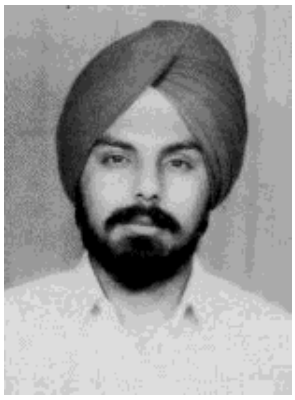
LinuxFocus article number 235
http://linuxfocus.org

Configurer Squid comme serveur proxy



par D.S. Oberoi
<ds_oberoi/at/yahoo.com>

L´auteur:
D.S. Oberoi vit à Jammu, aux Indes et rencontre des problèmes récurrents pour se connecter à Internet, suite aux tensions politiques.

Traduit en Français par:
Paul Delannoy
<freepol/at/free.fr>

Résumé:

Linux est devenu synonyme de mise en réseau. Il est utilisé aussi bien au bureau qu'à la maison comme serveur de fichiers, d'impression, de courrier ou d'applications et il est de plus en plus souvent serveur "proxy".

Un serveur "proxy" permet de partager un accès Internet entre plusieurs utilisateurs avec une seule connexion. Un bon serveur proxy propose également un mécanisme de cache des requêtes, qui permet d'accéder aux données en utilisant les ressources locales au lieu du web, réduisant les temps d'accès et la bande passante consommée. Squid est un logiciel de cette catégorie, qui autorise le proxy, le cache des protocoles HTTP, ftp, gopher, etc. Il supporte également SSL, les contrôles d'accès, le cache de DNS et fournit une trace complète (log) de toutes les requêtes. Squid est aussi disponible pour Windows NT chez Logi Sense.

L'objet de cet article est de proposer des lignes directrices pour la configuration d'un serveur proxy et les moyens de fournir un accès contrôlé aux utilisateurs.

Squid est-il installé ?

Le fichier rpm Squid est inclus dans la distribution RedHat 7.1 et s'installe automatiquement si l'option réseau est sélectionnée. La commande rpm ci-dessous permet de vérifier sa présence :

```
rpm -q squid
```

La dernière version de Squid est toujours disponible sur la page Squid et ses sites miroirs. Squid peut être installé sur un système par la commande :

```
rpm -ivh squid-2.3.STABLE4-10.i386.rpm
```

Configurer Squid

Le fonctionnement et le comportement de Squid sont gérés par les indications de configuration figurant dans le fichier squid.conf; ce fichier est en général placé dans le répertoire /etc/squid. Préparer ce fichier est une opération de longue haleine, car il comporte de nombreuses pages, mais chaque option est accompagnée d'explications précises.

La première chose à définir est la valeur http_port, qui spécifie le numéro de port que Squid va écouter pour satisfaire les requêtes des clients; c'est par défaut 3128, mais ce peut être une autre valeur à votre convenance. En même temps que ce numéro de port, vous pouvez préciser l'adresse IP de la machine qui exécute Squid; par exemple :

http_port 192.168.0.1:8080

Cette déclaration lie Squid à l'adresse IP 192.168.0.1 sur le port 8080. Tout numéro de port est utilisable; mais assurez-vous qu'aucune autre application n'utilise le port choisi. Des lignes semblables permettent de déclarer d'autres ports pour d'autres services.

Contrôle d'accès

Les possibilités de contrôler l'accès à Internet sont nombreuses, comme limiter cet accès à des plages horaires particulières, fournir des informations depuis le cache, autoriser seulement certains sites ou groupes de sites, etc... Squid dispose pour ces contrôles de deux types de composants : les éléments ACL (Access Control List) et la liste d'accès. Une liste d'accès, autorise ou refuse l'accès au service.

Ci-dessous quelques uns des plus importants éléments ACL

- src : Source c-à-d. l'adresse IP du client
- dst : Destination c-à-d. l'adresse IP du serveur
- srcdomain : Source c-à-d. le nom de domaine du client
- dstdomain : Destination c-à-d. le nom de domaine du serveur
- time : Heure du jour et jour de la semaine
- url_regex : Expression régulière décrivant une catégorie d'URL
- urlpath_regex: Expression régulière décrivant un ensemble d'URL sans le protocole ni le nom d'hôte
- proxy_auth : Procédé externe d'authentification d'un utilisateur
- maxconn : Nombre maximum de connexions pour une adresse IP cliente

Pour activer le contrôle, il faut d'abord définir un ensemble d'ACL et ensuite y appliquer des règles. Le format d'une ACL suit la syntaxe

acl acl_element_name type_of_acl_element values_to_acl

Note :

1. acl_element_name peut être n'importe quel nom attribué par l'utilisateur à un élément ACL.
2. Deux éléments distincts ne peuvent avoir le même nom.
3. Chaque ACL est une liste de valeurs. Pendant la vérification, les valeurs multiples utilisent un OU logique. Autrement dit un élément ACL correspond si l'une des valeurs est reconnue.
4. Tous les éléments ACL ne sont pas utilisables avec tous les types de listes d'accès.
5. Différents éléments ACL occupent plusieurs lignes et Squid les amalgame en une seule liste.

Différentes listes d'accès sont disponibles. Celles que nous utiliserons sont décrites ci-dessous

- **http_access:** Autorise les clients HTTP à accèder au port HTTP. C'est l'ACL primaire.
- **no_cache:** Définit le cache pour les réponses aux requêtes

Une règle de liste d'accès comporte les mots allow ou deny; ce qui autorise ou refuse un service pour un élément ACL particulier ou pour un groupe d'éléments.
Note:

1. Les règles sont vérifiées dans l'ordre où elles ont été écrites et se terminent dès qu'une correspondance a été établie.
2. Une ACL peut comporter plusieurs règles.
3. Si aucune correspondance n'est trouvée, l'action par défaut est l'inverse de la dernière règle de la liste; il est donc préférable d'être explicite sur l'action par défaut.
4. Tous les éléments d'une même entrée d'accès sont associés par un ET s'exécutant de la manière suivante
   http_access Action statement1 AND (ET) statement2 AND (ET) statement OR (OU).
   http_access Action statement3
   Des règles multiples de http_access sont comparées par des OU alors que les règles d'une entrée d'accès sont associées par des ET.
5. Rappelez-vous que les règles sont lues de haut en bas.

Retour à la configuration

Par défaut, Squid ne fournit aucun accès aux clients et vous devez modifier les contrôles pour le leur donner. Il faudra au moins une règle pour autoriser l'accès. Dans le fichier squid.conf allez jusqu'à la ligne http_access deny all, et insérez les lignes suivantes au-dessus:

acl mynetwork 192.168.0.0/255.255.255.0
http_access allow mynetwork

mynetwork est le nom ACL et la ligne suivante est la règle qui s'applique. 192.168.0.0 désigne l'adresse réseau dont le masque correspond à 255.255.255. mynetwork est le nom du groupe de machines du réseau et la règle suivante autorise l'accès aux clients. Ces modifications et la définition de la valeur http_port suffisent à rendre Squid opérationnel. Vous pouvez le lancer par la commande

service squid start

Note :

Squid peut aussi être lancé automatiquement au démarrage du système en l'activant dans ntsysv ou setup (System Service Menu). Après toute modification de squid.conf, le processus Squid doit être arrêté, puis redémarré pour que la nouvelle configuration soit prise en compte. Ces deux étapes s'effectuent par les commandes suivantes

1. service squid restart ou
2. /etc/rc.d/init.d/squid restart

Configuration d'une machine cliente

Puisque le service fourni utilise un port particulier du serveur, les machines clientes doivent bien sûr être configurées en conséquence. Nous supposons que ces machines sont déjà connectées sur le réseau local (avec une adresse IP valide) et sont capables d'établir un 'ping' vers le serveur Linux.
Pour Internet Explorer

1. Aller dans Outils -> Options Internet
2. Sélectionner l'onglet Connexion puis cliquer sur Réseau Local
3. Cocher Serveur Proxy et entrer l'adresse IP de celui-ci et le numéro de port sur lequel il écoute les requêtes (http_port address).

Pour Netscape Navigator

1. Aller dans Edition -> Préférences -> Avancées -> Proxies.
2. Cocher le bouton Configuration manuelle du proxy.
3. Cliquer sur le bouton Afficher&
4. Entrer l'adresse IP du serveur proxy et le numéro de port sur lequel il écoute les requêtes (http_port address).

Utiliser le contrôle d'accès

Les contrôles d'accès et les règles multiples offrent une grande souplesse dans le contrôle des accès clients à Internet. Des exemples très courants sont donnés ci-dessous; cela ne signifie pas du tout que ce soient les seuls disponibles.

1. Autoriser l'accès à Internet à certaines machines

   acl allowed_clients src 192.168.0.10 192.168.0.20 192.168.0.30
   http_access allow allowed_clients
   http_access deny !allowed_clients

   Seules les machines dont l'adresse IP est égale à 192.168.0.10, 192.168.0.20 et 192.168.0.30 ont accès à Internet et les autres (dont l'adresse n'est pas listée) se voient refuser ce service.

2. Restreindre l’accès selon les heures et les jours

   acl allowed_clients src 192.168.0.0/255.255.255.0
   acl regular_days time MTWHF 10:00-16:00
   http_access allow allowed_clients regular_days
   http_access deny allowed_clients

   Tous les clients du réseau 192.168.0.0 peuvent accèder à Internet du Lundi au Vendredi de 10h du matin à 4h de l’après-midi.
3. Plusieurs plages horaires selon le client

   acl hosts1 src 192.168.0.10
   acl hosts2 src 192.168.0.20
   acl hosts3 src 192.168.0.30
   acl matin time 10:00-13:00
   acl lunch time 13:30-14:30
   acl soir time 15:00-18:00
   http_access allow host1 matin
   http_access allow host1 soir
   http_access allow host2 lunch
   http_access allow host3 soir
   http_access deny all

   La règle ci-dessus donne l’accès à la machine host1 aux heures du matin et du soir; mais host2 et host3 n’accèdent, respectivement, qu’à l’heure du repas de midi et aux heures du soir.

   Note:
   Tous les éléments figurant dans la même entrée d’accès sont associés par un ET et exécutés de la manière suivante
   http_access Action statement1 AND (ET) statement2 AND (ET) statement OR (OU)

   Des règles multiples de http_access sont comparées par des OU alors que les règles d’une entrée d’accès sont associées par des ET; pour cette raison la règle

   http_access allow host1 matin soir

   ne fonctionnerait jamais (soir ET matin) puisqu’elle ne serait jamais vérifiée.
4. Interdire des sites
   Squid peut interdire l’accès à un site ou des sites qui contiennent tel ou tel mot. Ceci peut être obtenu ainsi

   acl allowed_clients src 192.168.0.1/255.255.255.0
   acl banned_sites url_regex abc.com *()(*.com
   http_access deny banned_sites
   http_access allow allowed_clients

   De la même façon, il est possible de bloquer l’accès à partir d’un mot, par ex. dummy, fake

```
acl allowed_clients src 192.168.0.1/255.255.255.0
acl banned_sites url_regex dummy fake
http_access deny banned_sites
http_access allow allowed_machines
```

Etablir ici la liste des sites et des mots à proscrire n'est guère pratique; cette liste peut être établie séparément (ici dans banned.list dans le répertoire /etc) et le mécanisme ACL viendra lire les informations nécessaires aux blocages désirés.

```
acl allowed_clients src 192.168.0.1/255.255.255.0
acl banned_sites url_regex "/etc/banned.list"
http_access deny banned_sites
http_access allow allowed_clients
```

5. Pour optimiser l'usage
   Squid peut limiter le nombre de connexions d'une machine cliente grâce à l'élément maxconn. Pour l'utiliser, activer d'abord la fonction client_db.

   ```
   acl mynetwork 192.168.0.0/255.255.255.0
   acl numconn maxconn 5
   http_access deny mynetwork numconn
   ```

   Note:
   l'ACL maxconn utilise la comparaison "inférieur à". Elle s'active lorsque le nombre de connexion est plus grand que la valeur donnée. C'est pourquoi elle n'est pas utilisée avec la règle http_access allow.
6. Mise en cache des données
   La réponse à une requête est mise en cache immédiatement, ce qui est excellent dans le cas des pages statiques. Il n'est pas nécessaire de cacher cgi-bin ou Servlet : on utilisera pour cela l'élément ACL no_cache.

   ```
   acl cache_prevent1 url_regex cgi-bin /?
   acl cache_prevent2 url_regex Servlet
   no_cache deny cache_prevent1
   no_cache deny cache_prevent2
   ```

7. Définir vos propres messages d'erreur
   Chaque règle deny offre la possibilité, grâce à l'option deny_info de définir un message d'erreur. Tous les messages par défaut de Squid sont placés dans le répertoire /etc/squid/errors. Le répertoire d'erreurs peut être changé par l'option error_directory. Vous pouvez aussi personnaliser les messages existants.

   ```
   acl allowed_clients src 192.168.0.1/255.255.255.0
   acl banned_sites url_regex abc.com *()(*.com
   http_access deny banned_sites
   deny_info ERR_BANNED_SITE banned_sites
   http_access allow allowed_clients
   ```

Dans cet exemple, une tentative d'accès aux sites bannis entraîne un message spécial. Le fichier de nom ERR_BANNED_SITE doit bien sûr exister dans le répertoire des erreurs. Ce fichier doit être au format HTML. Ces exemples ne montrent qu'une petite partie des possibilités d'ACL; pour en voir d'autres allez sur la Foire Aux Questions du serveur Squid pour d'autres informations et des explications sur les autres éléments ACL.

Fichiers de trace (Log)

Tous les 'logs' de Squid se trouvent dans le répertoire /var/log/squid; il y a des logs pour le cache, les accès et l'utilisation du disque. Le fichier access.log garde trace des requêtes des clients, de leur activitè, et fournit une ligne pour chaque requête HTTP& ICP reçue par le serveur proxy, adresse IP du client, méthode d'interrogation, URL demandée, etc. Les données de ce fichier peuvent être analysées pour disposer d'information sur les accès. Des programmes comme sarg, calamaris, Squid-Log-Analyzer sont disponibles pour analyser ces données et génèrent des rapports (au format HTML). Ces rapports peuvent être établis par utilisateurs, adresses IP, sites visités, etc.

Les options suivantes permettent de modifier la destination de ces fichiers de trace :

```
cache_access_log      pour access.log
cache_log             pour cache.log
cache_store_log       pour store.log (gestion du stockage)
pid_filename          nom indiquant l'ID du processus Squid
```

Méthodes d'authentification

La configuration par défaut de Squid ne demande aucune identification des utilisateurs. Pour authentifier ces derniers, c-à-d. pour n'autoriser que les utilisateurs valides (depuis n'importe quelle machine du réseau) à accéder à Internet, Squid s'appuiera sur un programme externe, pour lequel un nom d'utilisateur et un mot de passe sont requis. Ceci est obtenu par proxy_auth ACL et authenticate_program; qui forcent un utilisateur à donner son nom et son mot de passe avant d'autoriser l'accès. Ces programmes sont nombreux, et parmi eux on trouve

1. LDAP : se base sur le protocole Linux Lightweight Directory Access
2. NCSA : utilise un fichier de noms d'utilisateur et de mots de passe de style NCSA
3. SMB : utilise un serveur SMB comme SAMBA ou Windows NT
4. MSNT : utilise l'authentification de domaine Windows NT
5. PAM : utilise les modules Linux "Pluggable Authentication Modules"
6. getpwam : utilise le fichier passwd de Linux.

Vous indiquerez votre choix grâce à l'option authenticate_program. Assurez-vous d'abord que le programme choisi est installé et fonctionne correctement.

Les modifications dans squid.conf devraient maintenant contenir le même programme d'authentification /usr/local/bin/pam_auth

```
acl pass proxy_auth REQUIRED
acl mynetwork src 192.168.0.1/255.255.255.0
http_access deny !mynetwork
http_access allow pass
http_access deny all
```

Le programme d'authentification PAM est utilisé et chaque utilisateur doit s'authentifier avant d'accéder à Internet.

Des options comme authenticate_ttl et authenticate_ip_ttl permettent aussi de modifier le comportement du processus d'authentification, c-à-d de forcer la revalidation du nom d'utilisateur et du mot de passe.

Références

Cet article se contente d'effleurer la partie visible de l'iceberg Squid; pour plus ample information, visitez ces sites web :

- Squid, www.squid-cache.org
- Documentation du projet Squid, squid-docs.sourceforge.net
- visolve.com
- Sur l'authentification, home.iae.nl/users/devet/squid/proxy_auth

---

Site Web maintenu par l´équipe d´édition LinuxFocus	Translation information: en --> -- : D.S. Oberoi <ds_oberoi/at/yahoo.com> en --> fr: Paul Delannoy <freepol/at/free.fr>

http://www.LinuxFocus.org

2005-01-14, generated by lfparser_pdf version 2.51