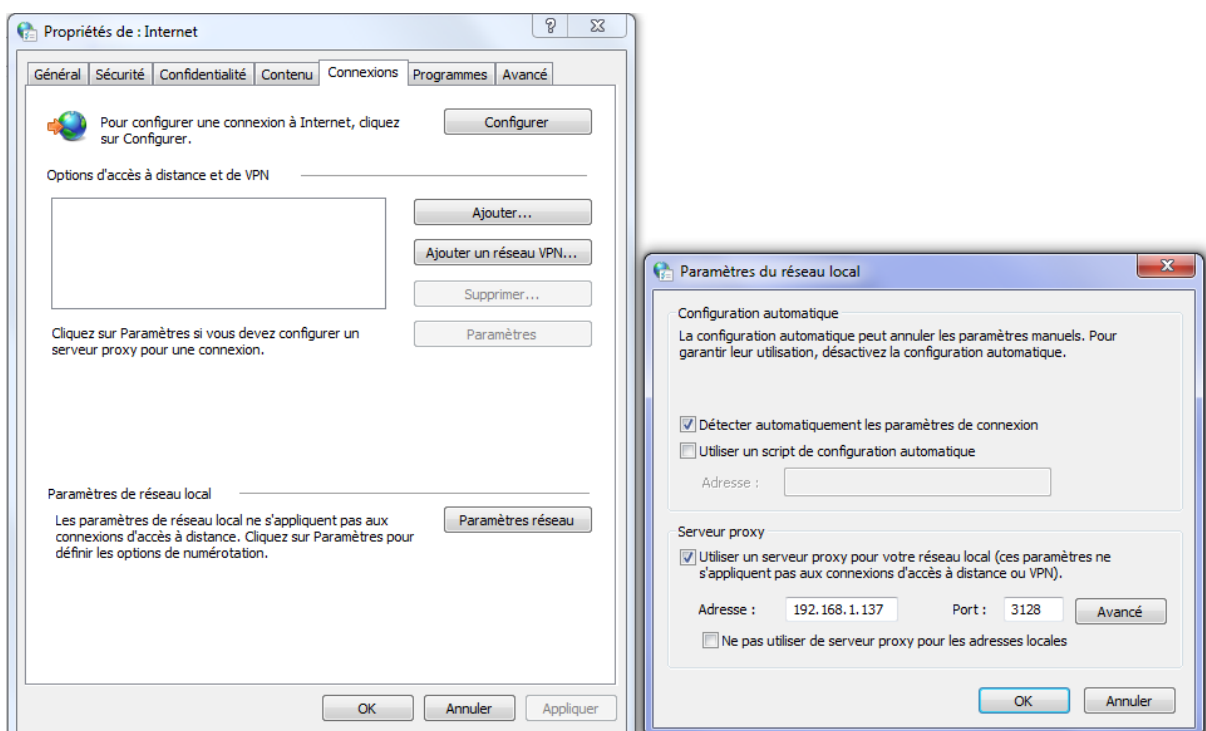
```
# Squid normally listens to port 3128
http_port 3128
```

On remarque aussi que lors de l'installation de Squid, un utilisateur proxy appartenant au groupe proxy a été créé :

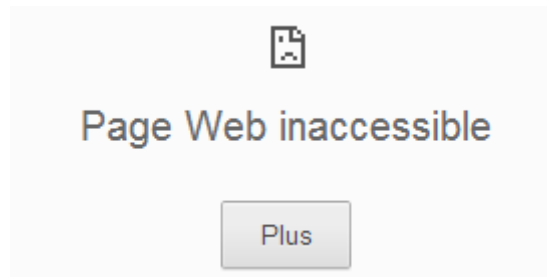
```
root@squid:~# cat /etc/passwd | grep proxy
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
root@squid:~# cat /etc/group | grep proxy
proxy:x:13:
```

II. Configuration de base

Nous allons paramétrer le navigateur pour qu'il utilise notre proxy. Dans les paramètres avancés d'internet puis « Modifier les paramètres du proxy » puis paramètre proxy, il faut entrer l'@IP du Serveur Proxy /port :



Maintenant, nous pouvons plus surfer sur Internet :



Quand on consulte le fichier de log de Squid, on remarque bien l'accès à Internet qui est interdit (denied) :

```
GNU nano 2.2.6      Fichier : /var/log/squid3/access.log      Modif
1443796089.626      0 192.168.1.56 TCP_DENIED/403 3604 CONNECT www.google.fr
1443796089.691      0 192.168.1.56 TCP_DENIED/403 3604 CONNECT www.google.fr
1443796089.744      0 192.168.1.56 TCP_DENIED/403 3604 CONNECT www.google.fr
```

Avant toute modification du fichier de conf de Squid, on crée une copie :

```
root@squid:/etc/squid3# cp squid.conf squid.conf.save
```

Maintenant, on va expurger les lignes de commentaires du fichier qui contient environ 5000 lignes :

```
root@squid:/etc/squid3# cat squid.conf.save | grep -v ^# | grep -v ^$ > squid.conf
```

Le fonctionnement de cette commande consiste à afficher la sauvegarde du fichier, le premier grep retire les lignes de commentaire grâce à l'option -v (inverse), le deuxième grep retire les lignes vides grâce encore à l'option -v (inverse).

Enfin, le > *squid.conf* permet de sauvegarder le résultat dans le fichier de configuration de Squid.

On va maintenant ajouter 4 lignes à la fin du fichier qui vont permettre à l'utilisateur proxy de faire des requêtes sur le serveur et créer un emplacement de stockage des données et réglage des niveaux :

```
cache_effective_user proxy
cache_effective_group proxy
cache_mem 16 Mb
cache_dir ufs /var/spool/squid3 120 16 128
```

La dernière ligne permet de spécifier le cache du disque dur qui sera affecté à Squid.

Ufs est le type de système de stockage, puis on précise l'emplacement du cache. 120 Mo est la taille du cache, 16 répertoires de niveau 1 de l'arborescence du cache est 128 pour le niveau 2.

On va tester à nouveau avec le client et nous avons toujours le même message erreur :

```
1443796720.286      0 192.168.1.56 TCP_DENIED/403 3604 CONNECT www.google.fr:443
- HIER_NONE/- text/html
1443796720.299      0 192.168.1.56 TCP_DENIED/403 3604 CONNECT www.google.fr:443
- HIER_NONE/- text/html
```

III. Les contrôles d'accès

On va utiliser maintenant les ACL qui permettent de contrôler les permissions que l'on attribue sur des adresses IP, c'est pourquoi on va vérifier que le noyau de la debian supporte les ACL (y=yes) :

```
root@squid:/# cat /boot/config-3.16.0-4-amd64 | grep ACL
CONFIG_EXT4_FS_POSIX_ACL=y
CONFIG_REISERFS_FS_POSIX_ACL=y
CONFIG_JFS_POSIX_ACL=y
CONFIG_XFS_POSIX_ACL=y
CONFIG_BTRFS_FS_POSIX_ACL=y
CONFIG_FS_POSIX_ACL=y
CONFIG_TMPFS_POSIX_ACL=y
# CONFIG_HFSPLUS_FS_POSIX_ACL is not set
CONFIG_JFFS2_FS_POSIX_ACL=y
CONFIG_F2FS_FS_POSIX_ACL=y
CONFIG_NFS_V3_ACL=y
CONFIG_NFSD_V2_ACL=y
CONFIG_NFSD_V3_ACL=y
CONFIG_NFS_ACL_SUPPORT=m
CONFIG_CEPH_FS_POSIX_ACL=y
CONFIG_CIFS_ACL=y
CONFIG_9P_FS_POSIX_ACL=y
```

Avec la commande `setfacl` (-h pour help):

```
root@squid:/# setfacl -h
setfacl 2.2.52 -- définir les listes de contrôle d'accès des fichiers (ACL)
Utilisation : setfacl [-bkndRLP] { -m|-M|-x|-X ... } file ...
-m, --modify=acl          modifier l'ACL(s) actuel de fichier(s)
-M, --modify-file=fichier lire l'entrée ACL à modifier du fichier
-x, --remove=acl          supprimer les entrées de l'ACL des fichier
-X, --remove-file=fichier lire les entrées ACL à supprimer du fichier
-b, --remove-all         supprimer toutes les entrées ACL étendues
-k, --remove-default     supprimer l'ACL par défaut
--set=acl                 set the ACL of file(s), replacing the current ACL
--set-file=file           read ACL entries to set from file
--mask                    do recalculate the effective rights mask
-n, --no-mask             ne pas recalculer les masques de droits en vigueur
-d, --default             les opérations s'appliquent à l'ACL par défaut
-R, --recursive           parcourir récursivement les sous-répertoires
-L, --logical             suivre les liens symboliques
-P, --physical            ne pas suivre les liens symboliques
--restore=fichier        restaurer les ACL (inverse de « getfacl -R »)
--test                   mode test (les ACL ne sont pas modifiés)
-v, --version             print version and exit
-h, --help                this help text
```

Avec la commande `getfacl` (-h pour help):

```
root@squid:/# getfacl -h
getfacl 2.2.52 -- obtenir les listes de contrôle d'accès du fichier
Utilisation : getfacl [-aceEsRLPtpndvh] fichier...
-a, --access              display the file access control list only
-d, --default             display the default access control list only
-c, --omit-header        do not display the comment header
-e, --all-effective       print all effective rights
-E, --no-effective       print no effective rights
-s, --skip-base           skip files that only have the base entries
-R, --recursive           recurse into subdirectories
-L, --logical             logical walk, follow symbolic links
-P, --physical            physical walk, do not follow symbolic links
-t, --tabular             use tabular output format
-n, --numeric             print numeric user/group identifiers
-p, --absolute-names     don't strip leading '/' in pathnames
-v, --version             print version and exit
-h, --help                this help text
```

On va ajouter la définition ACL pour notre réseau local puis l'autorisation comme premier. On ajoute ces deux lignes en jaune suivant notre réseau et entre les lignes `acl` et `https_access` :

```
acl Safe_ports port 591          # filemaker
acl Safe_ports port 777          # multiling http
acl CONNECT method CONNECT

acl lan src 192.168.1.0/24
http_access allow lan

http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
```

Les lignes commençant par ACL définissent une zone/plage d'IP ou de ports et pour http_access, ils définissent les droits accordés ACL.

On peut tester dans un navigateur maintenant et voir que cela fonctionne. Dans les logs, le TCP_DENIED n'est plus là mais TCP_MISS :

```
192.168.1.55 TCP_MISS/302 1618 GET http://redirector.
192.168.1.55 TCP_MISS/206 378184 GET http://r12---sn-
192.168.1.55 TCP_MISS/302 1618 GET http://redirector.
192.168.1.55 TCP_MISS/206 382913 GET http://r12---sn-
192.168.1.55 TCP_MISS/302 1618 GET http://redirector.
192.168.1.55 TCP_MISS/206 397746 GET http://r12---sn-
192.168.1.55 TCP_MISS/302 1618 GET http://redirector.
192.168.1.55 TCP_MISS/206 226340 GET http://r12---sn-
```

L'ordre d'application des ACL est important car le fichier de configuration de Squid se lit de haut en bas, c'est pourquoi il faut mettre les ACL en premier puis les http_access en second.

Squid permet de restreindre l'accès de certains postes clients à une plage horaire. Il y a une syntaxe à respecter et les jours sont indiqués par initiale en langue ANGLAISE.

On va changer le fichier et mettre l'accès par exemple au pc client ayant l'@IP 192.168.1.12 et aux horaires 16h à 17h30.

```
acl allowed_hosts src 192.168.1.12
acl limithour time 16:00-17:30
http_access allow allowed_hosts limithour
```

IV. Authentification des utilisateurs.

On va créer deux utilisateurs dans le fichier /etc/squid3/squidusers :

```
root@squid:/etc/squid3# touch squidusers
root@squid:/etc/squid3# htpasswd -b squidusers tintin reporter
Adding password for user tintin
root@squid:/etc/squid3# htpasswd -b squidusers milou chien
Adding password for user milou
```

On va modifier le fichier de conf et rajouter ces lignes au tout début du fichier :

```
auth_param basic program /usr/lib/squid3/basic_ncsa_auth /etc/squid3/squidusers
auth_param basic children 5
auth_param basic realm Squid proxy 2A
authenticate_ttl 1 hour
authenticate_ip_ttl 60 seconds
```

Puis les lignes jaunes en respectant l'ordre :

```
acl CONNECT method CONNECT
acl allowed_hosts src 192.168.1.12
acl limithour time 16:00-17:30
http_access allow allowed_hosts limithour

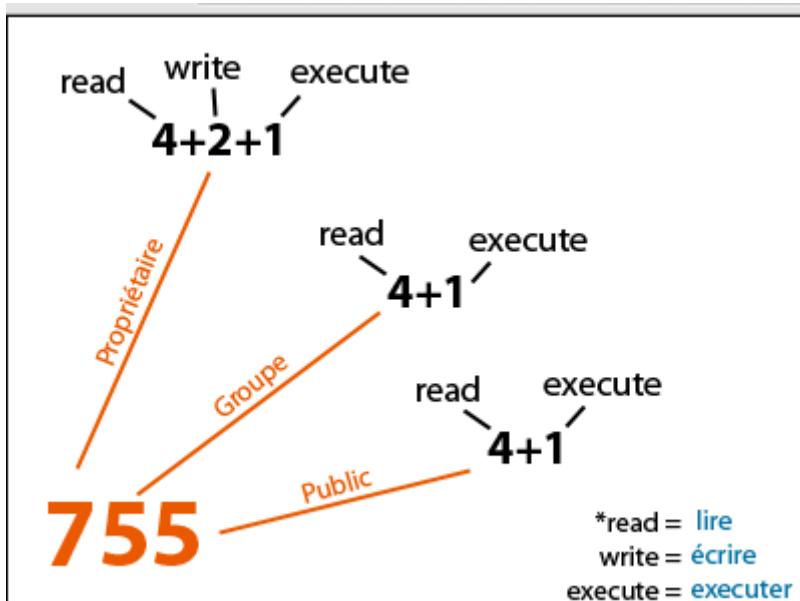
acl utilisateurs proxy_auth REQUIRED
acl lan src 192.168.1.0/24
http_access allow utilisateurs
http_access allow lan

http_access deny !Safe_ports_
http_access deny CONNECT !SSL_ports
```

On effectue ses modifications sur le fichier basic_ncsa_auth :

```
chmod 755 /usr/lib/squid3/basic_ncsa_auth
chown proxy:shadow /usr/lib/squid3/basic_ncsa_auth
```

Le principe du chmod n'est pas compliqué :



De plus, dans chmod 2750, le 2 représente le bit SUID. Si on suit 750, on a tous les droits pour le propriétaire, pas de lecture pour le groupe et aucun droit pour le public.

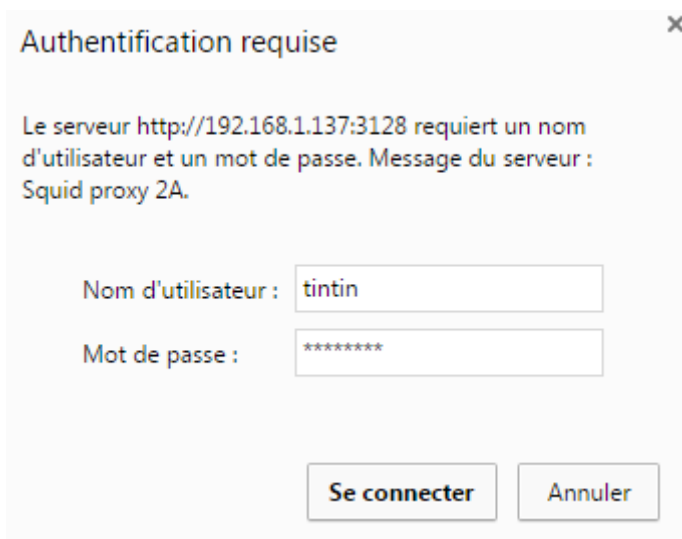
Enfin quand on regarde les droits sur le fichier basic_ncsa_auth, on remarque le 's' pour le droit SUID.

```
root@squid:/etc/squid3# ls -l /usr/lib/squid3/basic_ncsa_auth
-rwxr-s--- 1 proxy shadow 22496 juil. 27 00:04 /usr/lib/squid3/basic_ncsa_auth
```

On peut redémarrer le service squid3 et faire les test dans le navigateur :

```
root@squid:/usr/lib/squid3# /etc/init.d/squid3 reload
[ ok ] Reloading squid3 configuration (via systemctl): squid3.service.
```

Avec tintin par exemple :



Authentication requise

Le serveur http://192.168.1.137:3128 requiert un nom d'utilisateur et un mot de passe. Message du serveur : Squid proxy 2A.

Nom d'utilisateur :

Mot de passe :

NB : Pour une autorisation avec shadow, il faut remplacer la ligne suivante du fichier squid. Conf :

```
auth_param basic program /usr/lib/squid3/basic_ncsa_auth /etc/squid3/squidusers
```

Par :

```
auth_param basic program /usr/lib/squid3/basic_ncsa_auth /etc/shadow
```


V. SquidGuard

Pour utiliser SquidGuard, il faut installer apache2 avant :

```
apt-get install apache2 squidguard
```

Si l'on veut créer notre propre liste noire, il faut créer deux fichiers dans /etc/squid, un qui autorise et l'autre qui bloque. Prenons l'exemple black qui bloque et white qui autorise, dans squid.conf il faut ajouter ces 4 lignes :

```
acl whitelist dstdomain « /etc/squid/white »  
acl blacklist dstdomain « /etc/squid/black »  
http_access allow whitelist  
http_access allow blacklist
```

Cependant, l'université de Toulouse diffuse une liste noire d'URLs afin de permettre un meilleur contrôle de l'utilisation d'Internet. On va récupérer la liste noire :

```
root@squid:/var/lib/squidguard/db# wget http://cri.univ-tlse1.fr/blacklists/download/blacklists.tar.gz _
```

Puis on le décompresse :

```
root@squid:/var/lib/squidguard/db# tar xvzf blacklists.tar.gz
```

A la fin du fichier squid.conf, ajouter ces lignes qui redirigent Squid et le nombre de processus engendré (squid est le nom de ma machine) :

```
url_rewrite_program /usr/bin/squidGuard  
url_rewrite_children 5  
  
visible_hostname squid
```

On se place dans `/etc/squidguard` puis on va créer une copie du fichier de conf :

```
root@squid:/etc/squidguard# cp squidGuard.conf squidGuard.conf.save
```

Maintenant, on va effacer le fichier de conf puis en recréer un :

```
root@squid:/etc/squidguard# rm squidGuard.conf
```

```
GNU nano 2.2.6      Fichier : squidGuard.conf
dbhome /var/lib/squidguard/db/blacklists _
logdir /var/log/squid3/
src lan {
    ip 192.168.1.0-192.168.1.100
}
dest games {
    domainlist games/domains
    urllist games/urls
}
dest local {
}
acl {
    lan {
        pass !games all
        redirect http://127.0.0.1/proxy.html
    }
    default {
        pass local none
    }
}
```

On attribue la propriété de l'ensemble des fichiers de la liste noire à l'utilisateur proxy et au groupe proxy :

```
chown -Rf proxy:proxy /var/lib/squidguard/db
```

On créer maintenant une page html pour faire apparaitre un message d'interdiction :

```
nano /var/www/html/proxy.html
```

On redémarre Squid :

```
root@squid:/etc/squidguard# /etc/init.d/squid3 reload
[ ok ] Reloading squid3 configuration (via systemctl): squid3.service.
```

On test maintenant à une page de jeu, par exemple www.game.fr :



La page qui est demandee est bloquee par Pierre-Marie

Dans le fichier de log, on repère la ligne squidGuard ready for request, signe d'un bon lancement :

```
root@squid:/etc/squidguard# cat /var/log/squid3/squidGuard.log
2015-10-05 10:50:33 [6712] INFO: squidGuard 1.5 started (1444035033.580)
2015-10-05 10:50:33 [6712] INFO: squidGuard ready for requests (1444035033.581)
2015-10-05 10:53:39 [6712] INFO: squidGuard stopped (1444035219.626)
```

VI. Analyseur de log Lightsquid

Il faut installer une librairie avant de pouvoir utiliser Lightsquid, un outil web qui va permettre d'afficher l'usage du proxy :

```
apt-get install libgd-gd2-perl
```

On télécharge lightsquid :

```
root@squid:/var/www/html# wget http://sourceforge.net/projects/lightsquid/files/latest/download?source=files_
```

On renomme le fichier téléchargé :

```
mv download\?source\=files lightSquid.tgz
```

Puis on le décompresse :

```
tar xvzf lightSquid.tgz
```

Enfin, on renomme le dossier qui vient d'être décompressé :

```
mv lightsquid-1.8/ lightsquid
```

Maintenant, on va rendre les scripts pl et cgi exécutable puis changer le propriétaire du répertoire lightsquid par www-data :

```
chmod -R ugo+x lightsquid/*.pl
chmod -R ugo+x lightsquid/*.cgi
chown -R www-data:www-data lightsquid/
```

Maintenant, on va modifier un fichier d'apache dans /etc/apache2/sites-available/000-default.conf, puis insérer ces lignes :

```
<Directory "/var/www/html/lightsquid">
  AddHandler cgi-script.cgi
  AllowOverride All
  DirectoryIndex index.cgi
  Options +ExecCGI
</Directory>
```

Dans lightsquid.cfg, modifier ces lignes :

```
#path to access.log
$logpath                ="/var/log/squid3";
```

```
#language
#see `lang` folder (available: bg,eng,fr,hu,it,pt_br,ru,sp)
$lang                   ="fr";
```

Puis taper la commande ./chek-setup.pl :

```
root@squid:/var/www/html/lightsquid# ./check-setup.pl
LightSquid Config Checker, (c) 2005-9 Sergey Erokhin GNU GPL

LogPath      : /var/log/squid3
reportpath   : /var/www/html/lightsquid/report
Lang         : /var/www/html/lightsquid/lang/fr
Template     : /var/www/html/lightsquid/tpl/base
Ip2Name      : /var/www/html/lightsquid/ip2name/ip2name.simple

all check passed, now try access to cgi part in browser
```

Pas d'erreur, on peut parser le fichier maintenant :

```
root@squid:/var/www/html/lightsquid# ./lightparser.pl
```

On active les modules perl et cgi :

```
a2enmod cgi et a2enmod perl
```

Maintenant, on se rend sur cette adresse :

```
192.168.1.137/lightsquid/
```

Puis une interface web apparait, une boîte à outil à droite, permet de mieux visualiser les sites visités.

[Squid rapport d'accès utilisateur](#)
 Periode de travail: **Oct 2015**

Calendar											
2015											
01	02	03	04	05	06	07	08	09	10	11	12

Top Sites	Total	Groupe
ANNEE	ANNEE	ANNEE
MOIS	MOIS	MOIS

Date	Groupe	Utilisateurs	Quota Dépassé	Octets	Moyenne	Hit %
05 Oct 2015	grp	2	1	15.1 M	7.6 M	0.00%
Total/Moyenne:		2	1	15.1 M	7.6 M	0.00%

[LightSquid v1.8](#) (c) Sergey Erokhin AKA ESL

VII. Configuration d'un navigateur via un script

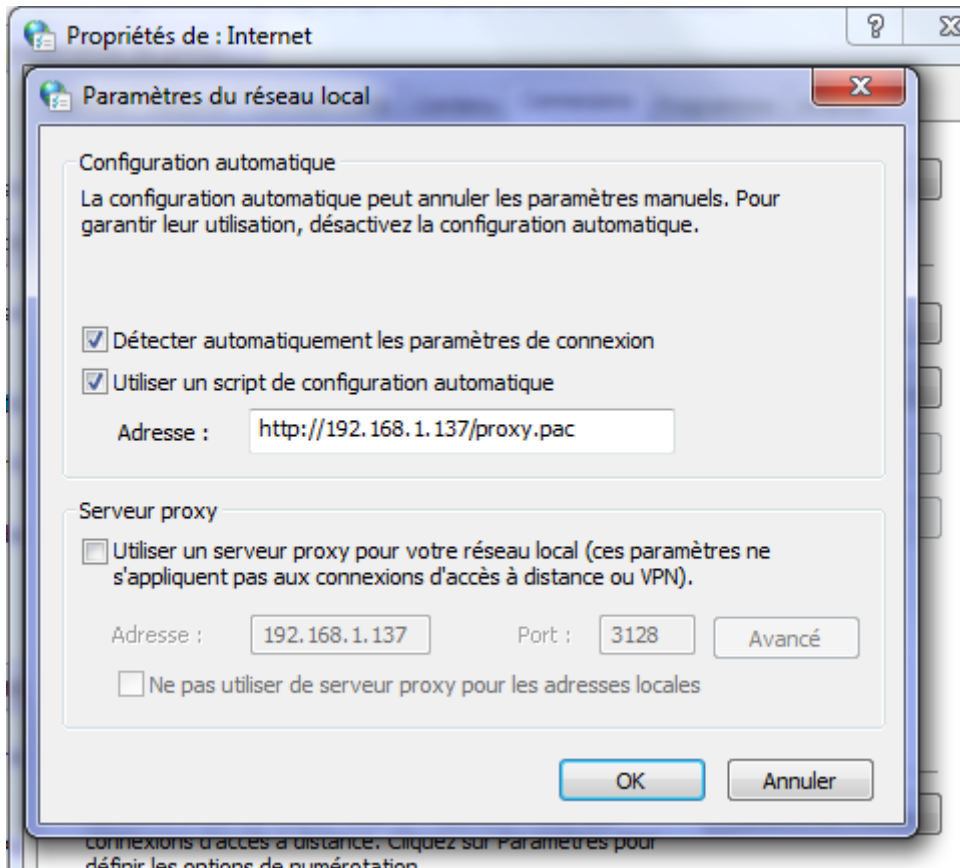
On va créer un fichier script, il s'appelle en général proxy.pac :

```
root@sguid:/var/www/html# nano proxy.pac
```

Puis le remplir comme suit :

```
GNU nano 2.2.6 Fichier : proxy.pac
function FindProxyForURL(url,host)
{
return "PROXY 192.168.1.137:3128;DIRECT";
}
```

Puis dans les paramètres du navigateur, ici Chrome :



Puis, on remarque dans les logs, que je me suis connecté avec milou et que cela a fonctionné grâce au TCP_MISS/200 :

```
1444134608.497 240467 192.168.1.56 TCP_MISS/200 5139 CONNECT www.google.fr:443 m
milou HIER_DIRECT/216.58.211.99 -
1444134615.017 246992 192.168.1.56 TCP_MISS/200 85328 CONNECT www.google.fr:443
milou HIER_DIRECT/216.58.211.99 -
1444134615.828 240670 192.168.1.56 TCP_MISS/200 6329 CONNECT plus.google.com:443
milou HIER_DIRECT/216.58.211.110 -
```

VIII. Annexe :

Squid.conf :

```

auth_param basic program /usr/lib/squid3/basic_ncsa_auth /etc/squid3/squidusers
auth_param basic children 5
auth_param basic realm Squid proxy 2A
authenticate_ttl 1 hour
authenticate_ip_ttl 60 seconds

acl SSL_ports port 443
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 CONNECT method CONNECT

acl allowed_hosts src 192.168.1.12
acl limithour time 16:00-17:30
http_access allow allowed_hosts limithour

acl utilisateurs proxy_auth REQUIRED
acl lan src 192.168.1.0/24
http_access allow utilisateurs
http_access allow lan

http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localhost manager
http_access deny manager
http_access allow localhost
http_access deny all
http_port 3128
coredump_dir /var/spool/squid3
refresh_pattern ^ftp:      1440    20% 10080
refresh_pattern ^gopher:  1440    0%  1440
refresh_pattern -i (/cgi-bin/|\?) 0 0%  0
refresh_pattern .         0       20% 4320

#Ajout des lignes suivantes par moi-même
cache_effective_user proxy
cache_effective_group proxy
cache_mem 16 Mb
cache_dir ufs /var/spool/squid3 120 16 128

url_rewrite_program /usr/bin/squidGuard
url_rewrite_children 5

visible_hostname squid

```

SquidGuard.conf :

```

dbhome /var/lib/squidguard/db/blacklists
logdir /var/log/squid3/
src lan {
    ip 192.168.1.0-192.168.1.100
}

dest games {
    domainlist games/domains
    urllist games/urls
}

dest local {
}

acl {
    lan {
        pass !games all
        redirect http://127.0.0.1/proxy.html
    }
    default {
        pass local none
    }
}

```

000-default.conf :

```

<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    <Directory "/var/www/html/lightsquid">
        AddHandler cgi-script .cgi
        AllowOverride All
        DirectoryIndex index.cgi
        Options +ExecCGI
    </Directory>

    # Available loglevels: trace8, ..., tracel, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

```