Proxy Squid & SquidGuard

Philippe Latu
philippe.latu(at)inetdoc.net
https://www.inetdoc.net

Résumé

Avec l'évolution des usages de l'Internet, il devient nécessaire de garantir des conditions correctes de navigation et une protection minimale de la confidentialité des informations personnelles dans l'infrastructure système et réseau. Le service mandataire avec filtrage d'URLs est un outil indispensable dans la panoplie de sécurisation du trafic Web. On étudie dans ce document la configuration des outils libres Squid et SquidGuard.

Table des matières

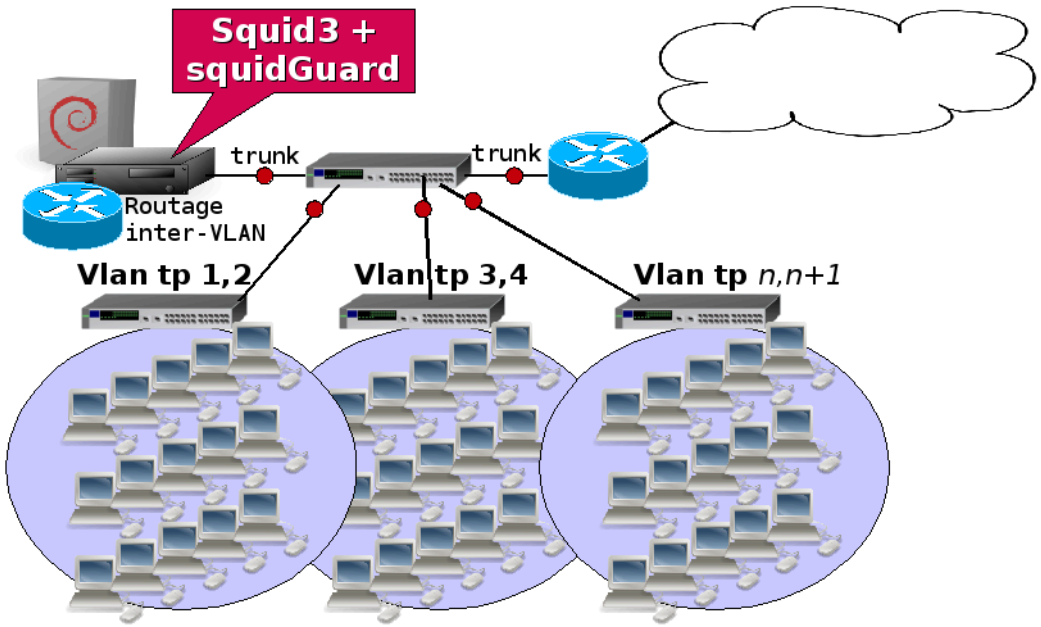
1. Copyright et Licence

1.1. Méta-information

Cet article est écrit avec DocBook XML sur un système Debian GNU/Linux. Il est disponible en version imprimable aux format PDF : `squid-guard.pdf`.

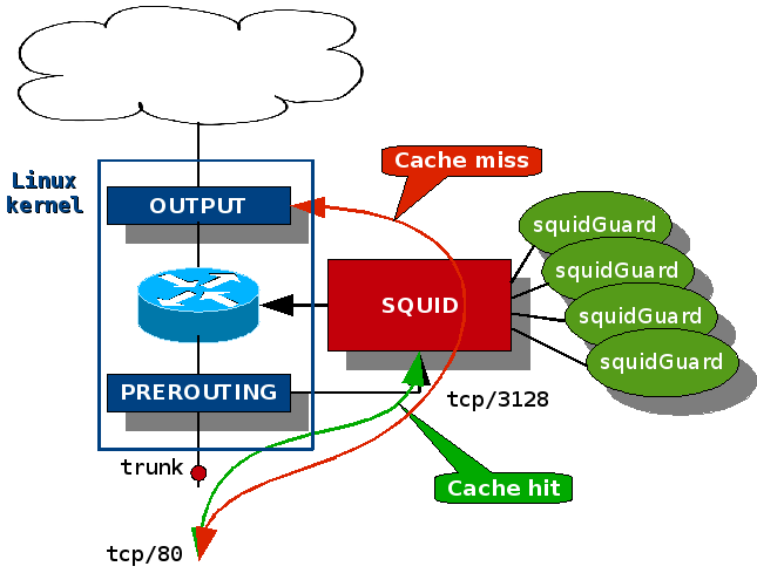
2. Le contexte

Dans une infrastructure d'enseignement, le service Web mandataire ou proxy vient logiquement s'insérer sur la passerelle à l'interface du réseau de campus. En effet, c'est à travers cette passerelle que transite tout le trafic utilisateur vers et depuis l'Internet.



Positionnement du service proxy dans l'architecture d'enseignement

Le service mandataire doit s'intégrer dans l'environnement des services existants de routage, de filtrage et de traduction d'adresses IP. On peut schématiser le fonctionnement de ce service de la façon suivante :



Positionnement du service proxy dans le système

Le mode de fonctionnement schématisé ici suppose que toutes les fonctions de routage, de filtrage, de traduction d'adresses, de proxying et de filtrage d'URLs soient implantées dans le même système physique. Compte tenu des contraintes usuelles d'exploitation, ce mode opératoire convient parfaitement. L'expérience à montré qu'un système moderne supporte très bien la charge. On verra plus loin les éléments de métrologie et de tuning.

Conventions typographiques

Tous les exemples d'exécution des commandes sont précédés d'une invite utilisateur ou prompt spécifique au niveau des droits utilisateurs nécessaires sur le système.

Toute commande précédée de l'invite `$` ne nécessite aucun privilège particulier et peut être utilisée au niveau utilisateur simple.

Toute commande précédée de l'invite `#` nécessite les privilèges du super-utilisateur.

---

3. L'installation des paquets Debian

Pour la mise en place du service Web mandataire et du filtrage d'URLs, on s'appuie sur les paquets suivants de la branche testing de la distribution Debian GNU/Linux.

- squid3 - A full featured Web Proxy cache (HTTP proxy)

  L'application de service Web mandataire proprement dite.

- squidclient - A full featured Web Proxy cache (HTTP proxy) - control utility

  L'outil de contrôle et d'analyse du fonctionnement du service Web mandataire.

- squidguard - filter, redirector and access controller plug for Squid

  L'outil de filtrage des URLs.

- iptables - administration tools for packet filtering and NAT

  La partie utilisateur du système de filtrage réseau d'un système GNU/Linux nécessaire à l'interception du trafic Web.

On débute avec l'installation des trois paquets listés ci-dessus. En plus de l'installation des logiciels, une première configuration des services est mise en place. Cette configuration de base est à retravailler pour l'adapter au contexte d'exploitation.

```
# aptitude install squid3 squidclient squidguard
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Lecture de l'information d'état étendu
Initialisation de l'état des paquets... Fait
Les NOUVEAUX paquets suivants vont être installés :
  squid3 squid3-common{a} squidclient squidguard
0 paquets mis à jour, 4 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de télécharger 1449ko d'archives. Après dépaquetage, 7881ko seront utilisés.
Voulez-vous continuer ? [Y/n/?]
Écriture de l'information d'état étendu... Fait
Prendre : 1 http://ftp.fr.debian.org testing/main squid3-common 3.0.STABLE8-1 [289kB]
Prendre : 2 http://ftp.fr.debian.org testing/main squid3 3.0.STABLE8-1 [936kB]
Prendre : 3 http://ftp.fr.debian.org testing/main squidclient 3.0.STABLE8-1 [87,4kB]
Prendre : 4 http://ftp.fr.debian.org testing/main squidguard 1.2.0-8.4 [136kB]
 1449ko téléchargés en 1s (1204ko/s)
Préconfiguration des paquets...
Sélection du paquet squid3-common précédemment désélectionné.
(Lecture de la base de données... 52990 fichiers et répertoires déjà installés.)
Dépaquetage de squid3-common (à partir de .../squid3-common_3.0.STABLE8-1_all.deb) ...
Sélection du paquet squid3 précédemment désélectionné.
Dépaquetage de squid3 (à partir de .../squid3_3.0.STABLE8-1_i386.deb) ...
Sélection du paquet squidclient précédemment désélectionné.
Dépaquetage de squidclient (à partir de .../squidclient_3.0.STABLE8-1_i386.deb) ...
Sélection du paquet squidguard précédemment désélectionné.
Dépaquetage de squidguard (à partir de .../squidguard_1.2.0-8.4_i386.deb) ...
Traitement des actions différées (« triggers ») pour « man-db »...
Paramétrage de squid3-common (3.0.STABLE8-1) ...
Paramétrage de squid3 (3.0.STABLE8-1) ...
Creating Squid HTTP proxy 3.0 spool directory structure
2008/10/30 18:44:57| Creating Swap Directories
2008/10/30 18:44:57| /var/spool/squid3 exists
2008/10/30 18:44:57| Making directories in /var/spool/squid3/00
2008/10/30 18:44:57| Making directories in /var/spool/squid3/01
2008/10/30 18:44:57| Making directories in /var/spool/squid3/02
2008/10/30 18:44:57| Making directories in /var/spool/squid3/03
2008/10/30 18:44:57| Making directories in /var/spool/squid3/04
2008/10/30 18:44:57| Making directories in /var/spool/squid3/05
2008/10/30 18:44:57| Making directories in /var/spool/squid3/06
2008/10/30 18:44:57| Making directories in /var/spool/squid3/07
2008/10/30 18:44:57| Making directories in /var/spool/squid3/08
2008/10/30 18:44:57| Making directories in /var/spool/squid3/09
2008/10/30 18:44:57| Making directories in /var/spool/squid3/0A
2008/10/30 18:44:57| Making directories in /var/spool/squid3/0B
2008/10/30 18:44:57| Making directories in /var/spool/squid3/0C
2008/10/30 18:44:57| Making directories in /var/spool/squid3/0D
2008/10/30 18:44:58| Making directories in /var/spool/squid3/0E
2008/10/30 18:44:58| Making directories in /var/spool/squid3/0F
Restarting Squid HTTP Proxy 3.0: squid3.
Paramétrage de squidclient (3.0.STABLE8-1) ...
Paramétrage de squidguard (1.2.0-8.4) ...
/usr/sbin/update-squidguard not run automatically
Check Debconf settings for squidguard to change
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Lecture de l'information d'état étendu
Initialisation de l'état des paquets... Fait
Écriture de l'information d'état étendu... Fait
```

Avant de s'attaquer à la configuration des services, on commence par modifier l'arborescence de l'outil de filtrage d'URLs squidguard. Par défaut, cet outil est prévu pour fonctionner avec la génération précédente du service mandataire squid : la version 2.xx

```
# mv /etc/squid/squidGuard.conf /etc/squid3/
# rm -rf /etc/squid
# ln -s /etc/squid3 /etc/squid
# ll /etc/ | grep squid
lrwxrwxrwx 1 root   root     11 nov  2 16:43 squid -> /etc/squid3
drwxr-xr-x 2 root   root   1,0K nov  2 16:42 squid3

# ll /etc/squid3/
total 162K
-rw-r--r-- 1 root root  421 jui 21 10:01 msntauth.conf
-rw-r--r-- 1 root root 158K jui 21 10:00 squid.conf
-rw-r--r-- 1 root root 1,3K jui 23 14:28 squidGuard.conf
```

Avec les opérations ci-dessus, les fichiers de configurations sont tous placés dans le même répertoire et le lien symbolique permet de s'assurer qu'une mise à jour du paquet squidguard se déroule correctement.

4. La configuration du service Web mandataire (proxy)

Tous les paramètres de configuration du démon `squid` sont rassemblés dans le fichier `squid.conf`. Ce fichier est particulièrement volumineux et par conséquent difficile à prendre en main. Décrire toutes les fonctionnalités offertes par ces paramètres dépasse très largement le cadre de ce billet. On se contente donc de présenter un patch qui fait ressortir les paramètres modifiés pour les besoins du contexte d'exploitation.

```
$ diff -uBb /etc/squid3/squid.conf squid.conf.new

--- /etc/squid3/squid.conf 2008-07-21 12:31:32.000000000 +0200
+++ squid.conf.new 2008-11-08 19:28:18.000000000 +0100
@@ -579,6 +579,7 @@
 #
 #Recommended minimum configuration:
 acl manager proto cache_object
+acl mynetworks src 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16
 acl localhost src 127.0.0.1/32
 acl to_localhost dst 127.0.0.0/8
 #
@@ -644,6 +645,7 @@
 # Adapt localnet in the ACL section to list your (internal) IP networks
 # from where browsing should be allowed
 #http_access allow localnet
+http_access allow mynetworks
 http_access allow localhost

 # And finally deny all other access to this proxy
@@ -677,6 +679,7 @@
 #
 #Allow ICP queries from local networks only
 #icp_access allow localnet
+icp_access allow mynetworks
 icp_access deny all

 #  TAG: htcp_access
@@ -867,7 +870,7 @@
 # visible on the internal address.
 #
 # Squid normally listens to port 3128
-http_port 3128
+http_port 3128 transparent

 #  TAG: https_port
 # Note: This option is only available if Squid is rebuilt with the
@@ -1564,6 +1567,7 @@
 #
 #Default:
 # cache_mem 8 MB
+cache_mem 64 MB

 #  TAG: maximum_object_size_in_memory (bytes)
 # Objects greater than this size will not be attempted to kept in
@@ -1573,6 +1577,7 @@
 #
 #Default:
 # maximum_object_size_in_memory 8 KB
+maximum_object_size_in_memory 128 KB

 #  TAG: memory_replacement_policy
 # The memory replacement policy parameter determines which
@@ -1582,7 +1587,7 @@
 #
 #Default:
 # memory_replacement_policy lru
-
+memory_replacement_policy heap LFUDA

 # DISK CACHE OPTIONS
 # -----------------------------------------------------------------------------
@@ -1624,6 +1629,7 @@
 #
 #Default:
 # cache_replacement_policy lru
+cache_replacement_policy heap LFUDA

 #  TAG: cache_dir
 # Usage:
@@ -1731,6 +1737,7 @@
 #
 #Default:
 # cache_dir ufs /var/spool/squid3 100 16 256
+cache_dir aufs /var/spool/squid3 16384 16 256

 #  TAG: store_dir_select_algorithm
 # Set this to 'round-robin' as an alternative.
@@ -2211,6 +2218,7 @@
 #
 #Default:
 # none
+url_rewrite_program /usr/bin/squidGuard -c /etc/squid/squidGuard.conf
```

Voici quelques éléments sur les paramètres introduits ou modifiés.

acl mynetworks src 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16

Cette liste de contrôle d'accès introduite avec l'option `acl` désigne les adresses réseau utilisées dans l'infrastructure de travaux pratiques. Si on se réfère aux tableaux de Correspondance entre VLAN et réseau IP de cette infrastructure, on considère que le trafic Web est susceptible de provenir de toutes les classes d'adresses IP privées désignées dans le document standard RFC1918 Address Allocation for Private Internets.

L'option `src` indique que ce sont les adresses IP sources qui sont utilisées comme critère d'accès au service proxy.

http_access allow mynetworks

La directive `http_access` contrôle l'accès au service via le protocole HTTP. Dans le cas présent, il s'agit d'ouvrir l'accès aux réseaux de l'infrastructure de travaux pratiques. La configuration par défaut, telle que fournie lors de l'installation du paquet, interdit tout accès au service. Cette règle restreint donc l'accès aux seules adresses IP utilisées pour les travaux pratiques.

icp_access allow mynetworks

La directive `icp_access` joue le même rôle que la précédente pour le protocole ICP. L'Internet Cache Protocol est utilisé pour le dialogue entre services mandataires. Tout comme dans le cas précédent, on autorise le fonctionnement du protocole à partir des adresses IP de l'infrastructure de travaux pratiques.

http_port 3128 transparent

La directive `http_port` désigne le numéro du port sur lequel le service mandataire est en écoute. Dans le cas de squid, le numéro usuel est le `3128`. Il n'est pas nécessaire de modifier ce numéro de port. L'option `transparent` est importante. Elle définit le mode d'utilisation du service : un cache transparent pour le trafic Web utilisateur.

cache_mem 64 MB

Le paramètre `cache_mem` désigne la quantité idéale de mémoire allouée pour le stockage de différents types d'objets en «transit» dans le service. Cette quantité peut très bien être dépassée en fonction du taux de requêtes entrantes. Si ce taux est élevé, ce paramètre est utilisé pour calculer le seuil de remplacement des anciens objets dans la mémoire.

maximum_object_size_in_memory 128 KB

Définition de la taille maximum d'un objet conservé en mémoire cache. Les objets de taille supérieure ne sont pas conservés en mémoire vive.

memory_replacement_policy heap LFUDA

Définition de la politique utilisée pour remplacer les objets stockés dans le cache mémoire lorsqu'il est nécessaire de trouver de la place libre. L'option retenue ici est Least Frequently Used with Dynamic Aging. Elle correspond à un usage de type pile basé sur le taux d'utilisation d'un objet du cache combiné à son ancienneté.

cache_replacement_policy heap LFUDA

Définition de la politique utilisée pour remplacer les objets stockés dans le cache disque lorsqu'il est nécessaire de trouver de la place libre. L'option retenue ici est identique à la précédente.

cache_dir aufs /var/spool/squid3 16384 16 256

Définition du mode d'utilisation du cache disque du service mandataire. L'option `aufs` correspond au format de stockage des objets sur disque. Elle utilise les fonctions multi-tâches POSIX Threads.

Du point de vue occupation disque, on réserve 16Go distribués sur 16 répertoires de premier niveau puis sur 256 répertoires de second niveau.

url_rewrite_program /usr/bin/squidGuard -c /etc/squid/squidGuard.conf

La directive `url_rewrite_program` désigne l'outil utilisé pour le filtrage des URLs : squidGuard.

url_rewrite_children 20

Définition du nombre de processus enfants générés pour répondre aux requêtes transmises par le démon `squid`. La valeur donnée doit permettre d'optimiser les temps de réponses en lançant suffisamment d'instances squidGuard tout en utilisant raisonnablement les ressources du système.

request_header_max_size 32 KB

Définition de la taille maximum d'un en-tête HTTP lors d'une requête.

Le choix d'une taille de 32Ko doit permettre de laisser passer le trafic «normal» sans entraver les accès. À voir en fonction d'utilisations particulières du protocole HTTP.

reply_header_max_size 32 KB

Définition de la taille maximum d'un en-tête de réponse HTTP. La valeur retenue est identique à celle du paramètre précédent.

acl debian dstdomain .debian.org, always_direct allow debian

Cette liste de contrôle d'accès concerne toutes les requêtes à destination du domaine `.debian.org` et de ses sous-domaines.

La directive `always_direct` spécifie les requêtes qui doivent toujours être directement transmises au serveur d'origine.

pipeline_prefetch on

Directive utilisée pour doper les performances et se rapprocher d'une navigation Web sans service mandataire.

Bien sûr, tous ces paramètres peuvent être «retravaillés» pour répondre mieux à d'autres contextes d'exploitation. Seule l'expérience acquise en cours d'exploitation peut permettre de répondre précisément à un besoin.

Les commentaires du fichier de configuration `/etc/squid/squid.conf` constituent le meilleur point d'entrée pour aborder la documentation du paquet squid3. Ensuite, le Wiki Squid fournit des exemples de configurations très intéressants.

Arrivé à cette étape, il ne reste plus qu'à appliquer le patch sur le fichier de configuration du service mandataire.

```
# cp /etc/squid/squid.conf /etc/squid/squid.conf.dpkg-dist
# patch -l -p0 /etc/squid/squid.conf squid.conf.patch
patching file /etc/squid/squid.conf

# ll /etc/squid/
total 336K
-rw-r--r-- 1 root root  421 jui 21 12:31 msntauth.conf
-rw-r--r-- 1 root root 158K nov  8 19:40 squid.conf
-rw-r--r-- 1 root root 158K nov  8 19:40 squid.conf.dpkg-dist
-rw-r--r-- 1 root root 1,3K jui 24 23:15 squidGuard.conf
```

À ce stade de la configuration, le service n'est pas encore opérationnel. Il faut maintenant configurer l'application de filtrage des URLs : squidGuard.

5. La configuration du service de filtrage des URLs

Comme dans le cas du service mandataire, le fichier `/etc/squid/squidGuard.conf` est point de départ de la configuration de l'outil. À la différence de squid, cette configuration dépend essentiellement de «l'alimentation» en listes noires d'URLs ou de noms de domaines.

Même si ce fichier est moins volumineux que celui de la section précédente, on reprend la technique du patch pour illustrer les différences entre le fichier distribué via le paquet et celui exploité dans le contexte courant.

```
$ diff -uBb /etc/squid3/squidGuard.conf squidGuard.conf.new

--- squidGuard.conf.dpkg-dist 2008-11-10 11:17:36.000000000 +0100
+++ squidGuard.conf.new 2008-11-10 11:15:29.000000000 +0100
@@ -3,7 +3,7 @@
 #

 dbhome /var/lib/squidguard/db
-logdir /var/log/squid
+logdir /var/log/squid3

 #
 # TIME RULES:
@@ -46,19 +46,106 @@
 # DESTINATION CLASSES:
 #

-dest good {
+dest adult {
+ domainlist adult/domains
+ expressionlist adult/expressions
+ urllist  adult/urls
 }

-dest local {
+dest agressif {
+ domainlist agressif/domains
+ expressionlist agressif/expressions
+ urllist  agressif/urls
 }

-#dest adult {
-# domainlist adult/domains
-# urllist  adult/urls
-# expressionlist adult/expressions
-# redirect  http://admin.foo.bar.no/cgi-bin/squidGuard.cgi?clientaddr=%a+clientname=%n+clientident=%i+srcclas
-#}
+dest astrology {
+ domainlist astrology/domains
+ urllist  astrology/urls
+}
+
+dest dangerous_material {
+ domainlist dangerous_material/domains
+ urllist  dangerous_material/urls
+}
+
+dest drogue {
+ domainlist drogue/domains
+ urllist  drogue/urls
+}
+
+dest financial {
+ domainlist financial/domains
+ urllist  financial/urls
+}
+
+dest forums {
+ domainlist forums/domains
+ urllist  forums/urls
+}
+
+dest gambling {
+ domainlist gambling/domains
+ urllist  gambling/urls
+}
+
+dest games {
+ domainlist games/domains
+ urllist  games/urls
+}
+
+dest hacking {
+ domainlist hacking/domains
+ urllist  hacking/urls
+}
+
+dest malware {
+ domainlist malware/domains
+}

+dest marketingware {
+ domainlist marketingware/domains
+}
```

Bien entendu, la sélection des catégories retenues est totalement arbitraire et le patch de configuration proposé ne prétend pas répondre à tous les besoins. Voici quelques éléments sur la configuration de squidguard en reprenant les mêmes limitations que pour le service mandataire : la documentation de toutes les options de l'outil sort complètement du cadre de ce billet.

dbhome

Définition du répertoire de stockage de la base de données des listes noires. L'utilisation de ce répertoire est détaillée dans Section 6, « La gestion des listes noires ».

logdir

Définition du répertoire de stockage du fichier de journalisation de l'outil. Généralement, on utilise le même répertoire que le service mandataire sachant que les processus des deux services sont exécutés sous la même identité. Ces processus ont donc les mêmes droits sur l'arborescence.

Comme toutes les entrées de liste noire utilisent la même syntaxe, on ne s'intéresse qu'à un exemple : les listes noires warez.

dest warez

Définition d'une catégorie de trafic à bloquer. À l'intérieur de cette catégorie, on peut définir des listes de noms de domaines, d'URLs et d'expressions rationnelles.

domainlist warez/domains

Fichier contenant la liste des noms de domaines de la catégorie à bloquer. Il s'agit d'un fichier texte avec un nom de domaine par ligne.

expressionlist warez/expressions

Fichier texte de définition des expressions rationnelles de filtrage. Généralement, il s'agit de filtrer les accès aux contenus stockés dans les caches des outils de recherche célèbres.

urllist warez/urls

Fichier texte contenant la liste des URLs accessibles à partir de noms de domaines «valides». Il s'agit généralement de sous-répertoires d'hébergement de fournisseurs d'accès Internet.

Une catégorie white pour «liste blanche» a été ajoutée manuellement pour traiter les exceptions ; c'est-à-dire les domaines pour lesquels l'accès est toujours autorisé.

La dernière section concerne l'application des règles de traitement des catégories définies plus haut.

acl

Section d'application des règles de traitement. Il est possible de définir plusieurs sections de ce type et de mettre en place une administration plus fine du service. Dans notre exemple, on ne définit qu'un seul mode de traitement global pour tous les utilisateurs situés dans le périmètre d'interception du service mandataire.

default

Section de définition des traitements par défaut.

pass

Tout le trafic vers les entrées enregistrées dans une catégorie précédée du caractère ! est bloqué. À l'inverse, tout le trafic correspondant à une catégorie listée est autorisé.

Dans l'exemple, on a utilisé plusieurs lignes débutant par l'instruction `pass` pour éviter d'avoir une ligne trop longue. Il faut considérer que les catégories sont examinées séquentiellement pour les lignes `pass` saisies.

La dernière «catégorie» notée `all` correspond à toutes les autres destinations non filtrées.

redirect

Instruction de redirection vers une page Web donnée pour les destinations bloquées. Ici, la redirection a lieu vers le service Web interne à l'infrastructure de travaux pratiques. Voir Section 7, « La redirection de page Web pour le trafic bloqué ».

6. La gestion des listes noires

Le choix de la source de listes noires d'URLs (et|ou) de noms de domaines a un impact très important sur la configuration de l'outil de filtrage squidGuard. Pour les besoins d'exploitation d'une infrastructure d'enseignement, le service offert par Centre de Ressources Informatiques de l'Université de Toulouse I convient parfaitement !.

La démarche suivie ici ne s'embarrasse pas du choix de tel ou tel service. On télécharge la totalité des listes en une opération unique et on utilise le script d'indexation fourni par le paquet squidguard pour construire la base de données en fonction de la configuration présentée dans Section 5, « La configuration du service de filtrage des URLs ».

L'adresse de la page de téléchargement des listes noires est : http://cri.univ-tlse1.fr/blacklists/. Voici un exemple de traitement manuel de ces listes.

```
$ cd /var/tmp/
$ wget http://dsi.ut-capitole.fr/blacklists/download/blacklists.tar.gz
--2015-09-05 21:02:12--  http://dsi.ut-capitole.fr/blacklists/download/blacklists.tar.gz
Résolution de dsi.ut-capitole.fr (dsi.ut-capitole.fr)… 193.49.48.249
Connexion à dsi.ut-capitole.fr (dsi.ut-capitole.fr)|193.49.48.249|:80… connecté.
requête HTTP transmise, en attente de la réponse… 200 OK
Taille : 9606794 (9,2M) [application/x-gzip]
Sauvegarde en : « blacklists.tar.gz »

blacklists.tar.gz  100%[=================>]   9,16M   346KB/s   ds 48s

2015-09-05 21:03:00 (195 KB/s) — « blacklists.tar.gz » sauvegardé [9606794/9606794]

$ tar xf blacklists.tar.gz
```

À l'issue de ce téléchargement et de la décompression du fichier d'archive, le répertoire `blacklist` contient les répertoires qui correspondent aux différentes «catégories» de listes noires. Le choix entre les différentes catégories se fait dans le fichier de configuration du paquet squidguard.

```
# mv blacklists/* /var/lib/squidguard/db/

# update-squidguard
Double checking directory and file permissions...done!
Re-building SquidGuard db files...done!
Reloading Squid...done!
```

Une fois que l'on a validé le processus manuellement, il est préférable d'automatiser le traitement pour assurer une mise à jour périodique des enregistrements en liste noire. Voici un exemple de Shell script à déposer dans le répertoire `/etc/cron.weekly/` pour qu'il soit exécuté toutes les semaines.

```
#!/bin/sh

cd /var/tmp
rm -rf blacklist* >/dev/null 2>&1
wget -q ftp://ftp.univ-tlse1.fr/pub/reseau/cache/squidguard_contrib/blacklists.tar.gz

if [ -f /var/tmp/blacklists.tar.gz ]; then
 tar xf blacklists.tar.gz
 cp -au blacklists/* /var/lib/squidguard/db/ >/dev/null 2>&1
 /usr/sbin/update-squidguard >/dev/null 2>&1
fi
```

Ce script reprend simplement les étapes réalisées manuellement plus haut et fait appel au script `update-squidguard` fourni avec le paquet squidguard.

Il faut juste noter que ce dernier script à été modifié pour relancer `squid3` en lieu et place de squid. Voici une copie dans laquelle les lignes 13 et 15 ont été modifiées.

```
#! /bin/sh
# db update script for Chastity
#

echo -n "Double checking directory and file permissions..."
chown -R proxy.proxy /var/lib/squidguard/db >/dev/null 2>&1
chmod 2770 /var/lib/squidguard/db >/dev/null 2>&1
echo "done!"
echo -n "Re-building SquidGuard db files..."
su - proxy -c "squidGuard -C all" >/dev/null 2>&1
su - proxy -c "squidGuard -u" >/dev/null 2>&1
echo "done!"
if [ -e /etc/init.d/squid3 ]; then
 echo -n "Reloading Squid..."
 /etc/init.d/squid3 reload >/dev/null 2>&1
 echo "done!"
fi
```

7. La redirection de page Web pour le trafic bloqué

Lorsqu'une requête HTTP correspond à une règle de blocage, elle est redirigé vers une page définie dans le fichier de configuration de squidguard. Le paquet fournit un exemple de script CGI qui renvoie une page Web type à l'utilisateur.

Voici les étapes nécessaires à la mise en œuvre de ce script.

```
# cd /usr/lib/cgi-bin/
# cp /usr/share/doc/squidguard/examples/squidGuard.cgi.gz .
# gzip -d squidGuard.cgi.gz
# iconv -f iso8859-1 -t utf8 /usr/lib/cgi-bin/squidGuard.cgi >squidGuard.cgi.tmp
# mv squidGuard.cgi.tmp /usr/lib/cgi-bin/squidGuard.cgi
# chmod +x squidGuard.cgi
```

Pour l'édition du script CGI qui est assez long, on reprend la technique du patch pour illustrer les différences entre la version délivrée avec le paquet et celle en exploitation.

```
$ diff -uBb squidGuard.cgi /usr/lib/cgi-bin/squidGuard.cgi

--- squidGuard.cgi 2008-11-10 18:37:30.326524640 +0100
+++ /usr/lib/cgi-bin/squidGuard.cgi 2008-11-10 18:35:19.000000000 +0100
@@ -64,10 +64,11 @@
   "nl (Nederlands),",
   "no (norsk)."
         );
-$image       = "/images/blocked.gif";     # RELATIVE TO DOCUMENT_ROOT
-$redirect    = "http://admin.your-domain/images/blocked.gif";  # "" TO AVOID REDIRECTION
-$proxy       = "proxy.your-domain";     #
-$proxymaster = "operator\@your-domain";    #
+$image       = "/images/access_denied.jpg";    # RELATIVE TO DOCUMENT_ROOT
+$redirect    = "http://www.stri/images/access_denied.jpg";  # "" TO AVOID REDIRECTION
+$proxy       = "proxy.stri";     #
+$proxymaster = "abuse\@stri";     #
+$lang        = "fr";
 $autoinaddr  = 2;   # 0|1|2;
      # 0 TO NOT REDIRECT
      # 1 TO AUTORESOLVE & REDIRECT IF UNIQUE
@@ -482,14 +483,14 @@
  @supported ];

   %logo->{"default"}->{"url"}
-    = "http://www.squidguard.org/images/squidGuard.gif";
+    = "http://www.squidguard.org/Logos/squidGuard.gif";
   %logo->{"default"}->{"href"}
     = "http://www.squidguard.org/";

   %logo->{"default"}->{"url"}
-    = "http://info.your-domain/images/eto.small.gif";
+    = "http://www.squidguard.org/Logos/squidGuard.gif";
   %logo->{"default"}->{"href"}
-    = "http://www.your-domain/";
+    = "http://www.stri/";
 }
 #
 # END OF CONFIGURABLE OPTIONS
@@ -608,7 +609,10 @@
       print "       $text\n";
     }
   }
-  print "  </TITLE>\n </HEAD>\n";
+  print "  </TITLE>\n";
+  print " <meta http-equiv=\"Content-Type\" content=\"text/html; charset=utf-8\" />\n";
+  print " <meta name=\"Content-Language\" content=\"fr\" />\n";
+  print " </HEAD>\n";
   print " <BODY BGCOLOR=\"#FFFFFF\">\n";
   print "  <TABLE BORDER=0 ALIGN=CENTER WIDTH=100%>\n";
   print "   <TR>\n";
```

Les modifications sont exclusivement d'ordre cosmétique. Il s'agit juste d'afficher le logo squidguard correctement et de donner des paramètres de redirection corrects dans le contexte d'exploitation. De plus, pour faciliter l'affichage entre les différents types de navigateurs, on ajoute deux jeux de balises `meta`, dans l'en-tête HTML de la page Web.

Voici le résultat d'une tentative de navigation sur un site de la catégorie warez.

Résultat de la rédirection d'affichage

8. L'interception du trafic Web

Il existe plusieurs techniques pour intercepter le trafic Web sortant d'un périmètre en fonction des équipements utilisés. Ces techniques sont résumées dans le document Transparent Proxy with Linux and Squid mini-HOWTO. Si ce document date un peu, les principes restent les mêmes. Dans l'architecture présentée plus haut, le service mandataire est exécuté directement sur la passerelle de routage. L'interception du trafic Web vers le service mandataire doit donc se faire sur le même système.

En utilisant le système de filtrage réseau netfilter/iptables sur un système GNU/Linux, la règle de base de l'interception est la suivante :

```
# iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 -j REDIRECT --to-port 3128
```

Avec cette règle de la table de traduction d'adresses (NAT), tout paquet entrant par l'interface `eth1` utilisant le protocole TCP avec le port destination `80` est redirigé vers le processus local en écoute sur le port `3128`. Dans ce contexte, le processus local correspond au numéro de port défini dans la configuration du service Web mandataire : squid. Il faut aussi préciser que l'interface `eth1` doit être située du côté interne de la passerelle d'interface avec le réseau du Campus.

Il faut aussi noter que le routage des paquets IP doit être activé sur le système. Dans le contexte présenté, cette condition est déjà remplie puisqu'il s'agit justement d'un routeur. On valide le routage au niveau du noyau Linux dans l'arborescence `/proc/` ou via le fichier `/etc/sysctl.conf` s'il s'agit d'une configuration permanente.

```
$ cat /proc/sys/net/ipv4/ip_forward
1

$ cat /etc/sysctl.conf |grep ip_forward
net/ipv4/ip_forward=1
#net/ipv6/ip_forward=1
```

À partir de ces deux conditions de base, il faut intégrer l'interception de trafic dans le fonctionnement global du système de filtrage. Voici un exemple fonctionnel de script iptables restreint à l'interception Web.

```
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# NAT
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#  P O S T R O U T I N G
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-A POSTROUTING -o eth0 -p tcp --tcp-flags SYN,RST SYN -m tcpmss --mss 1400:1536 -j TCPMSS --clamp-mss-to-pmtu
-A POSTROUTING -o eth0 -j SNAT --to-source aaa.bbb.ccc.ddd
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#  P R E R O U T I N G
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# NTP postes clients
-A PREROUTING -i eth1+ -p udp --dport 123 -j REDIRECT --to-port 123
# Interception du trafic Web des postes clients
-A PREROUTING -i eth1+ -p tcp --dport 80 -j REDIRECT --to-port 3128
COMMIT
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#  N e t f i l t e r
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#  I N P U T
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Suivi de communication chaîne INPUT
-A INPUT -p icmp -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A INPUT -p icmp --icmp-type destination-unreachable -m conntrack --ctstate RELATED -j ACCEPT
-A INPUT -p icmp --icmp-type time-exceeded -m conntrack --ctstate RELATED -j ACCEPT
-A INPUT -p ospf -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p igmp -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p udp -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp ! --syn -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A INPUT -p tcp --syn -m conntrack --ctstate RELATED -j ACCEPT
# Boucle locale
-A INPUT -i lo -j ACCEPT
# NTP
-A INPUT -i eth1+ -p udp --dport 123 -m conntrack --ctstate NEW -j ACCEPT
# WWW
-A INPUT -i eth1+ -p tcp --syn --dport 80 -m conntrack --ctstate NEW -j ACCEPT
-A INPUT -i eth1+ -p tcp --syn --dport 443 -m conntrack --ctstate NEW -j ACCEPT
# proxy Web
-A INPUT -i eth1+ -p tcp --syn --dport 3128 -m conntrack --ctstate NEW -j ACCEPT
# Poubelle
-A INPUT -m conntrack --ctstate INVALID -m limit --limit 5/min -j LOG --log-prefix "INPUT/rejected.iptables: "
-A INPUT -m conntrack --ctstate INVALID -j DROP
-A INPUT -p tcp -j REJECT --reject-with tcp-reset
-A INPUT -p udp -j REJECT --reject-with icmp-port-unreachable
-A INPUT -j LOG --log-prefix "INPUT/poubelle: "
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#  F O R W A R D
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Suivi de communication chaîne FORWARD
-A FORWARD -p icmp -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A FORWARD -p icmp -m conntrack --ctstate RELATED --icmp-type destination-unreachable -j ACCEPT
-A FORWARD -p icmp -m conntrack --ctstate RELATED --icmp-type time-exceeded -j ACCEPT
-A FORWARD -p ospf -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -p udp -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -p tcp -m conntrack --ctstate ESTABLISHED -m tcp ! --syn -j ACCEPT
-A FORWARD -p tcp -m conntrack --ctstate RELATED -m tcp --syn -j ACCEPT
# Boucle locale
-A FORWARD -i lo -j ACCEPT
# Salles de TP
-A FORWARD -i eth1+ -p tcp -m tcp --syn --sport 1024: -m conntrack --ctstate NEW -j ACCEPT
-A FORWARD -i eth1+ -p udp --sport 1024: -m conntrack --ctstate NEW -j ACCEPT
-A FORWARD -i eth1+ -p icmp --icmp-type echo-request -m limit --limit 5/sec -m conntrack --ctstate NEW -j ACCE
# Poubelle
-A FORWARD -m conntrack --ctstate INVALID -m limit --limit 5/min -j LOG --log-prefix "FORWARD/rejected.iptable
-A FORWARD -m conntrack --ctstate INVALID -j DROP
-A FORWARD -p tcp -j REJECT --reject-with tcp-reset
-A FORWARD -p udp -j REJECT --reject-with icmp-port-unreachable
-A FORWARD -j LOG --log-prefix "FORWARD/poubelle: "
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#  O U T P U T
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Règles anti-fuites
-A OUTPUT -o eth0 -p tcp --dport 135:139 -j DROP
-A OUTPUT -o eth0 -p tcp --dport 445 -j DROP
-A OUTPUT -o eth0 -p udp --dport 135:139 -j DROP
```

Sur le système illustré, ce script nommé `active` est placé dans le répertoire `/var/lib/iptables/`. On applique les règles contenues dans ce fichier à l'aide de la commande `# iptables-restore </var/lib/iptables/active`.

Pour plus de détails sur la rédaction de ce genre de script, on peut consulter le support Introduction au filtrage réseau et surtout le Tutoriel Iptables.

Pour valider le fonctionnement de l'interception de trafic Web, il suffit de visualiser les compteurs de paquets pour lesquels il y a eu correspondance avec une règle de filtrage.

- Pour la règle d'interception proprement dite, le compte des paquets traités dans la chaîne `PREROUTING` doit évoluer.

```
# iptables -t nat -vL PREROUTING | grep REDIRECT
16  1216 REDIRECT   udp  --  eth1+  any  anywhere  anywhere   udp dpt:ntp redir ports 123
11   528 REDIRECT   tcp  --  eth1+  any  anywhere  anywhere   tcp dpt:www redir ports 3128
```

- Pour l'utilisation du service mandataire Web proxy, on doit retrouver le même compte du nombre de paquets dans la chaîne `INPUT`.

```
# iptables -vL INPUT |grep 3128
11   528 ACCEPT     tcp  --  eth1+  any  anywhere  anywhere   tcp dpt:3128 flags:FIN,SYN,RST,ACK/SYN state NEW
```

- Enfin, les journaux doivent montrer qu'il y a bien eu sollicitation du service.

```
# tail /var/log/squid3/access.log
1226421464.379  60869 172.16.48.64 TCP_MISS/200 925 POST \
  http://update.microsoft.com/v6/UpdateRegulationService/UpdateRegulation.asmx \
  - DIRECT/207.46.17.93 text/xml
1226422061.838     48 172.16.48.65 TCP_MISS/200 407 HEAD \
  http://download.windowsupdate.com/v8/windowsupdate/redir/muv3wuredir.cab? \
  - DIRECT/204.160.98.121 application/octet-stream
1226422062.528    670 172.16.48.65 TCP_MISS/200 407 HEAD \
  http://update.microsoft.com/v8/windowsupdate/selfupdate/wuident.cab? \
  - DIRECT/65.55.184.93 application/octet-stream
1226422062.563     24 172.16.48.65 TCP_MISS/200 408 HEAD \
  http://download.windowsupdate.com/v8/windowsupdate/a/selfupdate/WSUS3/x86/Other/wsus3setup.cab? \
  - DIRECT/204.160.98.121 application/octet-stream
1226422062.680    114 172.16.48.65 TCP_MISS/200 25492 GET \
  http://download.windowsupdate.com/v8/windowsupdate/a/selfupdate/WSUS3/x86/Other/wsus3setup.cab? \
  - DIRECT/204.160.98.121 application/octet-stream
1226422065.039     24 172.16.48.65 TCP_REFRESH_UNMODIFIED/200 405 HEAD \
  http://download.windowsupdate.com/v8/windowsupdate/redir/muv3wuredir.cab? \
  - DIRECT/204.160.98.121 application/octet-stream
1226422069.106     35 172.16.48.65 TCP_REFRESH_UNMODIFIED/200 406 HEAD \
  http://download.windowsupdate.com/v8/windowsupdate/redir/muv3wuredir.cab? \
  - DIRECT/205.128.69.124 application/octet-stream
```

9. Pour conclure, en attendant la suite

Être parvenu jusqu'à la lecture de cette ligne de ce billet relève déjà de l'exploit ! Ce billet est bien trop long et l'objectif de sécurisation n'est pas encore atteint. Pour autant, on dispose d'un service fonctionnel avec filtrage des URLs. C'est la raison pour laquelle on s'arrête ici en attendant la rédaction du volet authentification et sécurisation du dialogue entre le navigateur du poste client et le service Web mandataire.

En espérant que vous aurez trouvé les informations de ce billet pertinentes !