Squid - SquidGuard

Master 2 I2L

Année 2009/2010

D. Duvivier
LIL – Université du Littoral Côte d'Opale
duvivier@lil.univ-littoral.fr

1 Préparer son serveur

```
aptitude update
aptitude upgrade
aptitude clean
```

Attention : Ne rendez visible que le strict nécessaire sur votre machine, protégez en mode 700 /root, $home et /usr/src/…

2 Installer squid

```
aptitude install squid
cp /etc/squid/squid.conf /etc/squid/squid.conf.orig
emacs /etc/squid/squid.conf
```

Le fichier /etc/squid/squid.conf sera modifié de manière à contenir les lignes suivantes (attention la plupart des lignes sont déjà présentes (pas forcément dans le même ordre) dans le fichier de configuration.

Si vous voulez obtenir rapidement un fichier similaire au fichier situé ci-dessous si toutefois vous avez pris la précaution de créer le fichier squid.conf.orig (en fait on enlève les lignes vides et les commentaires) :

```
egrep -v '^#|^$' /etc/squid/squid.conf.orig >/etc/squid/squid.conf
```

La plupart des lignes de ce fichier sont expliquées en cours…

```
# Recommended minimum configuration: → ce qui est situé avant est du commentaire !

# Liste d'ACL par défaut consitutée de 4 lignes, selon les versions de squid, vous aurez la liste présentée en cours
# (commentée ci-dessous)
# acl all src 0.0.0.0/0.0.0.0
# acl manager proto cache_object
# acl localhost src 127.0.0.1/255.255.255.255
# acl to_localhost dst 127.0.0.0/8
# Ou la liste suivante, garder la liste par défaut issue de VOTRE version de squid
# (on définit notamment « qui est considéré comme localhost » via une acl…) :
acl all src all
acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32

# Une liste d'ACL qui si elle est détectée fera que la requête ne sera pas acceptée
acl QUERY urlpath_regex cgi-bin \?
# Si on utilise DENY, le nom de l'ACL ne sera pas mis dans le cache
no_cache deny QUERY

# Pour un réseau interne de très grande taille (classe A par défaut) utilisez ceci :
# acl localnet src 10.0.0.0/8       # RFC1918 possible internal network
# Pour un réseau interne de grande taille (classe B par défaut) utilisez ceci :
# acl localnet src 172.16.0.0/12  # RFC1918 possible internal network
# Pour un réseau interne de petite taille (classe C) utilisez ceci :
# acl localnet src 192.168.0.0/16 # RFC1918 possible internal network
# Nous définissons le/les réseau(x) utilisant le cache via une/des ACL
# Ceci permet de définir des groupes afin de leurs attribuer des droits différents.
# Cette (ou ces lignes) définit/définissent les adresses IP des utilisateurs du réseau
# On doit personnaliser selon sa configuration réseau, pour ma configuration cela donne :
acl ReseauI2L src 172.16.0.0/255.255.255.0
# acl ReseauDD src 192.168.0.0/255.255.255.0
# acl ReseauSecondaire src 192.168.2.0/255.255.255.0
```

```
# A l'aide des lignes suivantes, définissez les ports auxquels certains utilisateurs auront droit d'accès
# ici ils auront accès à http (port 80), https (port 443), ftp (port 21)
# On peut facilement interdire les accès en commentant ou supprimant les lignes correspondantes
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

# Taille maximum de mémoire vive utilisée pour stocker du cache
cache_mem 16 MB
# Taille maximum des objets stockés dans le cache
maximum_object_size 15 MB

# Chemin des fichiers/répertoires de cache
# on peut définir plusieurs répertoires, si possible sur des disques différents afin de gagner en vitesse
cache_dir ufs /var/spool/squid 100 16 256

# Placé sur off, squid utilise son propre format de logs, mais la date et l'heure ne sont pas lisibles
# mais compatible avec les outils d'analyse de log pour squid
emulate_httpd_log off

# Les deux lignes suivantes permettent d'intégrer le plugin SquidGuard, attention au « G » majuscule !
redirect_program /usr/bin/squidGuard
redirect_children 4

# Définition des accès …
# Attention Les règles sont lues dans l'ordre et s'arrête à la première interdiction correspondant au critère
# de l'utilisateur, un peu comme un firewall, l'ordre des règles est donc important !
# Par défaut squid ne « nous » autorise pas à purger des éléments, sauf si nous ajoutons des acl pour le faire,
# comme ci-dessous…
# On définit l'acl pour purger, on définit l'acl pour se connecter, on autorise le cache manager à accéder au cache
# (pour faire des statistiques via des outils ou…) depuis localhost ; comme les règles/acl sont lues dans l'ordre,
# si le cache manager est sur localhost, l'acl « http_access allow manager localhost » est validée, l'accès est validé
# et on arrête la lecture des acl, sinon (cache manager pas sur localhost ou autre cas)
# l'acl « http_access deny manager » s'applique et on se fait jeter !
# Les deux acl suivantes font la même chose avec « purge » : autorisée en localhost, refusée sinon !
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
```

```
# L'acl « http_access deny !Safe_ports » interdit l'accès à tous les ports sauf ceux définis par les acl Safe_ports
http_access deny !Safe_ports
# L'acl suivante interdit la connexion à tous les ports sauf ceux définis par les acl SSL_ports
http_access deny CONNECT !SSL_ports

# On donne ensuite accès au WEB au localhost et aux réseaux « ReseauI2L, ReseauDD et ReseauSecondaire »
# définis précédemment
# (→ pour tester, les 3 sous-réseaux « ReseauI2L » « ReseauDD » et « ReseauSecondaire » accèdent à http)
http_access allow localhost
http_access allow ReseauI2L
#http_access allow ReseauDD
#http_access allow ReseauSecondaire

# Maintenant L'ACL LA PLUS IMPORTANTE :
# On interdit le reste, important car sinon notre proxy pourra être utilisé par n'importe qui via internet
http_access deny all

# ATTENTION : pour l'instant les 3 sous-réseaux (ReseauI2L, ReseauDD et ReseauSecondaire)
# accèdent directement au proxy ☹
icp_access allow ReseauI2L
#icp_access allow ReseauDD
#icp_access allow ReseauSecondaire
icp_access deny all

# Il peut être utile d'ajouter ceci (voir la documentation pour les détails) il suffit de décommenter
# (par défaut c'est positionné sur http_reply_access allow all) :
#http_reply_access allow ReseauI2L
#http_reply_access allow ReseauDD
#http_reply_access allow ReseauSecondaire
#http_reply_access deny all

# Liste de mots qui, si elle trouvée dans l'url, sera prise en charge directement par le cache
hierarchy_stoplist cgi-bin ?

# Gestion des ports ICP (utilisé par squid pour échanger des infos si nécessaire avec d'autres caches).
# Dans notre cas, nous avons un seul cache, donc seulement autorisé en local (ICP_port = 3130 laissé par défaut).
# Il reste à préciser le port http pour le proxy : 3128
# Nous précisons qu'il s'agit d'un proxy transparent :
http_port 3128 transparent

# Indiquer les DNS : ATTENTION VERIFIEZ AVEC VOTRE CONFIGURATION
# via nslookup puis le mot clef « server » ou …
# Il faut décommenter l'une des deux lignes suivantes (cf cours) :
# dns_nameservers 195.220.130.2 192.220.130.10
dns_nameservers 172.16.0.1

# On définit ensuite l'host de notre proxy et le mail de l'administrateur
# (ATTENTION il faut adapter à votre configuration) :
# Par exemple « visible_hostname monproxy.domain » devient
visible_hostname i2lproxy
# Par exemple « cache_mgr webmaster@domain » devient
cache_mgr root
```

```
# Les lignes suivantes étaient présentes par défaut, je les laisse en l'état
# (il s'agit de fréquences de rafraîchissement du cache (...)) :
access_log /var/log/squid/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 apache rep_header Server ^Apache
broken_vary_encoding allow apache
extension_methods REPORT MERGE MKACTIVITY CHECKOUT
hosts_file /etc/hosts
coredump_dir /var/spool/squid
```

Pour relancer squid :

```
/etc/init.d/squid restart
```

3 Quelques outils et commandes utiles

```
# dnsdomainname –v → retourne le nom de domaine
# hostname       → nom de machine
# hostname –i    → adresse ip de la machine (voir ifconfig si plusieurs cartes réseau)
# hostname –f    → nom complet (Fully Qualified Domain Name) de la machine
# nslookup
> server
> exit           → retourne le DNS par défaut
# /etc/init.d/squid status → indique si squid fonctionne ou non
```

Remarque : si vous voulez que le proxy « i2lproxy » soit situé sur le réseau 172.16.0.1 mis en place lors des précédents TP et que son nom complet (FQDN) soit i2lproxy.m2i2l.org, il faut ajouter la ligne suivante à au fichier de configuration « perso » du DNS (/etc/bind/db.m2i2l.org) :

```
1    IN    PTR        ns1.m2i2l.org.
```

En ce cas, vous pouvez écrire ceci dans le fichier de configuration de squid :

```
visible_hostname i2lproxy.m2i2l.org
```

Pour fonctionner en proxy transparent, si l'on suppose que la machine recevant le flux réseau est 172.16.0.1 (à adapter selon ce que vous voulez faire, voir également le cours), nous utilisons NetFilter/IPTables :

```
iptables -t nat -I PREROUTING -s 172.16.0.1 -p tcp -m tcp --dport 80 -j REDIRECT --to-ports 3128
```

4 Installer squidGuard

```
aptitude install squidguard
cp /etc/squid/squidGuard.conf /etc/squid/squidGuard.conf.orig
emacs /etc/squid/squidGuard.conf
```

Le fichier /etc/squid/squidGuard.conf sera modifié de manière à contenir les lignes suivantes (attention la plupart des lignes sont déjà présentes (pas forcément dans le même ordre) dans le fichier de configuration.

Je vais commencer avec un fichier de configuration minimum, vous pourrez le compléter et l'adapter à votre guise en prenant comme exemple la seconde version (incomplète) de fichier de configuration :

```
# Indiquer où se trouvent blacklist et log :
dbhome /var/lib/squidguard/db
logdir /var/log/squid

# Définir la base contenant ce que l'on veut interdire :
dest asupprimer {
    domainlist asupprimer/domains
    urllist asupprimer/urls
}

# Définir les ACL
# (regardez dans le fichier de config initial c'est plus complet) :
acl {
    default {
          pass !asupprimer all
          redirect http://172.16.0.1
    }
}
```

Il faut maintenant créer les fichiers contenant la liste des domaines interdits (asupprimer/domains) et la liste des url interdites (asupprimer/urls). Attention aux noms des fichiers, attention à l'endroit où vous placez ces fichiers (forcément dans /var/lib/squidguard/db/..). et attention aux permissions sur ces fichiers. Pour ce TP, je place les fichiers dans le répertoire /var/lib/squidguard/db. Le fichier /var/lib/squidguard/db/asupprimer/domains contient :

```
un_domaine_a_supprimer.com
```

Le fichier /var/lib/squidguard/db/asupprimer/urls contient :

```
un_autre_domaine_a_supprimer.com/page_a_supprimer.html
```

Il faut maintenant demander à squidGuard de convertir ces fichiers textes en base de type Berkeley DB, via la commande suivante :

```
squidGuard -d -C all
```

ATTENTION : seules les bases mentionnées dans le fichier squidGuard.conf seront effectivement converties dans le fichier /var/lib/squidguard/db. Si vous omettez une base, elle ne sera pas prise en compte.

Vérifiez que dans le répertoire /var/lib/squidguard/db/asupprimer, vous avez bien deux fichiers .db qui ont été créées.

Par sécurité, assurons nous que squidGuard puisse accéder aux fichier (car nous avons lancé la commande an tant que root et non « proxy ») :

```
chown -R proxy:proxy /var/lib/squidguard/db/*
```

Vérifiez également les permissions, mais normalement c'est correct.

On relance squid :

```
/etc/init.d/squid restart
```

Vérifiez (à l'aide de la commande nmap 172.16.0.1 par exemple) que squid est bien à l'écoute sur le port 3128. Vous pouvez tester les proxy 172.16.0.1:3128 en paramétrant +/- temporairement votre navigateur.

Seconde version (INCOMPLETE) de fichier de configuration squidGuard.conf :

```
# Indiquer où se trouvent blacklist et log :
dbhome /var/lib/squidguard/db
logdir /var/log/squid

# Définir quand on veut travailler (à adapter à votre rythme ☺) :
# TIME RULES:
# abbrev for weekdays:
# s = sun, m = mon, t =tue, w = wed, h = thu, f = fri, a = sat
time workhours {
        weekly s 09:30-12:00 13:00-19:00
        weekly m 09:00-12:00 13:00-19:00
        weekly t 09:00-11:00 12:00-19:00
        weekly w 09:00-12:00 12:00-18:00
        weekly h 09:00-13:00 13:00-18:00
        weekly f 09:00-12:00 13:30-18:00
        weekly a 08:20-13:00 13:30-19:00
}

#Groupes de listes interdites
dest redirector {
    domainlist redirector/domains
    urllist redirector/urls
    expressionlist  redirector/expressions
}
dest warez {
    domainlist warez/domains
    urllist warez/urls
}
dest ads {
    domainlist ads/domains
    urllist ads/urls
}
dest aggressive {
    domainlist aggressive/domains
    urllist aggressive/urls
}
dest drugs {
    domainlist drugs/domains
    urllist drugs/urls
}
dest gambling {
    domainlist gambling/domains
    urllist gambling/urls
    log /var/log/squid/jeux.log
}
dest violence {
    domainlist violence/domains
    urllist violence/urls
    expressionlist  violence/expressions
}

# … LA SUITE … C'EST VOUS QUI LA FAITES …
```

N'oubliez pas de redémarrer le daemon squid pour valider les changement dans les fichiers situés dans le répertoire /etc/squid via une commande du genre « /etc/init.d/squid restart ».

La suite … c'est vous qui la faites ☺ … certains détails sont dans les slides de cours ... et dans la documentation de squid !