TUTOS-INFO
L'informatique par l'exemple

TUTORIEL

Installer SQUID

sur Debian 12

ACL

SOMMAIRE

DIFFICULTE

1 – QU'EST-CE QUE SQUID ?

Un serveur Squid est un serveur mandataire ("proxy") et un mandataire inverse ("reverse-proxy") conçu pour relayer les protocoles FTP, HTTP, Gopher, et HTTPS. Contrairement aux serveurs proxy classiques, un serveur Squid gère toutes les requêtes en un seul processus d'entrée/sortie asynchrone.

La dernière **version stable** en date et utilisée dans ce tutoriel est la version 5.7.2. La réalisation de ce tutoriel suppose que vous disposez d'une machine Debian fonctionnelle avec les accès "root" ou un utilisateur "sudo" (version 12.5 pour ce tutoriel).

Squid fait office de serveur "proxy-cache", c'est-à-dire, qu'il stocke les données fréquemment consultées sur les pages web (notamment les images) sur un serveur cache du réseau local pour éviter de les télécharger à chaque connexion.

De même, il peut mettre en mémoire cache les requêtes DNS. Il permet ainsi de réduire et d'optimiser l'usage de la bande passante vers Internet et du réseau en général, d'ouvrir Internet aux machines situées derrière un pare feu, de restreindre les ressources web utilisables, de contrôler l'utilisation du web par les utilisateurs.

Squid est un "daemon", il tourne donc en tâche de fond. Il écoute en permanence un port ; d'origine, **le port par défaut de Squid est le 3128**. Squid cache les pages visitées, ce qui permet de les charger plus vite lors d'une nouvelle visite car les images ne sont pas rechargées depuis internet mais depuis le serveur (en local). Lorsqu'une page est demandée, Squid vérifie si les informations de la page sont plus récentes que celles qu'il possède en cache. Si c'est le cas, il met à jour son cache et envoie les informations à l'utilisateur.

2 – INSTALLATION DE SQUID SUR DEBIAN 12

L'installation du proxy Squid sur Debian 12 est assez simple.

Dans ce tutoriel, nous travaillons en "root" pour simplifier les commandes mais il est recommandé d'utiliser un utilisateur "sudo" de votre système.

On commence par mettre à jour les paquets de notre machine Debian 12.5 avec les commandes suivantes :

apt update
apt upgrade -y

Le proxy Squid peut être installé directement à partir des dépôts officiels de Debian 12 avec la commande suivante (à faire précéder de "sudo" si vous n'avez pas les accès "root") :

apt install squid -y

3 – MISE EN ŒUVRE ET CONFIGURATION DE SQUID SUR DEBIAN 12

Une fois les paquets Squid installés, il convient de configurer le proxy afin de le mettre en œuvre. Pour cela, plusieurs étapes sont nécessaires. Les fichiers nécessaires au bon fonctionnement du proxy Squid se trouvent à l'emplacement suivant : **"/etc/squid".**

Le fichier "**squid.conf**" est le fichier qui permet de configurer le serveur proxy :

```
root@debian-tests:/etc/squid# ls
conf.d  errorpage.css  squid.conf
```

1ère étape : copie du fichier "squid.conf" (pour sauvegarder le fichier d'origine)

On commence par copier le fichier d'origine en modifiant l'extension afin de le conserver. Ainsi, en cas de problème, il sera simple de reprendre la configuration par défaut. Pour cela, exécutez la commande suivante :

cp /etc/squid/squid.conf squid.conf.backup

On créé ensuite, dans **"/etc/squid"**, un fichier "**squid.conf**" avec la commande suivante :

nano /etc/squid/squid.conf

On colle, dans ce fichier vide, les lignes essentielles suivantes (ces lignes correspondent aux directives par défaut nécessaires au bon fonctionnement du proxy) :

```
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 deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localhost manager
http_access deny manager
http_access allow localhost
http_access deny all
http_port 3128
coredump_dir /var/spool/squid
refresh_pattern ^ftp:      1440    20% 10080
refresh_pattern ^gopher:    1440 0% 1440
refresh_pattern -i (/cgi-bin/|\?) 0    0% 0
refresh_pattern (Release|Packages(.gz)*)$     0      20%     2880
refresh_pattern .      0    20% 4320
cache_mem 256 MB
maximum_object_size 512 MB
```

2ème étape : explication des directives contenues dans le fichier "squid.conf"

Squid utilise les **listes de contrôle d'accès** (**ACL**) pour contrôler qui peut accéder à quelle ressource et effectuer quelle opération. **Par défaut, personne ne peut accéder au proxy HTTP, à l'exception de localhost**.

Le fichier d'origine comporte plus de 7 900 lignes (avec les commentaires) ! Le fichier "squid.conf" que vous avez créé précédemment est nettoyé et débarrassé des commentaires et se présente avec les lignes de configurations utiles par défaut.

Toutes ces lignes correspondent à des directives dont la signification est expliquée page suivante.

La directive "**http_port**" indique l'adresse et le port sur lesquels écoute Squid. Par défaut, **Squid écoute sur toutes les interfaces réseau en utilisant le port 3128**. Vous pouvez configurer plusieurs directives "http_port" pour que Squid écoute sur plusieurs adresses ou ports.

La directive "**coredump_dir**" indique le répertoire dans lequel Squid peut écrire des fichiers "core" qui sont des fichiers d'erreur.
Les directives "**refresh_pattern**" précisent les règles qui établissent si un fichier est "frais" ou "périmé". Un fichier "périmé" est supprimé du cache. Vous pouvez laisser les règles par défaut.

Les directives les plus importantes à connaître sont "acl" et "http_access".

ACL signifie **A**ccess **C**ontrol **L**ist. Les ACL sont donc des **critères de contrôle d'accès qui seront ensuite utilisés par la directive "http_access"** pour autoriser ou interdire les connexions HTTP en fonction de ces critères.

Explication de la syntaxe d'une directive de type "**acl**" :

```
acl Safe_ports port 80
```

- **le 2ème champ représente le nom de l'ACL**. Plusieurs directives ACL avec le même nom vont cumuler les critères de chaque directive.

- **le 3ème champ indique le type d'ACL**. Il existe de nombreux types, mais les plus importants sont "**src**" (source), "**dst**" (destination), "**port**" (numéro du port) et "**time**" (temps).

- **le 4ème champ indique la valeur de l'ACL**. En fonction du type, cette valeur pourra être l'adresse d'un réseau, d'un port, etc.

Dans la configuration par défaut, vous pouvez voir qu'une **ACL "Safe_ports"** est définie. **Elle regroupe un ensemble de ports TCP courants et considérés comme sûrs**, auxquels vous pourrez donner accès à vos utilisateurs.

```
http_access deny !Safe_ports
```

La directive "**http_access**" est suivie de "**allow**" ou "**deny**" pour **autoriser ("allow")** ou **interdire ("deny")** **l'accès au proxy-cache**.

Le dernier champ est le nom de l'ACL qui va être évaluée pour autoriser ou interdire l'accès. Le nom de l'ACL peut être précédé d'un **"!"** qui pourrait être assimilé au mot "sauf".

Par exemple, la directive "**http_access deny !Safe_ports**" signifie "Accès interdit à tous les ports sauf les ports définis dans l'ACL Safe_ports".

Les directives "http_access" sont lues dans l'ordre et sont exécutées dès qu'une règle correspond. Il est donc recommandé de toujours terminer par la directive "http_access deny all" pour interdire l'accès si aucune autre règle ne correspond.

Pour terminer, dans les règles par défaut, vous pouvez voir les règles suivantes :

"**http_access allow localhost manager**" et "**http_access deny manager**" qui n'autorisent l'accès à une interface de gestion du cache qu'en local. Squid est fourni avec un "Cache Manager, qui est une interface web de gestion du cache (nous verrons cela plus tard).

3ème étape : modification du fichier "squid.conf" pour l'adapter à notre infrastructure réseau

On va maintenant ajouter des directives au fichier "squid.conf" afin de l'adapter à notre infrastructure réseau et à notre politique de sécurité. Pour cela, on édite le fichier "squid.conf" dans lequel on avait copié les directives essentielles et on lui ajoute les directives qui suivent :

a) On restreint l'accès à Squid en indiquant l'IP du serveur Squid et on indique le port d'écoute :

http_port 192.168.100.108:3128 ***# attention, adaptez l'adresse IP en fonction de votre configuration !***

Ici, nous avons laissé le port d'écoute par défaut de Squid mais vous pouvez le modifier (souvent, le port "8080" est utilisé pour le proxy).

b) Autorisation du réseau "LAN" à utiliser le proxy :

acl networklan src 192.168.100.0/24 ***# attention, adaptez l'adresse à votre réseau local LAN !***

c) Permettre l'accès Internet à mon réseau LAN :

http_access allow networklan

Nous devons obtenir ceci avec les 3 modifications :

```
acl SSL_ports port 443
acl Safe_ports port 80      # http
acl Safe_ports port 21      # ftp
acl Safe_ports port 443     # https
acl Safe_ports port 70      # gopher
acl Safe_ports port 210     # wais
acl Safe_ports port 1025-65535   # unregistered ports
acl Safe_ports port 280     # http-mgmt
acl Safe_ports port 488     # gss-http
acl Safe_ports port 591     # filemaker
acl Safe_ports port 777     # multiling http
acl CONNECT method CONNECT
acl networklan src 192.168.100.0/24
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localhost manager
http_access deny manager
http_access allow localhost
http_access allow networklan
http_access deny all
http_port 192.168.100.108:3128
coredump_dir /var/spool/squid
refresh_pattern ^ftp:      1440    20% 10080
refresh_pattern ^gopher:    1440 0% 1440
refresh_pattern -i (/cgi-bin/|\?) 0    0% 0
refresh_pattern (Release|Packages(.gz)*)$      0      20%    2880
refresh_pattern .      0    20% 4320
cache_mem 256 MB
maximum_object_size 512 MB
```

4ème étape : activation du proxy Squid

a) Activation du service au démarrage de la machine Debian (serveur Squid) :

systemctl enable squid

b) Démarrage du service Squid :

systemctl start squid

c) Arrêt du service Squid (en cas de modifications du fichier de configuration) :

systemctl stop squid

d) Redémarrage du service Squid (en cas de modification du fichier de configuration) :

systemctl restart squid

e) Vérification du statut du service Squid :

systemctl status squid

4 – CONFIGURATION MANUELLE DU PROXY DANS LE NAVIGATEUR D'UNE MACHINE CLIENTE

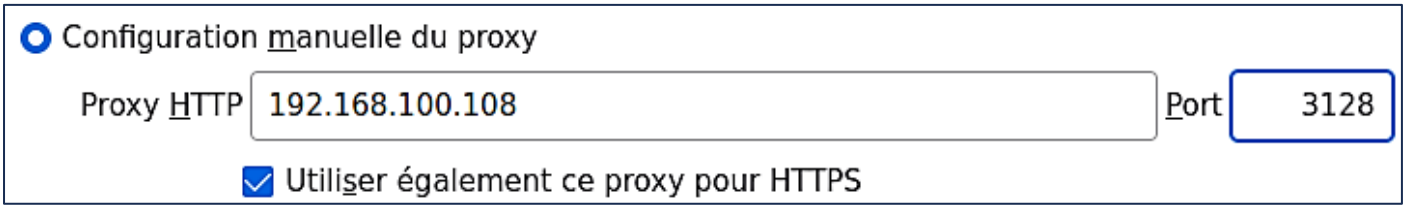
Le serveur proxy étant configuré et démarré, nous allons maintenant tester son fonctionnement depuis une machine cliente du réseau local autorisé. **Nous allons, dans un premier temps, configurer le proxy dans le navigateur de la machine cliente** (nous utiliserons Firefox dans ce tutoriel).

Ouvrez le navigateur (Firefox ici) et effectuez les manipulations suivantes :

- Cliquez, en haut à droite, sur l'icône "**Paramètres**" 
- Dans le volet de gauche, cliquez sur "**Général**" 

- Dans le bas de la fenêtre, cliquez le bouton "**Paramètres**" de la rubrique "**Paramètres réseau**" :

- Cliquez "**Configuration manuelle du proxy**" et indiquez les paramètres de votre serveur Squid :



- Cliquez le bouton "**Ok**"

5 – BLOCAGE DE SITES ET/OU DE DOMAINES AVEC DES ACL

Le serveur proxy Squid peut bloquer l'accès à certains sites ou domaines en fonction de votre politique de navigation sur le web au sein de votre organisation.

Pour information les "logs" du proxy Squid sont enregistrés dans le fichier "**access.log**" à l'emplacement :

/var/log/squid/access.log

Avec la commande "**tail /var/log/squid/acces.log**", on voit, par exemple, que la machine de notre réseau LAN avec l'IP 192.168.100.109 s'est connectée au site web Youtube :

```
1711989878.825  14088 192.168.100.109 TCP_TUNNEL/200 723366 CONNECT www.gstatic.com:443 - HIER_DIRECT/216.58.215.35 -
1711989878.825  14243 192.168.100.109 TCP_TUNNEL/200 4349796 CONNECT www.youtube.com:443 - HIER_DIRECT/216.58.214.174 -
1711989878.825  14357 192.168.100.109 TCP_TUNNEL/200 6586 CONNECT encrypted-tbn0.gstatic.com:443 - HIER_DIRECT/142.250.7
```

Il est possible de voir les logs en temps réel avec la commande suivante :

tail -f /var/log/squid/access/log

Nous souhaitons bloquer l'accès à Youtube et Facebook au sein du réseau LAN. Pour cela, il existe plusieurs manières de procéder. Nous allons présenter une 1ère méthode avec inscription de nouvelles ACL directement dans le fichier de configuration de Squid.

a) Ajout d'ACL de blocage dans le fichier de configuration "squid.conf" :

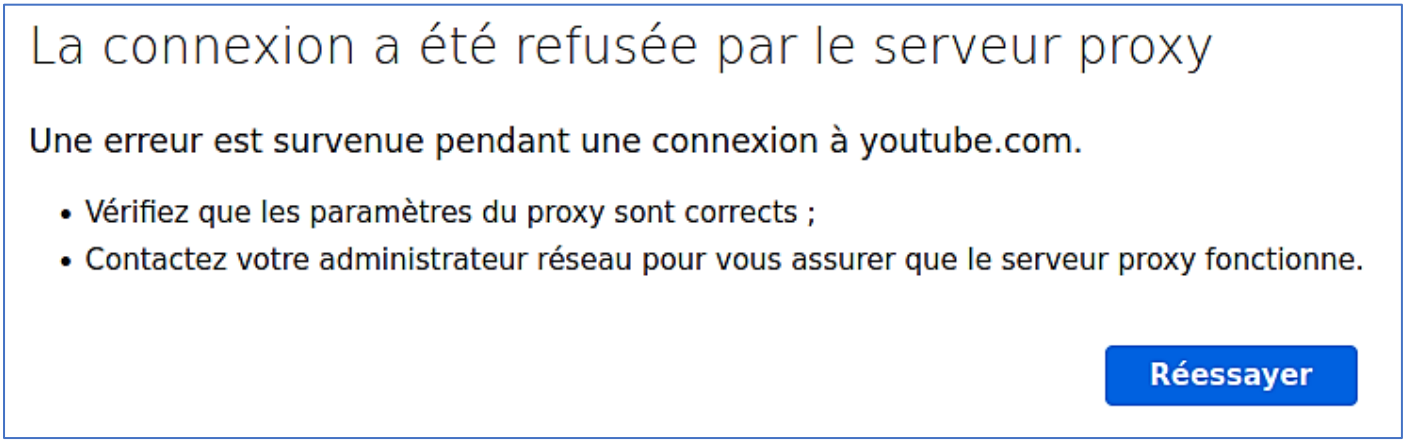
nano /etc/squid/squid.conf

Voici les modifications à apporter pour bloquer Youtube et Facebook :

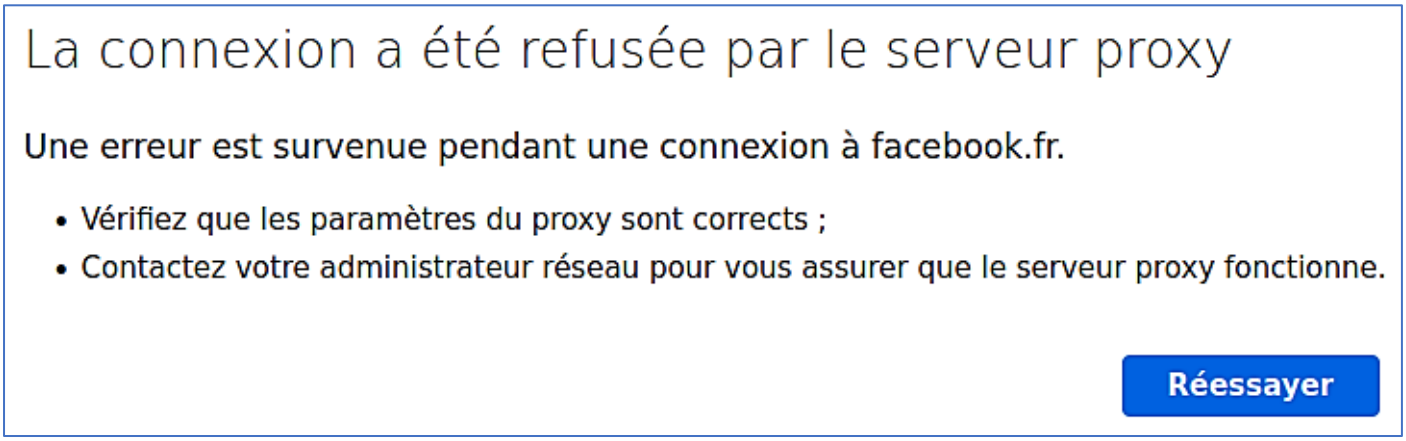
```
acl SSL_ports port 443
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443          # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210          # wais
acl Safe_ports port 1025-65535    # unregistered ports
acl Safe_ports port 280          # http-mgmt
acl Safe_ports port 488          # gss-http
acl Safe_ports port 591          # filemaker
acl Safe_ports port 777          # multiling http
acl CONNECT method CONNECT
acl networklan src 192.168.100.0/24

# ACL de blocage de sites
acl blockYoutube  dstdomain .youtube.com
acl blockFacebookFR dstdomain .facebook.fr
acl blockFacebookCOM dstdomain .facebook.com

# Refuser l'accès aux sites interdits
http_access deny blockYoutube
http_access deny blockFacebookFR
http_access deny blockFacebookCOM

http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localhost manager
http_access deny manager
http_access allow localhost
http_access allow networklan

http_access deny all
```

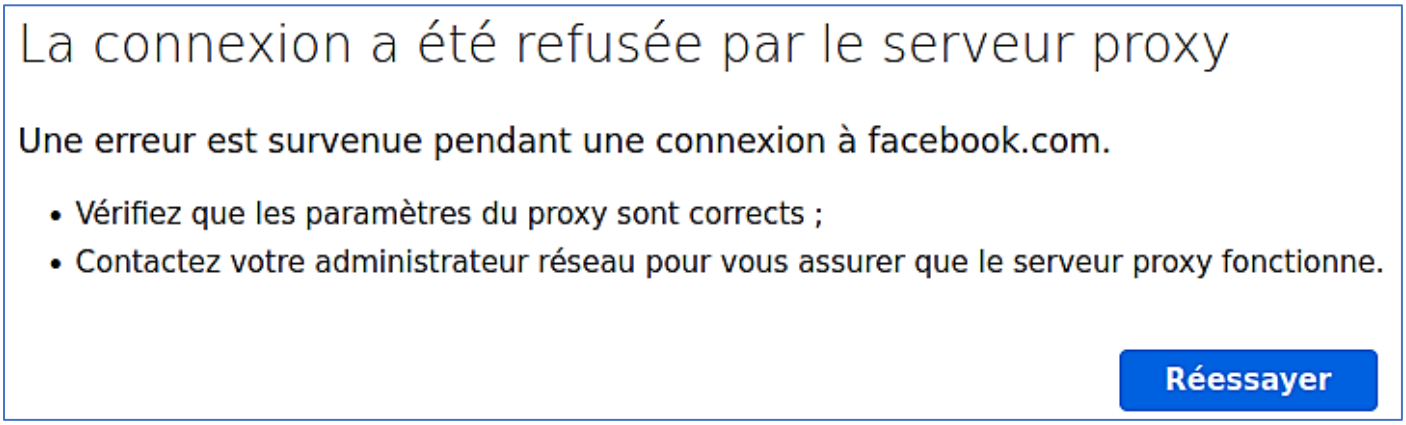
```
# ACL de blocage de sites
acl blockYoutube dstdomain .youtube.com
acl blockFacebookFR dstdomain .facebook.fr
acl blockFacebookCOM dstdomain .facebook.com

# Refuser l'accès aux sites interdits
http_access deny blockYoutube
http_access deny blockFacebookFR
http_access deny blockFacebookCOM
```

On teste, sur la machine cliente, le blocage de "youtube.com" :



On teste la connexion à Facebook (".fr" et ".com") :





Les "logs" nous indiquent que l'accès a bien été refusé avec l'erreur 403 ("**denied**") :

```
1711991656.251      0 192.168.100.109 TCP_DENIED/403 4057 CONNECT youtube.com:443 - HIER_NONE/- text/html
1711991706.661     10 192.168.100.109 TCP_MISS/200 489 GET http://detectportal.firefox.com/canonical.html - HIER_DIRECT/
34.107.221.82 text/html
1711991706.671      7 192.168.100.109 TCP_MISS/200 387 GET http://detectportal.firefox.com/success.txt? - HIER_DIRECT/34
.107.221.82 text/plain
1711991706.678     15 192.168.100.109 TCP_MISS/200 387 GET http://detectportal.firefox.com/success.txt? - HIER_DIRECT/34
.107.221.82 text/plain
1711991714.731      0 192.168.100.109 TCP_DENIED/403 4305 GET http://facebook.fr/ - HIER_NONE/- text/html
1711991714.779      0 192.168.100.109 TCP_HIT/200 13108 GET http://debian-tests.gnommet.info:3128/squid-internal-static/
icons/SN.png - HIER_NONE/- image/png
1711991714.788      0 192.168.100.109 TCP_DENIED/403 4272 GET http://facebook.fr/favicon.ico - HIER_NONE/- text/html
1711991729.383      0 192.168.100.109 TCP_DENIED/403 4057 CONNECT facebook.fr:443 - HIER_NONE/- text/html
1711991757.981      0 192.168.100.109 TCP_DENIED/403 4060 CONNECT facebook.com:443 - HIER_NONE/- text/html
```

b) Autoriser l'accès à certains sites en fonction des jours et de l'horaire :

Squid permet de restreindre l'accès à des sites en fonction des jours et des horaires souhaités. Pour cela, on procède ainsi en éditant le fichier de configuration "**squid.conf**" :

Par exemple, nous souhaitons **accorder l'accès à Facebook uniquement pendant la pause du midi (de 12 h à 14 h) du lundi au vendredi uniquement**. Pour cela, on ajoute 2 nouvelles ACL dans le fichier "**squid.conf**" :

nano /etc/squid/squid.conf

L'ACL comporte le type "**time**" et indique les jours de la semaine sous la forme "**M T W T F**" avec l'horaire qui est autorisé dans notre cas "**12:00-14:00**" :

acl FacebookFR_planning time M T W T F 12:00-14:00
http_access allow FacebookFR_planning

```
# ACL de blocage de sites
acl blockYoutube  dstdomain .youtube.com
acl blockFacebookFR dstdomain .facebook.fr
acl FacebookFR_planning time M T W T F 12:00-14:00
acl blockFacebookCOM dstdomain .facebook.com

# Refuser l'accès aux sites interdits
http_access deny blockYoutube
http_access deny blockFacebookFR
http_access deny blockFacebookCOM

# Facebook autorisé de 12 h à 14 h du lundi au vendredi
http_access allow FacebookFR_planning
```

On redémarre le proxy (il ne doit pas y avoir d'erreur renvoyée) :

systemctl restart squid

Compte tenu de l'heure ici (voir en bas à droite : 19 h 40), l'accès à Facebook reste non autorisé :

c) Blocage de plusieurs sites listés dans un fichier :

Il est possible, également, de bloquer plusieurs sites contenus dans un fichier présent dans le répertoire **"/etc/squid"**. Pour cela, effectuez les manipulations suivantes :

- Créez le fichier "**block_domaines.txt**" dans **"/etc/squid"** :

nano /etc/squid/block_domaines.txt

- Saisissez, sur une ligne à la fois, les domaines que vous souhaitez interdire :

tutos-info.fr
ndlaprovidence.fr
ecoledirecte.com

- Quittez et sauvegardez le fichier
- Modifiez votre fichier "**squid.conf**" :

nano /etc/squid.conf

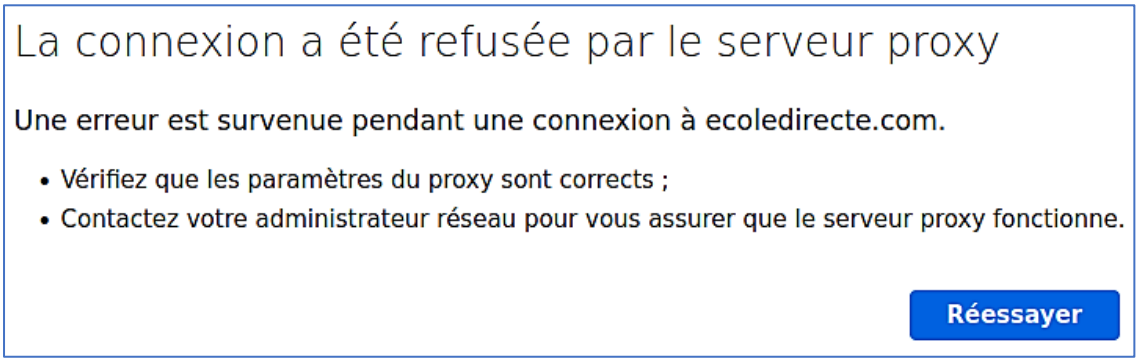
On ajoute l'ACL "**url_regex -i**" comme ceci :

acl deny_domain url_regex -i "/etc/squid/block_domaines.txt"
http_access deny deny_domain

On relance Squid :

systemctl restart squid

Ici, la connexion à https://ecoledirecte.com a bien été bloquée :



Note :

Lorsque l'on met cette l'option "url_regex" dans une ACL, cela permet de rechercher une chaîne de caractères contenu dans l'URL. À noter : l'url_regex est sensible à la casse (majuscules / minuscules).

6 – BLOCAGE DE CERTAINES EXTENSIONS DE FICHIERS

Il est possible, également, de bloquer certaines extensions de fichiers en créant un fichier dans le répertoire **"/etc/squid"**. Pour cela, effectuez les manipulations suivantes :

- Créez le fichier "**block_extensions.conf**" dans **"/etc/squid"** :

nano /etc/squid/block_extensions.conf

- Saisissez, sur une ligne à la fois, les extensions que vous souhaitez interdire :

.exec
.mp4
.mp3
.zip
.pdf

- Quittez et sauvegardez le fichier
- Modifiez votre fichier "**squid.conf**" :

nano /etc/squid.conf

On ajoute l'ACL "**urlpath_regex -i**" comme ceci :

acl EXTENSIONS urlpath_regex -i "/etc/squid/block_extensions.conf"
http_access deny EXTENSIONS

```
# ACL de blocage des extensions
acl EXTENSIONS urlpath_regex -i "/etc/squid/block_extensions.conf"
http_access deny EXTENSIONS
```

On relance Squid :

systemctl restart squid

On pourrait également procéder ainsi pour bloquer des extensions **".avi**" directement dans le fichier "**squid.conf**" (lignes à ajouter) :

 # Bloquer les fichiers AVI
acl extension_avi url_regex –i \.avi$
http_access deny extension_avi

On relance Squid :

systemctl restart squid

7 – BLOCAGE DE CERTAINS SITES AVEC DES MOTS-CLES

Il est possible de restreindre l'accès à certains sites avec des mots-clés. Pour cela, procédez ainsi :

- Créez le fichier "**block_keywords.conf**" dans **"/etc/squid"** :

nano /etc/squid/block_keywords.conf

- Saisissez, sur une ligne à la fois, les mots-clés que vous souhaitez interdire :

adult
sexe
porn

- Quittez et sauvegardez le fichier
- Modifiez votre fichier "**squid.conf**" :

nano /etc/squid.conf

On ajoute l'ACL "**urlpath_regex -i**" comme ceci :

acl block_keywords url_regex -i "/etc/squid/block_keywords.conf"
http_access deny block_keywords

```
# ACL de blocage de sites par mots-clés
acl block_keywords url_regex -i "/etc/squid/block_keywords.conf"
http_access deny block_keywords
```

On relance Squid :

systemctl restart squid

8 – RESTREINDRE L'ACCES A INTERNET

Il est possible, également, de restreindre l'accès à Internet ajoutant une ACL fichier dans le fichier "**squid.conf**". Pour cela, effectuez les manipulations suivantes :

- Editez le fichier "**squid.conf**" dans **"/etc/squid"** :

nano /etc/squid/squid.conf

- Saisissez l'ACL suivante, quittez et sauvegardez le fichier :

acl JOURS_HEURES_OUVRES time M T W T F 09:00-18:00
http_access allow JOURS_HEURES_OUVRES

```
# ACL d'autorisation Internet (jours ouvrés de la semaine)
acl JOURS_HEURES_OUVRES time M T W T F 09:00-18:00
http_access allow JOURS_HEURES_OUVRES
```

On relance Squid :

systemctl restart squid

La connexion Internet est suspendue en-dehors des heures autorisées :