DETROZ Sébastien Version 2 01/10/2016

# Sommaire :



# Introduction

Objectif : L'objectif de cette procédure est de réaliser l'installation et à la configuration d'un serveur proxy. Le proxy fonctionne comme un cache pour le réseau : c'est un point de sorti unique vers l'extérieur. Il est d'abord mandataire, c'est à dire qu'il va pouvoir faire des requêtes sur Internet à la place de l'hôte. Le serveur proxy fait également office de cache et mémorise les pages les plus visitées. On peut aussi mettre en place un filtrage et faire, par exemple, des ACL pour interdire la visite de certains sites, ainsi que des black-list par adresses IP ou par mot-clés. Un proxy peut enfin mémoriser les identifiants en fonction des URL.

Pré-requis : Nous utiliserons un système d'exploitation Debian 8.5 pour la réalisation de cette procédure. Il ne faut pas oublier de mettre à jour les fichiers de gestion de dépôts de notre machine Debian afin de pouvoir télécharger tous les paquets nécessaires à la réalisation de cette procédure. Dans le fichier ***/etc/apt/sources.list***, il faut donc rajouter les lignes suivantes :

***deb http://ftp.fr.debian.org/debian/ jessie main***
***deb http://ftp.fr.debian.org/debian/ jessie-updates main***

Et avant de commencer, il est important de lancer la commande ***apt update***.

Norme adoptée : Tous les noms et les commandes issus ou utilisés sur le système d'exploitation Debian seront écrits en ***gras et en italique***.

# 1 – Installation de Squid 3 :

Après avoir mis à jour les fichiers de gestion de dépôts de Debian, et mis à jour le système lui-même, nous pouvons procéder au téléchargement du logiciel Squid 3 avec la commande ***apt install squid3***. Lors de cette étape, il vous sera demandé d'insérer le CD d'installation de l'OS pour démarrer le téléchargement :

```
root@debiansquid:~# apt-get install squid3
```

```
« Debian GNU/Linux 8.6.0 _Jessie_ - Official amd64 CD Binary-1 20160917-14:25 »
dans le lecteur « /media/cdrom/ » et appuyez sur la touche Entrée
```

Par défaut, le port d'écoute de Squid 3 est le port 3128. Nous pouvons vérifier cette information grâce à la commande ***netstat -ltp*** :

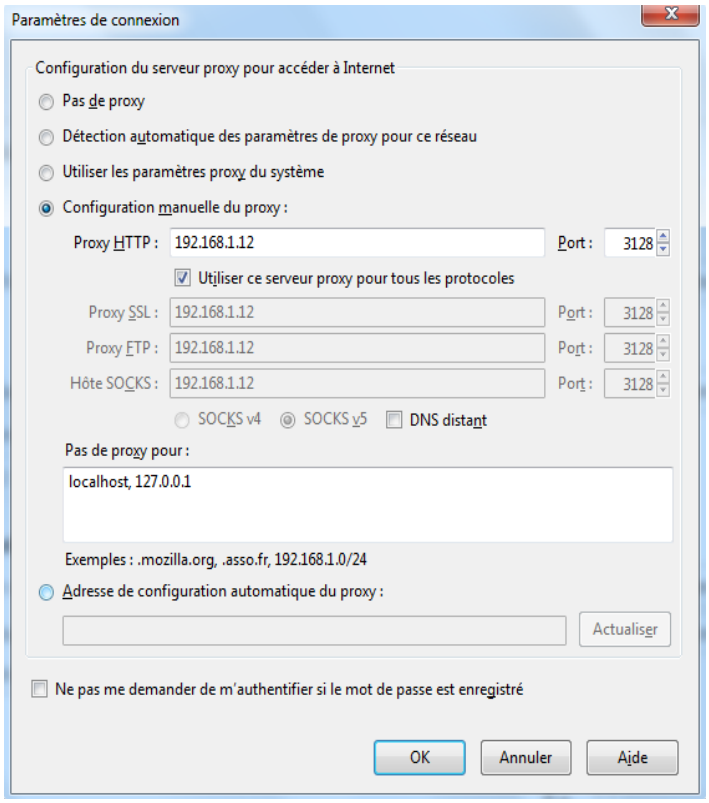
```
root@debiansquid:~# netstat -ltp
Connexions Internet actives (seulement serveurs)
Proto Recv-Q Send-Q Adresse locale          Adresse distante        Etat
PID/Program name
tcp        0      0 localhost:smtp          *:*                     LISTEN
738/exim4
tcp        0      0 *:42789                 *:*                     LISTEN
456/rpc.statd
tcp        0      0 *:sunrpc                *:*                     LISTEN
447/rpcbind
tcp6       0      0 [::]:3128               [::]:*                  LISTEN
1275/(squid-1)
```

Lors de l'installation, l'utilisateur proxy ainsi que le groupe proxy ont été ajoutés. Il est possible de vérifier cela avec les commandes ***cat /etc/passwd*** ou ***cat /etc/group*** *:*

```
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
```

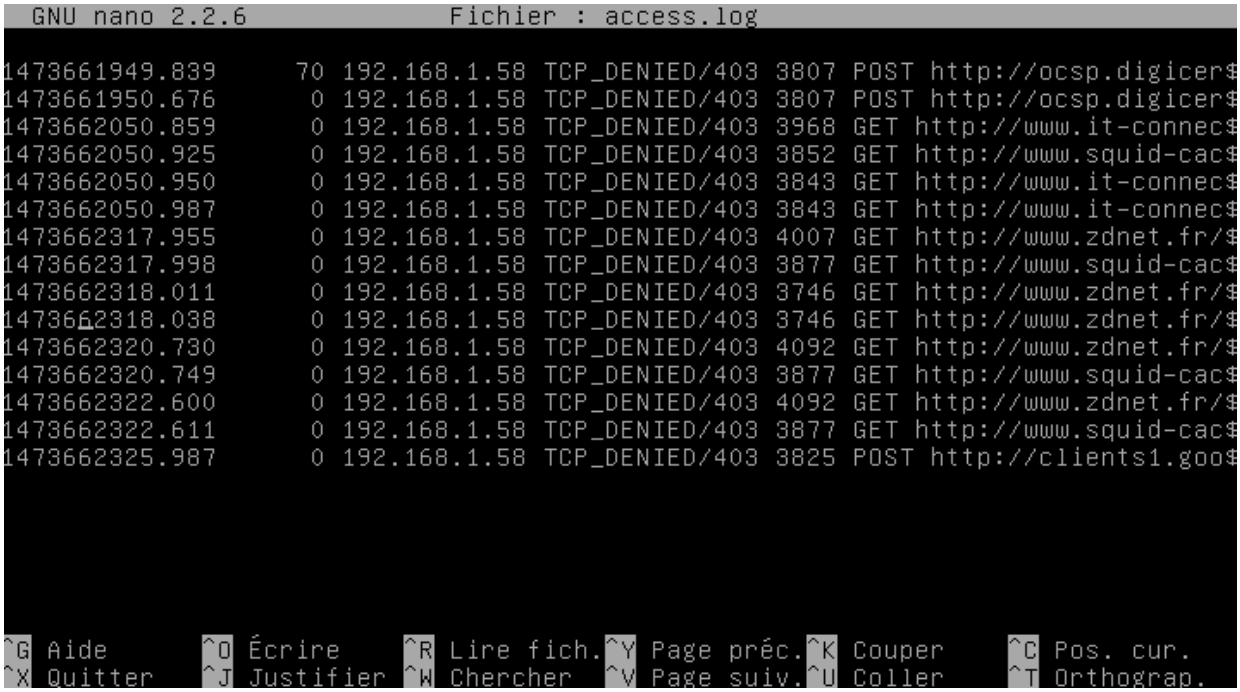
# 2 – Configuration de base :

Nous allons ensuite procéder à la configuration du proxy sur un navigateur Firefox. Pour cela, il faut accéder à ***Options > Avancé > Réseau > Paramètres*** pour entrer les informations relatives à notre serveur proxy :




Une fois le proxy paramétré, nous obtenons le message suivant lorsque l'on essaye de se connecter à Internet :

Lorsque la connexion est refusée, des lignes de ce type apparaissent dans les fichiers de logs, visibles grâce à la commande ***tail /var/log/squid3/access.log*** :




Afin de pouvoir mieux paramétrer notre serveur Squid 3, et sachant que son fichier de configuration comporte environ 5000 lignes, nous allons expurger, c'est à dire exclure tous les commentaires et ne laisser que les commandes importantes, le fichier ***squid.conf.sauv*** (une sauvegarde du fichier initial) grâce à la commande ***cat squid.conf.sauv | grep -v# | grep -v ^$ > squid.conf***. Nous obtenons finalement un fichier de configuration lisible que nous pouvons commencer à modifier en ajoutant les lignes suivantes :

```
# Utilisateur faisant les requêtes sur le serveur
cache_effective_user proxy
cache_effective_user group

cache_mem 16 MB
cache_dir ufs /var/spool/squid3 120 16 128
visible_hostname sebproxy
```

La ligne ***cache_dir ufs /var/spool/squid3 120 16 128*** permet de créer un répertoire de cache selon la taille indiquée. Son fonctionnement est expliqué dans le fichier de configuration initial :

```
====  The ufs store type  ====

"ufs" is the old well-known Squid storage format that has always
been there.

Usage:
        cache_dir ufs Directory-Name Mbytes L1 L2 [options]

'Mbytes' is the amount of disk space (MB) to use under this
directory.  The default is 100 MB.  Change this to suit your
configuration.  Do NOT put the size of your disk drive here.
Instead, if you want Squid to use the entire disk drive,
subtract 20% and use that value.

'L1' is the number of first-level subdirectories which
will be created under the 'Directory'.  The default is 16.
```

Lorsque le proxy interdit l'accès Internet, l'on obtient à présent la page suivante :

**ERREUR**

**L'URL demandée n'a pas pu être trouvé**

L'erreur suivante s'est produite en essayant d'accéder à l'URL : http://www.zdnet.fr/actualites/google-va-inaugurer-une-statue-android-nougat-a-montelimar-39841742.htm

**Accès interdit.**

La configuration du contrôle d'accès, empêche votre requête d'être acceptée. Si vous pensez que c'est une erreur, contactez votre fournisseur d'accès.

Votre administrateur proxy est webmaster.

Générée le Mon, 12 Sep 2016 06:40:30 GMT par debianseb1 (squid/3.4.8)

Le nom de le hôte, et donc du serveur, refusant la connexion est indiquée tout en bas.

# 3 – Les contrôles d'accès :

Nous allons maintenant mettre en place un contrôle de l'accès Internet pour certaines recherches en particulier. Pour cela, nous allons utiliser les ACL. Sur Squid 3, les ACL permettent de définir des conditions sur les adresses IP, les ports, le contenu de certains textes, etc. Afin de vérifier si notre système d'exploitation prend en charge les ACL, nous utilisons la commande ***cat /boot/fichier_de_conf_de_notre_version | grep ACL*** :

```
root@debiansquid:/boot# cat 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
```

Si toutes les lignes sont positionnés sur le « ***y*** » de « ***yes*** », alors les ACL sont pris en charge. Nous allons ensuite procéder à création d'une ACL n'autorisant qu'une plage d'adresses IP à surfer. Pour cela, il faut accéder au fichier ***/etc/squid3/squid.conf*** et y insérer les lignes :

***acl LAN src 192.168.1.0 /24***
***http_access allow LAN***

```
acl LAN src 192.168.1.0/24
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

# Ajout du droit AU DESSUS des autres http_access

http_access allow LAN
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localhost manager
```

La première ligne permet de définir une ACL, qui va rechercher toutes les adresses IP de 192.168.1.0 à 192.168.1.255. Puis la seconde ligne va appliquer cette ACL, en autorisant donc toutes les adresses recherchées par l'ACL à utiliser Internet. Pour résumer, les lignes commençant par « ***acl*** » définissent les listes d'autorisation, tandis que celles qui commencent par « ***http_access*** » activent les ACL, en les autorisant, ou en les interdisant.

Nous pouvons également mettre en place des restrictions horaires, avec l'ACL suivante :

***acl horaire time 16:00-17:30***
***http_access allow horaire***

```
acl LAN src 192.168.1.0 /24
acl horaire time 16:00-17:30
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

http_access allow LAN
http_access allow horaire_
```

Ici, c'est l'argument « ***time*** » qui va permettre d'utiliser des heures sous un certain format. L'argument précédent, « ***src*** » permettait quand à lui la recherche d'une plage d'adresses IP.

# 4 – Authentification des utilisateurs :

Le principe de surveillance des accès ne vaut que dans la mesure d'une trace de la connexion avec une identification de l'utilisateur qui surf sur notre proxy. C'est pourquoi Squid 3 permet l'authentification des utilisateurs, afin de conserver une trace des sites que chacun visites. Pour mettre en place l'authentification, nous allons créer des utilisateurs avec les commandes suivantes (nous aurons besoin d'un module d'Apache 2) :

***apt install apache2-utils***

***touch /etc/squid3/squidusers***

***htpasswd -b /etc/squid3/squidusers tintin reporter***

***htpasswd -b /etc/squid3/squidusers milou chien***

Les utilisateurs tintin et milou, avec leurs mots de passes respectifs, ont donc été créés. Il est maintenant nécessaire de modifier le fichier /etc/squid3/squid.conf afin d'y ajouter les options permettant d'utiliser notre module d'identification, ainsi que l'ACL permettant de gérer l'authentification lors de la connexion sur un navigateur. Voici les modifications à apporter sur l'image suivante :

```
  GNU nano 2.2.6            Fichier : squid.conf                    Modifié

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 utilisateurs proxy_auth REQUIRED
acl LAN src 192.168.1.0 /24
acl horaire time 16:00-17:30
acl SSL_ports port 443
```

Les options après ***authenticate*** permettent de gérer le ***time to live*** de la validité de la session de l'utilisateur. Lorsque le temps sera écoulé, le navigateur demandera à nouveau à l'utilisateur de s'identifier pour continuer à surfer. Il faut également penser à activer l'ACL « ***utilisateurs*** » :

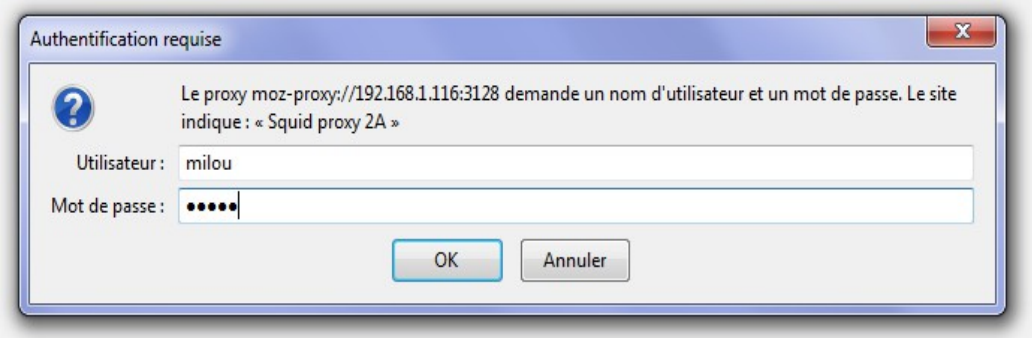
```
http_access allow utilisateurs
http_access allow LAN
http_access allow horaire_
http_access deny !Safe_ports
```

Finalement, afin de mettre en place ce système d'authentification, il faut ajouter des droits d'accès à l'utilisateur proxy vers certains fichiers de configuration de Squid 3, avec les commandes suivantes :

***chown proxy:shadow /usr/lib/squid3/basic_ncsa_auth***

***chmod 2750 /usr/lib/squid3/basic_ncsa_auth***

Lorsque tout est correctement paramétré, nous lançons un navigateur, et l'identification de l'utilisateur est bien requise :



# 5 – SquidGuard :

Nous allons maintenant installer et configurer le logiciel SquidGuard. Ce dernier permet de mettre en place des fonctions avancées de filtrage, notamment la mise en place de Blacklists, des listes de sites interdits à la navigation. Nous allons par exemple utiliser une blacklist régulièrement mise à jour par l'université de Toulouse. Pour cela, il faut donc commencer par installer SquidGuard, avec la commande ***apt install squidguard***.

Pour commencer, nous pouvons créer nos propres blacklists et whitelists, pour interdire ou autoriser certains sites. Nous créons donc deux fichiers ***black*** et ***white*** dans le répertoire ***/etc/squid3*** :

```
root@debiansquid:/etc/squid3# ls
black  errorpage.css  msntauth.conf  squid.conf  squid.conf.sauv  white
```

Avec, par exemple, dans le fichier black :

```
  GNU nano 2.2.6                Fichier : black

www.elysee.fr
```

Nous modifions ensuite le fichier ***/etc/squid3/squid.conf*** afin d'y ajouter les ACL requises :

```
acl whitelist dstdomain "/etc/squid3/white"
acl blacklist dstdomain "/etc/squid3/black"
#acl utilisateurs proxy_auth REQUIRED
#acl LAN src 192.168.1.0 /24
#acl horaire time 16:00-17:30

http_access deny blacklist
http_access allow whitelist
#http_access allow utilisateurs
#http_access allow LAN
#http_access allow horaire
```

Pour les blacklists, il est impératif de :

- Saisir les URL sous cette forme : www.google.fr
- Activer la blacklist en début d'ACL : ***http_access deny blacklist***

La connexion vers le site blacklisté est bien interdite :


Nous allons maintenant utiliser la blacklist de l'Université de Toulouse. Pour la télécharger, nous utilisons l'outil ***wget*** qui permet de télécharger le fichier sous forme d'archive via un URL : ***http://cri.univ-tlse1.fr/blacklists/download/blacklists.tar.gz***

```
  -I,  --include-directories=LISTE liste des répertoires permis.
       --trust-server-names        use the name specified by the redirection
                                   url last component.
  -X,  --exclude-directories=LISTE liste des répertoires exclus.
  -np, --no-parent                 ne pas remonter dans le répertoire parent.

Veuillez signaler toutes anomalies ou suggestions à <bug-wget@gnu.org>.
root@debianseb1:/etc/squid3# wget http://cri.univ-tlse1.fr/blacklists/download/b
lacklists.tar.gz
--2016-09-13 09:45:36--  http://cri.univ-tlse1.fr/blacklists/download/blacklists
.tar.gz
Résolution de cri.univ-tlse1.fr (cri.univ-tlse1.fr)… 193.49.48.249
Connexion à cri.univ-tlse1.fr (cri.univ-tlse1.fr)|193.49.48.249|:80… connecté.
requête HTTP transmise, en attente de la réponse… 302 Found
Emplacement : http://dsi.ut-capitole.fr/blacklists/download/blacklists.tar.gz [s
uivant]
--2016-09-13 09:45:36--  http://dsi.ut-capitole.fr/blacklists/download/blacklist
s.tar.gz
Résolution de dsi.ut-capitole.fr (dsi.ut-capitole.fr)… 193.49.48.249
Réutilisation de la connexion existante à cri.univ-tlse1.fr:80.
requête HTTP transmise, en attente de la réponse… 200 OK
Taille : 8403519 (8,0M) [application/x-gzip]
Sauvegarde en : « blacklists.tar.gz »

blacklists.tar.gz    39%[=======>             ]   3,19M   334KB/s   eta 66s
```

Ensuite, nous déplaçons l'archive vers le répertoire ***/var/lib/squidguard/db***, puis nous la décompressons avec la commande ***tar xzvf blacklists.tar.gz***. Nous obtenons finalement tous les fichiers suivants, qui répertorient des centaines d'URL interdit à la navigation sur notre proxy :

```
blacklists/download/urls
blacklists/download/usage
blacklists/ddos/
blacklists/ddos/domains
blacklists/ddos/usage
blacklists/update/
blacklists/update/domains
blacklists/update/usage
blacklists/associations_religieuses/
blacklists/associations_religieuses/domains
blacklists/associations_religieuses/usage
blacklists/shortener/
blacklists/shortener/domains
blacklists/shortener/urls
blacklists/shortener/usage
blacklists/aggressive
blacklists/mail
blacklists/violence
blacklists/ads
blacklists/drugs
blacklists/porn
blacklists/proxy
root@debianseb1:/var/lib/squidguard/db# ls
blacklists  blacklists.tar.gz
```

Afin de rediriger ***Squid 3*** vers ***Squid Guard***, qui lui même entraînera une redirection vers une page HTML présente dans notre répertoire ***Apache 2***, il faut ajouter une redirection dans le fichier **/etc/squid3/squid.conf** :

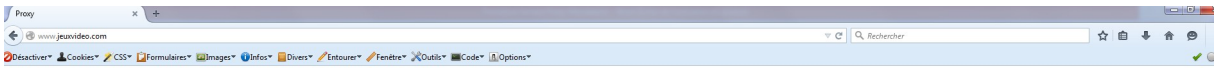
```
# Redirection de Squid3 vers SquidGuard :

url_rewrite_program /usr/bin/squidguard
url_rewrite_children 5_
```

Ensuite, il nous reste à créer le fichier de configuration ***/etc/squid/squidguard.conf*** afin de définir le réseau utilisé, ainsi que les restrictions par les ACL à activer ou non. Pour cet exemple, nous allons interdire les sites de jeux vidéos présents dans la liste « ***games*** » de la blacklist de l'Université de Toulouse . Sur l'image suivante, l'option « ***!games all*** » permet d'interdire la blacklist « ***games*** », grâce au point d'exclamation, et « ***redirect http://192.168.1.116/proxy.html*** » permet de rediriger le proxy vers une page HTML présente dans le répertoire ***/var/www/html*** :

```
  GNU nano 2.2.6           Fichier : squidguard.conf

dbhome /var/lib/squidguard/db/blacklists
logdir /var/log/squid3

src lan {
        ip 192.168.1.1-192.168.1.254
}

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

acl {
        lan {
                pass !games all
                redirect http://192.168.1.116/proxy.html
        }
}_
```

Enfin, il ne reste plus qu'à reconstruire la base de la blacklist dans ***SquidGuard***, afin qu'il puisse l'utiliser, avec la commande : ***squidGuard -C all -d /var/lib/squidguard/db***. Puis finalement, donner les droits d'accès sur la blacklist ainsi présente dans ***SquidGuard***, à l'utilisateur et au groupe ***proxy*** avec : ***chown -Rf proxy.proxy /var/lib/squidguard/db***.

Après toutes les manipulations de cette procédure, la connexion à ***www.games.fr*** ne fonctionne pas, comme voulu, et la page est redirigée vers ***http://192.168.1.116/proxy.html*** :




# NON

Dans les logs, on peut voir que ***SquidGuard*** fonctionne correctement, et qu'il « attend des requêtes » :

```
root@debianseb1:/var# tail /var/log/squid3/squidGuard.log
2016-09-13 11:06:25 [1306] INFO: squidGuard 1.5 started (1473757585.678)
2016-09-13 11:06:25 [1306] INFO: squidGuard ready for requests (1473757585.700)
```

L'argument ***-f*** de la commande ***tail –f /var/log/squid3/access.log*** ou ***tail -f /var/log/squid3/squidGuard.log***, permet de vérifier le fonctionnement de notre proxy en direct, au fur et à mesure de la navigation.

# 6 – Analyseur de log Lightsquid :

Afin de mieux suivre la navigation de nos utilisateurs sur notre serveur proxy, nous pouvons utiliser un analyseur de log, nommé ***Lightsquid***. C'est un analyseur de log ***Squid*** open source écrit en perl permettant d'afficher sous forme de page web l'usage du proxy. Pour l'installer, nous devons d'abord télécharger la librairie ***Apache 2*** utilisant perl avec ***apt-get install libgd-gd2-perl***.

Puis nous téléchargeons le fichier ***tar.gz*** de ***Lightsquid*** sur le site officiel, avec la commande ***wget http://downloads.sourceforge.net/project/lightsquid/lightsquid/1.8/lightsquid-1.8.tgz?r=&ts=1474026944&use_mirror=freefr*** dans le répertoire ***/var/www/html***. Nous décompressons ensuite le fichier avec ***tar xzvf lightsquid-1.8.tar.gz*** :

```
root@debianseb1:/var/www/html/lightsquid-1.8# ls
bigfiles.cgi    group.cfg.src     month_detail.cgi  user_detail.cgi
check-setup.pl  group_detail.cgi  realname.cfg      user_month.cgi
common.pl       index.cgi         report            user_time.cgi
day_detail.cgi  ip2name           skipuser.cfg      whousesite.cgi
doc             lang              tools
get.cgi         lightparser.pl    topsites.cgi
graph.cgi       lightsquid.cfg    tpl
```

Les fichiers ***pl*** et ***cgi*** doivent être exécutables par la machine, nous utilisons donc la commande ***chmod ugo+x *.cgi *.pl*** pour cela. Ensuite, un ***chown www-data lightsquid*** permet de donner les droits d'accès au service ***Apache 2*** sur le logiciel ***Lightsquid*** :

```
root@debianseb1:/var/www/html# ls -l
total 80
-rw-r--r-- 1 root     root  11104 sept. 13 11:20 index.html
drwxrwxr-x 8 www-data staff  4096 juil.  2  2009 lightsquid
-rw-r--r-- 1 root     root  60868 juil.  5  2009 lightsquid-1.8.tgz?r=
-rw-r--r-- 1 root     root    191 sept. 13 11:26 proxy.html
```

Il est nécessaire de configurer ***Apache 2*** dans un fichier à l'emplacement ***/etc/apache2/sites-available*** pour permettre la création de la page ***lightsquid*** grâce aux scripts ***cgi*** :

```
  GNU nano 2.2.6                Fichier : lightsquid

<Directory "/var/www/html/lightsquid">
        AddHandler cgi_script .cgi
        AllowOverride All
        DirectoryIndex index.cgi
        Options +ExecCHI
</Directory>
```

Il nous reste encore à le module ***cgi*** d'***Apache 2*** afin de pouvoir exécuter les scripts précédents avec la commande : ***a2enmod cgi***. Nous modifions ensuite le fichier ***lightsquid.cfg*** dans le dossier ***/var/www/html/lightsquid*** pour indiquer le bon chemin vers les fichiers de log à analyser :

```
  GNU nano 2.2.6              Fichier : lightsquid.cfg

# -------------------- GLOBAL VARIABLES  ---------------------------

#path to additional `cfg` files
$cfgpath              ="/var/www/html/lightsquid";
#path to `tpl` folder
$tplpath              ="/var/www/html/lightsquid/tpl";
#path to `lang` folder
$langpath             ="/var/www/html/lightsquid/lang";
#path to `report` folder
$reportpath           ="/var/www/html/lightsquid/report";
#path to access.log
$logpath              ="/var/log/squid3";
#path to `ip2name` folder
$ip2namepath          ="/var/www/html/lightsquid/ip2name";

#path to `lockfile` ;-)
$lockpath             =$reportpath;

#if lockfile older $maxlocktime second, remove old lock file.
$maxlocktime          = 30*60;

^G Aide      ^O Écrire    ^R Lire fich.^Y Page préc.^K Couper    ^C Pos. cur.
^X Quitter   ^J Justifier ^W Chercher  ^V Page suiv.^U Coller    ^T Orthograp.
```

Finalement, nous testons l'installation et la configuration de Lightsquid avec la commande ***./check-setup.pl*** :

```
root@debianseb1:/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
```

Puis ***./lightparser.pl***, qui ne doit pas renvoyer de réponse négative :

```
root@debianseb1:/var/www/html/lightsquid# ./lightparser.pl
```

Enfin, lorsque nous accédons à l'adresse ***http://192.168.1.116/lightsquid*** :

**Squid rapport d'accès utilisateur**
**Periode de travail: Sep 2016**

| Calendar | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2016 | | | | | | | | | | | |
| 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 % |
|---|---|---|---|---|---|---|
| 16 Sep 2016 | grp | 2 | 0 | 2.9 M | 1.4 M | 0.04% |
| 13 Sep 2016 | grp | 2 | 1 | 27.9 M | 14.0 M | 0.00% |
| 12 Sep 2016 | grp | 3 | 1 | 12.6 M | 4.2 M | 0.00% |
| Total/Moyenne: | | 2 | 0 | 43.4 M | 6.5 M | 0.01% |

LightSquid v1.8 (c) Sergey Erokhin AKA ESL

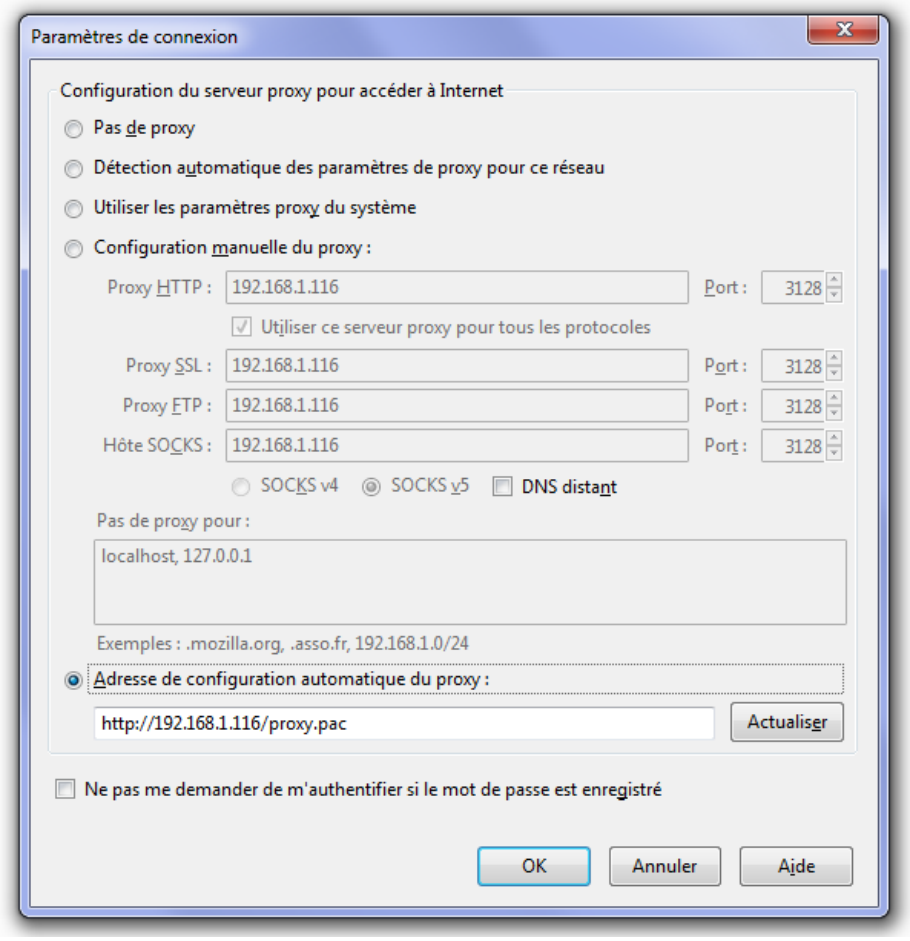
Depuis cette interface, nous pouvons maintenant suivre la navigation de nos différents utilisateurs (ici, tintin et milou), leurs sites les plus visités, leurs recherches Internet, etc.

# 7 – Configuration d'un navigateur via un script :

Pour paramétrer plus rapidement son navigateur à l'utilisation d'un serveur proxy, nous pouvons créer un script, comme le suivant :

```
  GNU nano 2.2.6                Fichier : proxy.pac

function FindProxyForURL(url,host)
{
return "PROXY 192.168.1.116:3128;DIRECT";
}
```

Sur notre navigateur, en cochant l'option adéquate et en indiquant l'URL de notre script, le navigateur va pouvoir récupérer les informations du serveur proxy, et ainsi le mettre en application.



# 8 – Configuration d'un proxy transparent :

Avec l'utilisateur du WPAD (Web Proxy Auto Discovery Protocol), il est possible de créer un serveur de proxy transparent sur le réseau. Tous les navigateurs détecteront automatiquement le proxy et l'utiliseront, sans que l'utilisateur n'en sache rien. Mais l'application de ce genre de serveur fera l'objet d'une prochaine procédure.

# Conclusion :

Afin de suivre et de contrôler la navigation Internet des utilisateurs d'une entreprise, par exemple, il existe une multitude de solutions pour mettre en place un serveur proxy. La solution vue dans cette procédure reste assez simple de configuration mais peut être limitée. Les utilisateurs peuvent notamment facilement désactiver le proxy. Il serait donc intéressant de mettre en place un serveur plus transparent, ou contrôlant la navigation directement sur les routeurs.