TP Proxy ASSR - Configuration des ACL dans Squid

Date : 13 janvier 2026

Système : Debian 13

Service : Squid Proxy Server

Editeur : Luka Valencelle

Table des matières

1. Objectif du TP

Configurer un serveur proxy Squid sur Debian en définissant des ACL (Access Control Lists) pour contrôler l'accès au proxy selon les adresses IP et réseaux. Identifier et corriger les erreurs de configuration.

2. Configuration réalisée

2.1. Définition des ACL

Les ACL suivantes ont été définies dans le fichier /etc/squid/squid.conf :

1. groupe0 : Correspond à l'IP du poste client (172.20.10.3/32 ip du serveur proxy sur lequel les testes on été réalisé)
2. groupe1 : IP unique (10.10.10.10/32)
3. groupe2 : Réseau 254.180.1.0/24 (corrigé depuis 256.180.1.0)
4. groupe3 : Réseau 172.16.10.0/24

```
#ACl TP
acl group0 src 172.20.10.3/32
acl group1 src 10.10.10.10/32
acl group2 src 254.180.1.0/24
acl group3 src 172.16.10.0/24
```

2.2. Code de configuration

Port d'écoute Squid (par défaut : 3128)
http_port 8080

=== ACL TP ===
acl groupe0 src 172.20.10.3/32
acl groupe1 src 10.10.10.10/32
acl groupe2 src 254.180.1.0/24
acl groupe3 src 172.16.10.0/24

=== Règles d'accès ===
http_access allow groupe0
http_access allow groupe1
http_access allow groupe2
http_access deny groupe3
http_access deny all

```
#Config TP ASSR
http_port 8080
#ACl TP
acl group0 src 172.20.10.3/32
acl group1 src 10.10.10.10/32
acl group2 src 254.180.1.0/24
acl group3 src 172.16.10.0/24
#Règle d'accès
http_access allow group0
http_access allow group1
http_access allow group2

http_access deny group3
http_access deny all
```

```
#Règle d'accès
http_access allow group0
http_access allow group1
http_access allow group2

http_access deny group3
http_access deny all
```

2.3. Autorisations d'accès configurées

- Groupes autorisés : groupe0, groupe1, groupe2
- Groupe non-autorisé : groupe3
- Tous les autres accès : bloqués par la règle "http_access deny all"

```
#Règle d'accès
http_access allow group0
http_access allow group1
http_access allow group2

http_access deny group3
http_access deny all
```

Les règles sont lu du haut vers le bas, ainsi ce qui n'est pas explicitement autorisé est bloqué par la dernière règle.

3. Réponses aux questions du TP

Question 1 : Que devez-vous ajouter pour bloquer tout ce qui n'est pas explicitement autorisé ?

Réponse :

http_access deny all

Cette directive doit être placée obligatoirement en dernière position après toutes les règles d'autorisation. Elle bloque tous les accès qui n'ont pas été explicitement autorisés par les règles précédentes.

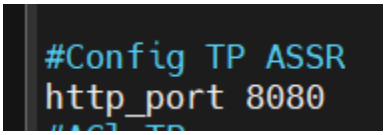
```
http_access deny all
```

Question 2 : Quel est le port d'écoute par défaut de Squid ?

Réponse : 3128

Pour ce TP, le port a été modifié en 8080 via la directive :

http_port 8080



Question 3 : Quelles sont les erreurs rencontrées ? Comment les corriger ?

Erreur 1 : Adresse IP invalide

Problème : L'adresse réseau 256.180.1.0/24 fournie dans l'énoncé est invalide car une adresse IPv4 ne peut pas dépasser 255 et les réseaux en 255 sont réservés pour les brodcasts.

Correction : L'adresse a été corrigée en 254.180.1.0/24 pour respecter les contraintes du protocole IPv4.

Erreur 2 : Ordre des règles http_access

Problème : Les règles par défaut de Squid (notamment http_access allow localhost et les restrictions Safe_ports) bloquaient les ACL personnalisées et empêchaient le bon fonctionnement du proxy.

Correction : Les règles restrictives par défaut ont été commentées temporairement pour permettre l'application des ACL du TP. En production, ces règles devraient être adaptées plutôt que désactivées.

```
#
# Deny requests to certain unsafe ports
#http_access deny !Safe_ports

# Deny CONNECT to other than secure SSL ports

#ttp_access deny CONNECT !SSL_ports

# Only allow cachemgr access from localhost
#http_access allow localhost manager
#http_access deny manager

# This default configuration only allows localhost requests because a more
# permissive Squid installation could introduce new attack vectors into the
# network by proxying external TCP connections to unprotected services.
#http_access allow localhost

# The two deny rules below are unnecessary in this default configuration
# because they are followed by a "deny all" rule. However, they may become
# critically important when you start allowing external requests below them.

# Protect web applications running on the same server as Squid. They often
# assume that only local users can access them at "localhost" ports.
#http_access deny to_localhost

# Protect cloud servers that provide local users with sensitive info about
# their server via certain well-known link-local (a.k.a. APIPA) addresses.
```

4. Tests et validation

4.1. Vérification de la configuration

La syntaxe du fichier de configuration a été vérifiée avec la commande :

sudo squid -k check

Résultat : Syntax OK

```
root@deiban-sql-srv:~# ^C
root@deiban-sql-srv:~# ^C
root@deiban-sql-srv:~# nano /etc/squid/squid.conf
root@deiban-sql-srv:~# squid -k check
2026/01/13 12:10:38| Processing Configuration File: /etc/squid/squid.conf (depth
 0)
2026/01/13 12:10:38| Processing Configuration File: /etc/squid/conf.d/debian.con
f (depth 1)
2026/01/13 12:10:38| Set Current Directory to /var/spool/squid
root@deiban-sql-srv:~#
```

4.2. Redémarrage du service

Le service Squid a été redémarré avec succès :

sudo systemctl restart squid

```
● squid.service - Squid Web Proxy Server
     Loaded: loaded (/usr/lib/systemd/system/squid.service; enabled; preset: enabled)
     Active: active (running) since Tue 2026-01-13 12:12:56 CET; 50s ago
```

4.3. Vérification du port d'écoute

Confirmation que Squid écoute bien sur le port 8080 :

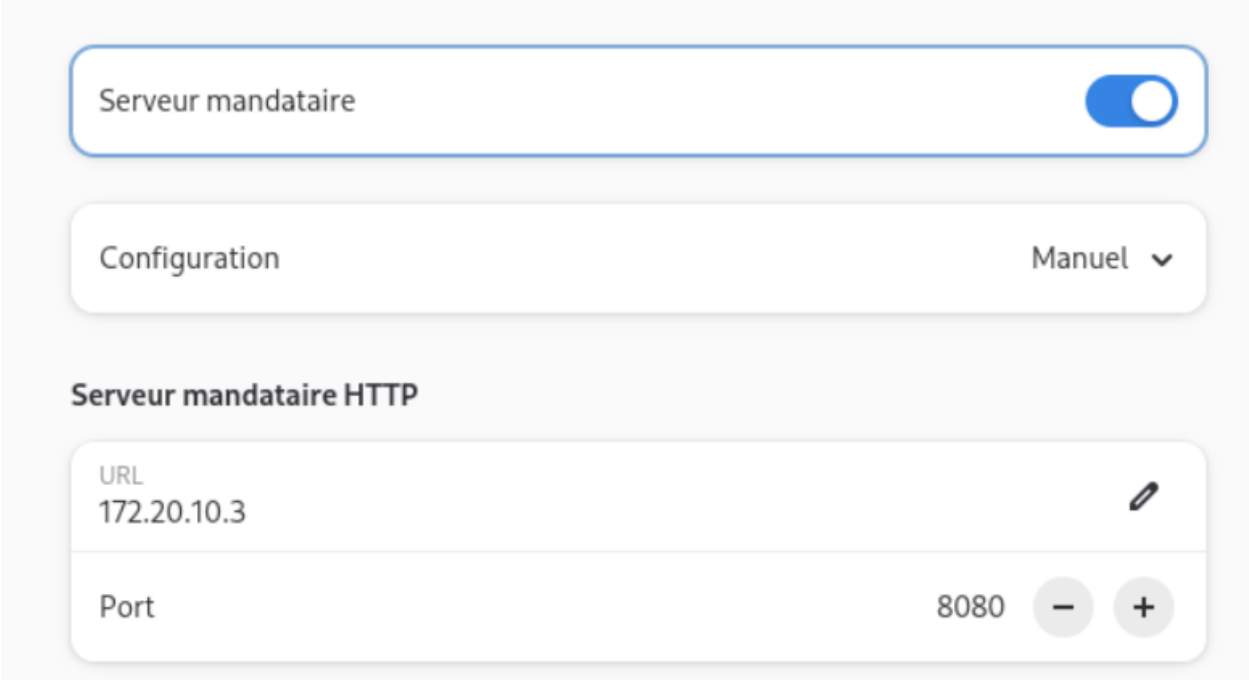
ss -lntp | grep squid

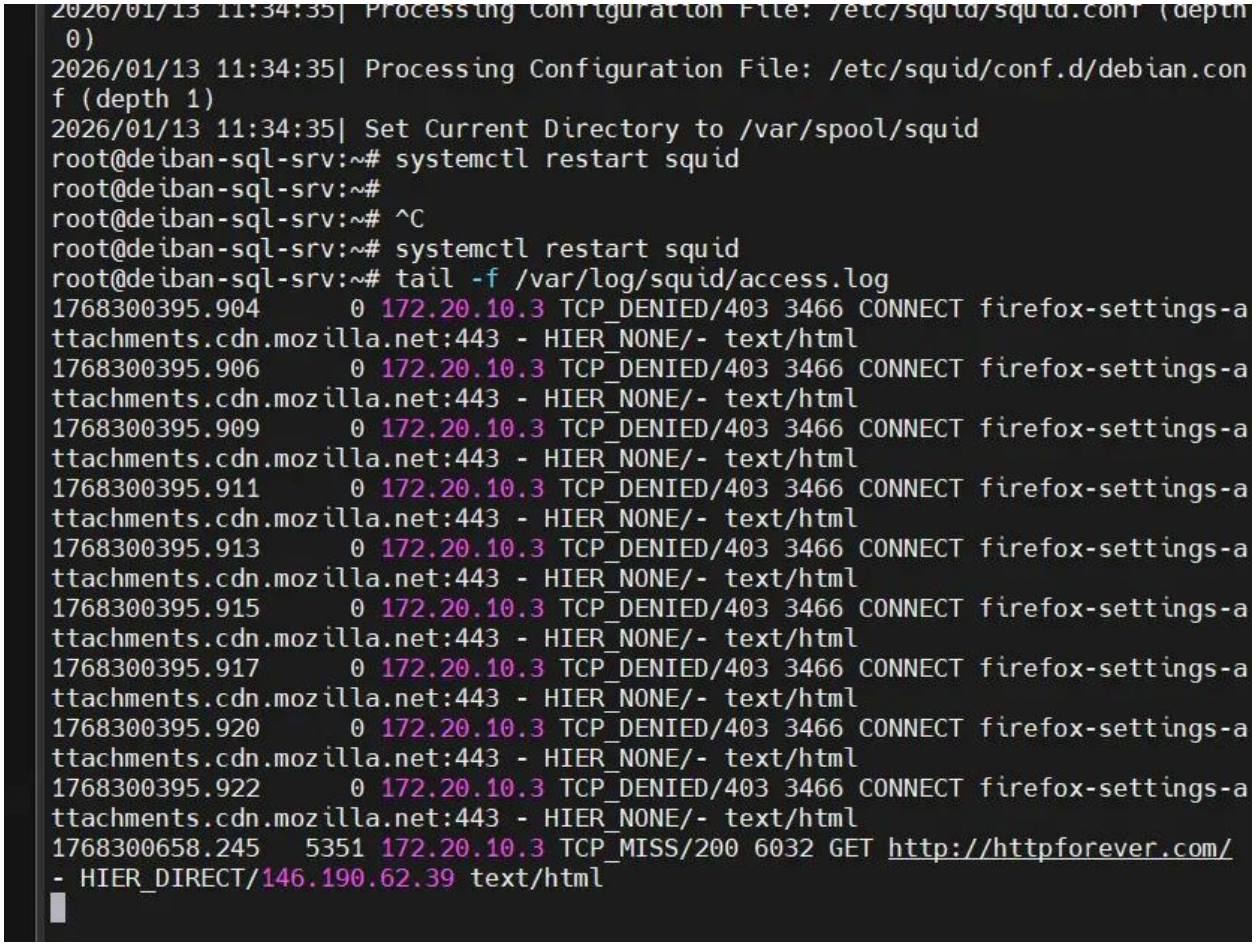
```
root@deiban-sql-srv:~# ss -lntp | grep squid
LISTEN 0      256                *:8080            *:*    users:(("squid",pid=5919,fd=13))
```

4.4. Tests de connexion

Tests effectués depuis le serveur proxy(172.20.10.3) :

- Configuration du proxy dans les paramètres de la carte réseau : IP du serveur Debian, port 8080
- Accès à http://httpforever.com (site HTTP non chiffré)
- Vérification des logs avec : tail -f /var/log/squid/access.log





4.5. Résultats des tests

Résultat observé dans les logs :

172.20.10.3 TCP_MISS/200 6032 GET http://httpforever.com/

Conclusion : Le proxy fonctionne correctement et autorise bien les connexions du groupe0.

5. Partie 5 : Configuration du filtrage et modification d'en-têtes

5.1. Objectifs

- Configurer Squid pour bloquer l'accès à des sites spécifiques
- Modifier les en-têtes HTTP des requêtes

5.2. Blocage de facebook.com

Configuration mise en place :

Blocage de Facebook
acl facebook dstdomain .facebook.com
http_access deny facebook

```
#ACl TP
acl group0 src 172.20.10.3/32
acl group1 src 10.10.10.10/32
acl group2 src 254.180.1.0/24
acl group3 src 172.16.10.0/24
acl facebook dstdomain .facebook.com
#Règle d'accès
http_access deny facebook
http_access allow group0
http_access allow group1
http_access allow group2

http_access deny group3
http_access deny all
```

Explications :

- acl facebook dstdomain .facebook.com : Définit une ACL de destination pour Facebook
- Le point (.) devant facebook.com capture aussi tous les sous-domaines (www.facebook.com, m.facebook.com, etc.)
- http_access deny facebook : Bloque l'accès à cette destination

Ordre des règles **:** Il est crucial de placer la règle de blocage avant les règles d'autorisation générales. Squid lit les règles de haut en bas et applique la première règle correspondante.

5.3. Modification du User-Agent

Configuration mise en place :

Modification du User-Agent
request_header_replace User-Agent LukaPSTB2025/1.0

Cette directive modifie automatiquement le champ "User-Agent" de toutes les requêtes HTTP passant par le proxy.

5.4. Configuration finale complète

```
#Config TP ASSR

http_port 8080

#Modifier le user agent

request_header_replace User-Agent LukaPSTB2025/1.0

#ACl TP

acl group0 src 172.20.10.3/32

acl group1 src 10.10.10.10/32

acl group2 src 254.180.1.0/24

acl group3 src 172.16.10.0/24

acl facebook dstdomain .facebook.com

#Règle d'accès

http_access deny facebook

http_access allow group0

http_access allow group1

http_access allow group2


http_access deny group3

http_access deny all
```

5.5. Tests et validation

Test 1 : Blocage de Facebook

Procédure : Accès à http://facebook.com depuis le navigateur client

Résultat : Accès refusé

Le navigateur affiche le message : "La connexion a été refusée par le serveur proxy"

Logs Squid :

172.20.10.3 TCP_DENIED/403 3402 CONNECT facebook.com:443

Analyse :

- TCP_DENIED/403 : Requête refusée par le proxy
- facebook.com:443 : Tentative d'accès à Facebook en HTTPS
- Le blocage fonctionne correctement

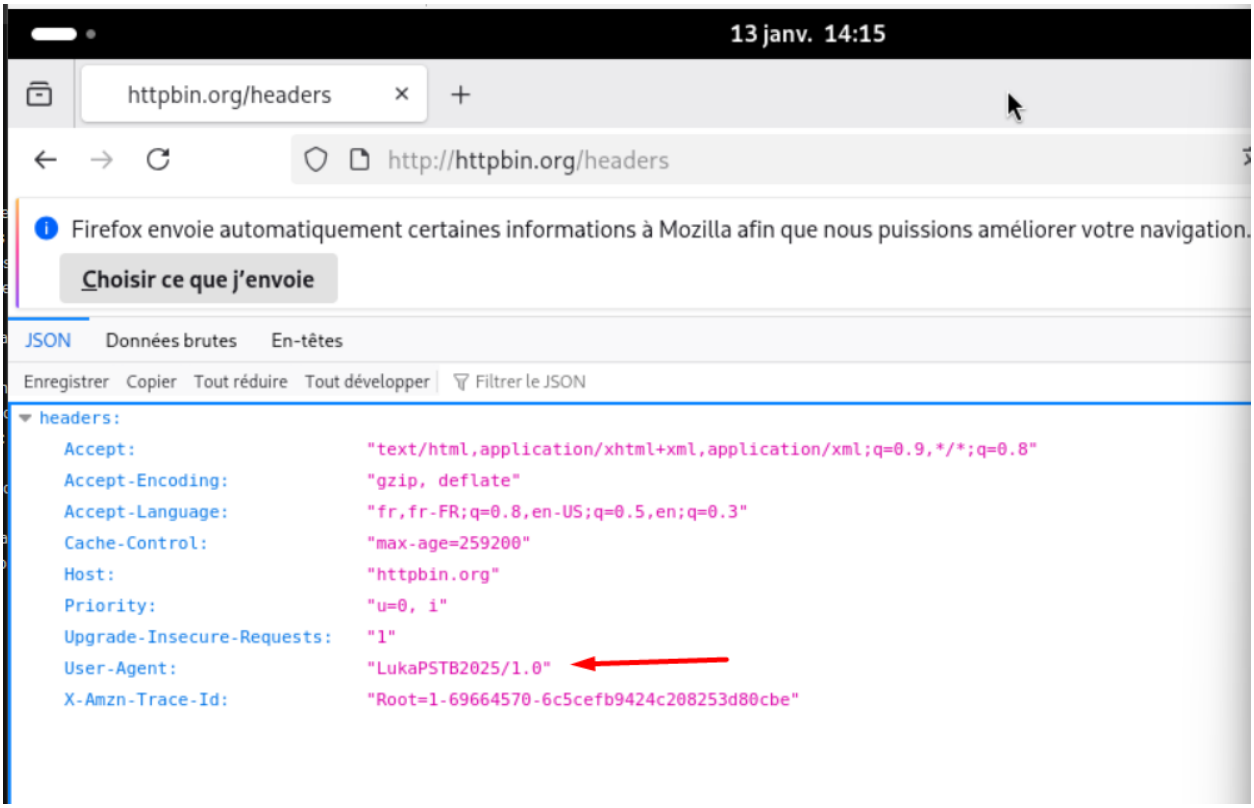
Test 2 : Modification du User-Agent

Procédure : Accès à http://httpbin.org/headers pour visualiser les en-têtes HTTP

Résultat attendu : Le champ "User-Agent" doit afficher "LukaPSTB2025/1.0"

Résultat obtenu : Le User-Agent a bien été modifié

Le champ User-Agent des requêtes HTTP a été remplacé avec succès par la valeur personnalisée.



6. Question bonus

Comment autoriser toutes les machines sans restriction ?

Réponse :

http_access allow all

7. Configuration et utilisation de SquidGuard

1. Objectif

Configurer SquidGuard comme outil de filtrage avancé permettant de bloquer des catégories complètes de sites web selon des listes prédéfinies et des règles par groupe.

2. Installation de SquidGuard

sudo apt update
sudo apt install squidguard -y

3. Intégration dans Squid

Ajout des directives suivantes dans /etc/squid/squid.conf :

url_rewrite_program /usr/bin/squidGuard -c /etc/squid/squidguard.conf
url_rewrite_children 10

Explication : Ces directives indiquent à Squid d'utiliser SquidGuard comme redirecteur d'URL pour le filtrage avancé.

4. Structure des listes de blocage

Création de la structure de répertoires :

sudo mkdir -p /var/lib/squidguard/db/adult
sudo mkdir -p /var/lib/squidguard/db/mail
sudo mkdir -p /var/lib/squidguard/db/perso
sudo chown -R proxy:proxy /var/lib/squidguard
sudo chmod -R 755 /var/lib/squidguard

5. Listes de blocage configurées

- Liste personnelle (perso/domains) : youtube.com, chess.com
- Liste mail (mail/domains) : Liste fourni par le professeur
- Liste adult (adult/domains) : Liste fourni par le professeur

6. Configuration squidguard.conf

```
# Répertoires
dbhome /var/lib/squidguard/db
logdir /var/log/squidguard

# SOURCES
src Lukaadmin {
   ip 172.20.10.3/32
}
```

```
src groupe1 {
   ip 10.10.10.10/32
}
src groupe2 {
   ip 254.180.1.0/24
}

# DESTINATIONS
dest adult {
   domainlist adult/domains
}
dest mail {
   domainlist mail/domains
}
dest perso {
   domainlist perso/domains
}

# ACL (Règles)
acl {
   Lukaadmin {
      pass !perso all
      redirect https://fr.wikipedia.org/wiki/Mauvais
   }
   groupe1 {
      pass mail
      redirect https://fr.wikipedia.org/wiki/Mauvais
   }
   groupe2 {
      pass !adult all
      redirect https://fr.wikipedia.org/wiki/Mauvais
   }
   default {
      pass none
      redirect https://fr.wikipedia.org/wiki/Mauvais
   }
}
```

8. Règles de filtrage appliquées

Lukaadmin (172.20.10.3) : Accès à tout sauf youtube.com et chess.com

groupe1 (10.10.10.10) : Accès uniquement aux sites de mail

groupe2 (254.180.1.0/24) : Accès à tout sauf sites adultes

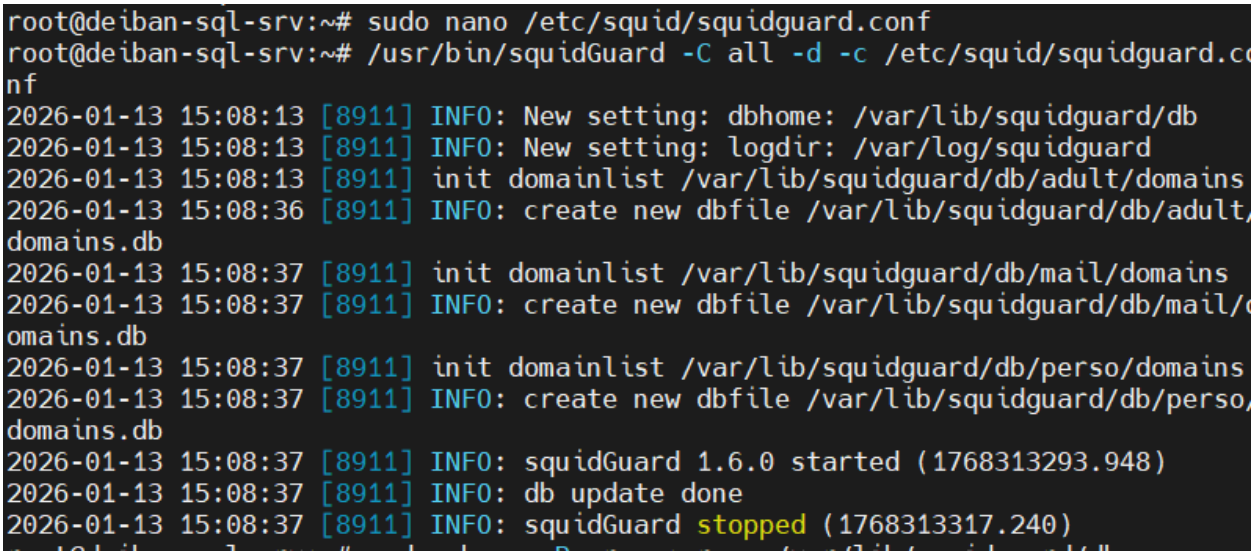
Par défaut : Tout bloqué

```
# === ACL ===

acl {
    Lukaadmin {
        pass !perso all
        redirect https://fr.wikipedia.org/wiki/Mauvais
    }

    groupe1 {
        pass mail
        redirect https://fr.wikipedia.org/wiki/Mauvais
    }

    groupe2 {
        pass !adult all
        redirect https://fr.wikipedia.org/wiki/Mauvais
    }

    default {
        pass none
        redirect https://fr.wikipedia.org/wiki/Mauvais
    }
}
```

8.1 Compilation des bases de données

sudo /usr/bin/squidGuard -C all -d -c /etc/squid/squidguard.conf
sudo chown -R proxy:proxy /var/lib/squidguard/db
sudo chmod -R 755 /var/lib/squidguard/db

Cette commande compile les listes de domaines en bases de données optimisées (.db) pour un filtrage rapide.



8.2 Tests et validation

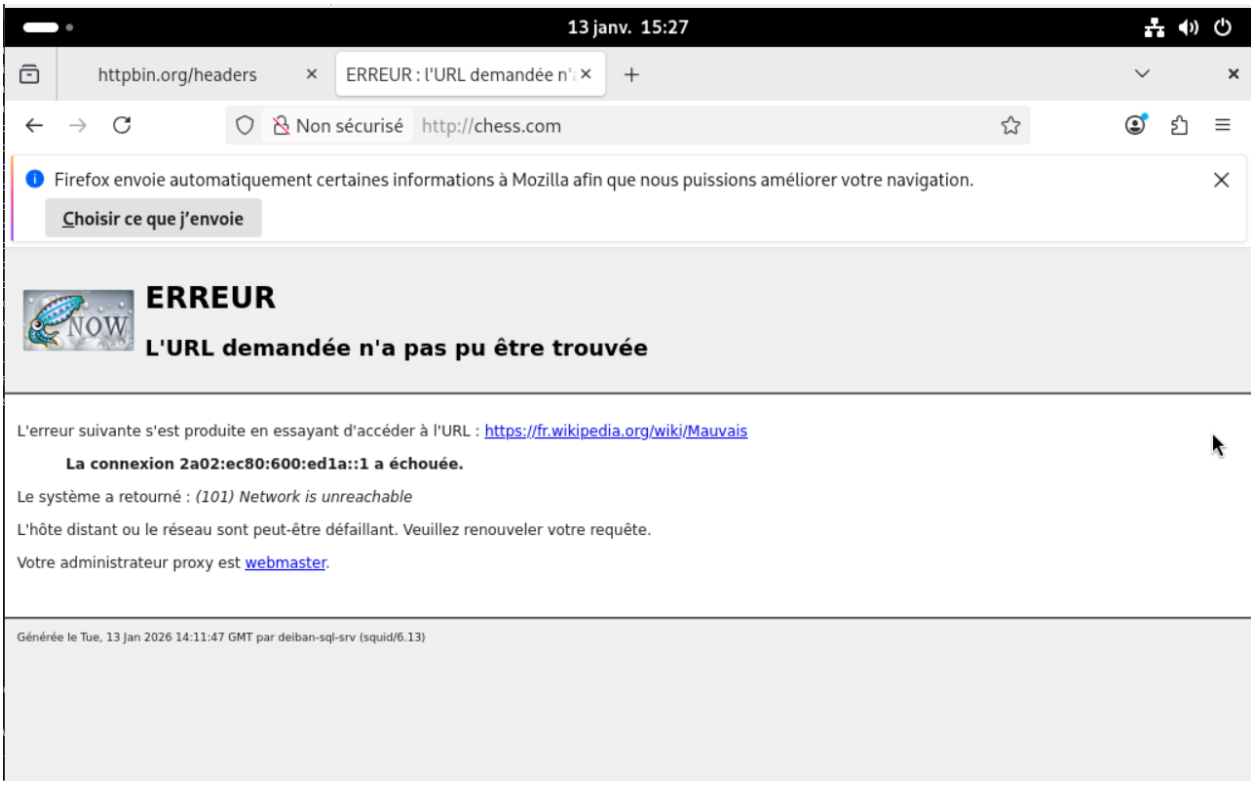
Test : Blocage de chess.com

Commande : Accès à http://chess.com depuis le navigateur client

Résultat : Site bloqué avec succès

Le navigateur affiche : "ERREUR - L'URL demandée n'a pas pu être trouvée"

Tentative de redirection vers https://fr.wikipedia.org/wiki/Mauvais détectée


13 janv. 15:27
httpbin.org/headers
ERREUR : l'URL demandée n'
Non sécurisé
http://chess.com
Firefox envoie automatiquement certaines informations à Mozilla afin que nous puissions améliorer votre navigation.
Choisir ce que j'envoie
ERREUR
L'URL demandée n'a pas pu être trouvée
L'erreur suivante s'est produite en essayant d'accéder à l'URL : https://fr.wikipedia.org/wiki/Mauvais
La connexion 2a02:ec80:600:ed1a::1 a échouée.
Le système a retourné : (101) Network is unreachable
L'hôte distant ou le réseau sont peut-être défaillant. Veuillez renouveler votre requête.
Votre administrateur proxy est webmaster.
Générée le Tue, 13 Jan 2026 14:11:47 GMT par deiban-sql-srv (squid/6.13)

9. Rédaction d'un fichier Proxy PAC

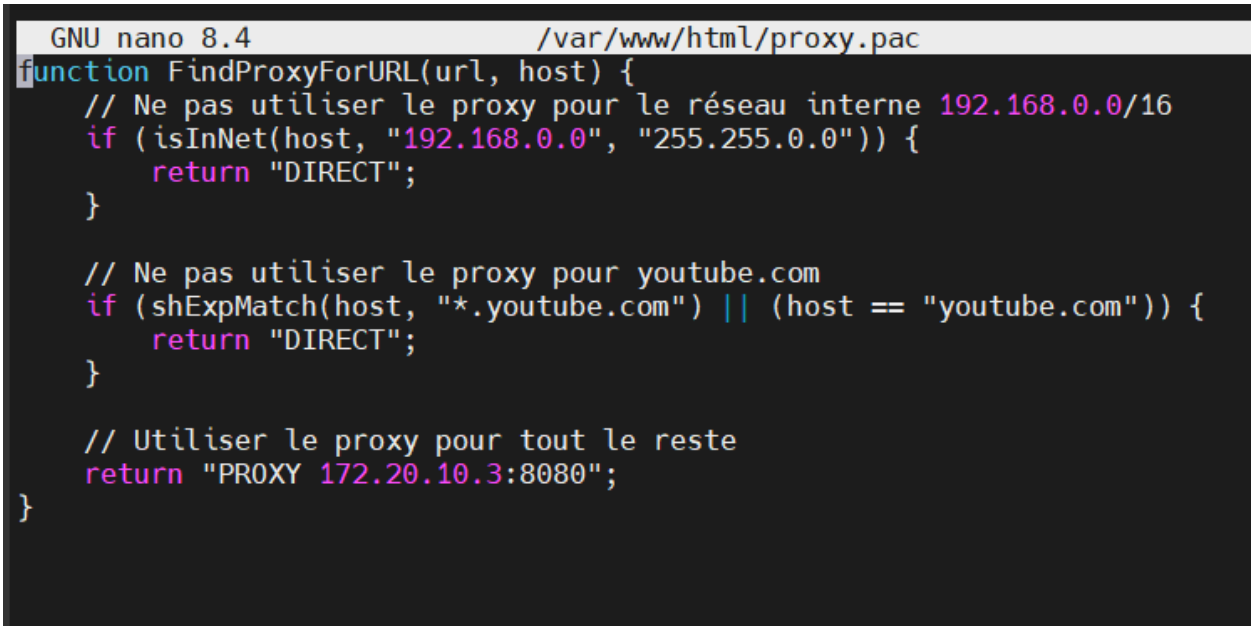
9.1. Objectif

Créer un fichier PAC (Proxy Auto-Config) permettant aux navigateurs de décider automatiquement quand utiliser le proxy selon des règles définies.

9.2. Création du fichier proxy.pac

Le fichier a été créé dans /var/www/html/proxy.pac

```
function FindProxyForURL(url, host) {
   // Ne pas utiliser le proxy pour le réseau interne
   if (isInNet(host, "192.168.0.0", "255.255.0.0")) {
      return "DIRECT";
   }

   // Ne pas utiliser le proxy pour youtube.com
   if (shExpMatch(host, "*.youtube.com") || (host == "youtube.com")) {
      return "DIRECT";
   }

   // Utiliser le proxy pour tout le reste
   return "PROXY 172.20.10.3:8080";
}
```



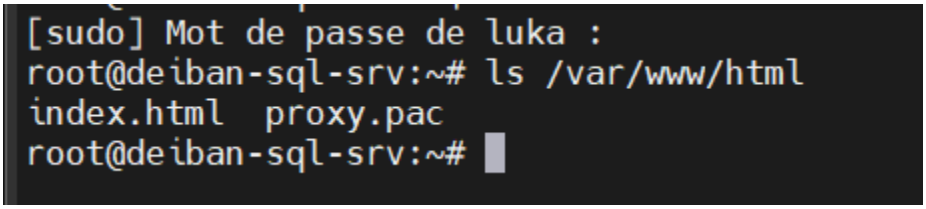
9.3. Règles configurées

- Réseau interne 192.168.0.0/16 : Connexion directe (DIRECT)
- youtube.com et ses sous-domaines : Connexion directe (DIRECT)
- Tous les autres sites : Utilisation du proxy (PROXY 172.20.10.3:8080)

9.4. Mise à disposition du fichier

Installation d'Apache pour servir le fichier PAC :

sudo apt install apache2 -y
sudo systemctl start apache2
sudo chmod 644 /var/www/html/proxy.pac



Le fichier est accessible via : http://172.20.10.3/proxy.pac

9.5. Configuration du navigateur

Dans Firefox :

5. Paramètres → Réseau → Paramètres de connexion
6. Sélectionner "URL de configuration automatique du proxy"
7. Entrer l'URL : http://192.168.1.36/proxy.pac (l'ip à changer car j'ai déplacé le labo et que je mets ma VM sur mon réseau physique pour pouvoir y accéder depuis mon ordinateur fixe)

9.6. Tests et validation

Test 1 : Accès à youtube.com

Résultat : Connexion directe (DIRECT)

Vérification dans les logs Squid : Aucune requête vers youtube.com n'apparaît, confirmant que la connexion s'effectue en mode direct sans passer par le proxy.

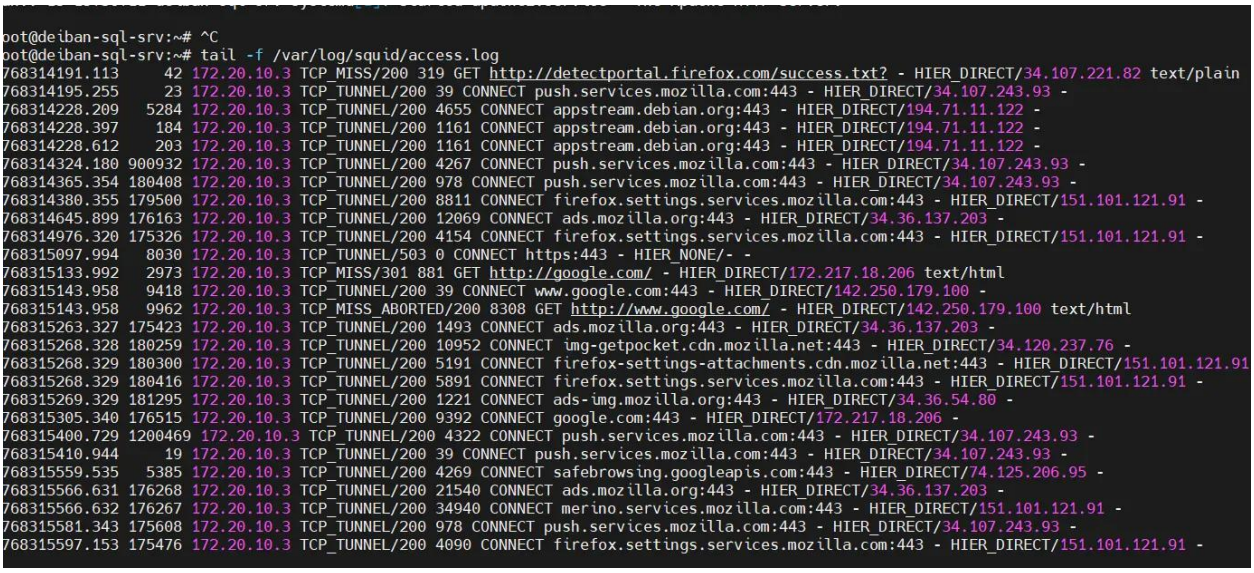
Test 2 : Accès à google.com

Résultat : Connexion via le proxy

Vérification dans les logs Squid :

172.20.10.3 TCP_MISS/301 881 GET http://google.com/

La requête vers google.com apparaît dans les logs, confirmant le passage par le proxy.



```
oot@deiban-sql-srv:~# ^C
oot@deiban-sql-srv:~# tail -f /var/log/squid/access.log
768314191.113     42 172.20.10.3 TCP_MISS/200 319 GET http://detectportal.firefox.com/success.txt? - HIER_DIRECT/34.107.221.82 text/plain
768314195.255     23 172.20.10.3 TCP_TUNNEL/200 39 CONNECT push.services.mozilla.com:443 - HIER_DIRECT/34.107.243.93 -
768314228.209   5284 172.20.10.3 TCP_TUNNEL/200 4655 CONNECT appstream.debian.org:443 - HIER_DIRECT/194.71.11.122 -
768314228.397    184 172.20.10.3 TCP_TUNNEL/200 1161 CONNECT appstream.debian.org:443 - HIER_DIRECT/194.71.11.122 -
768314228.612    203 172.20.10.3 TCP_TUNNEL/200 1161 CONNECT appstream.debian.org:443 - HIER_DIRECT/194.71.11.122 -
768314324.180 900932 172.20.10.3 TCP_TUNNEL/200 4267 CONNECT push.services.mozilla.com:443 - HIER_DIRECT/34.107.243.93 -
768314365.354 180408 172.20.10.3 TCP_TUNNEL/200 978 CONNECT push.services.mozilla.com:443 - HIER_DIRECT/34.107.243.93 -
768314380.355 179500 172.20.10.3 TCP_TUNNEL/200 8811 CONNECT firefox.settings.services.mozilla.com:443 - HIER_DIRECT/151.101.121.91 -
768314645.899 176163 172.20.10.3 TCP_TUNNEL/200 12069 CONNECT ads.mozilla.org:443 - HIER_DIRECT/34.36.137.203 -
768314976.320 175326 172.20.10.3 TCP_TUNNEL/200 4154 CONNECT firefox.settings.services.mozilla.com:443 - HIER_DIRECT/151.101.121.91 -
768315097.994   8030 172.20.10.3 TCP_TUNNEL/503 0 CONNECT https:443 - HIER_NONE/- -
768315133.992   2973 172.20.10.3 TCP_MISS/301 881 GET http://google.com/ - HIER_DIRECT/172.217.18.206 text/html
768315143.958   9418 172.20.10.3 TCP_TUNNEL/200 39 CONNECT www.google.com:443 - HIER_DIRECT/142.250.179.100 -
768315143.958   9962 172.20.10.3 TCP_MISS_ABORTED/200 8308 GET http://www.google.com/ - HIER_DIRECT/142.250.179.100 text/html
768315263.327 175423 172.20.10.3 TCP_TUNNEL/200 1493 CONNECT ads.mozilla.org:443 - HIER_DIRECT/34.36.137.203 -
768315268.328 180259 172.20.10.3 TCP_TUNNEL/200 10952 CONNECT img-getpocket.cdn.mozilla.net:443 - HIER_DIRECT/34.120.237.76 -
768315268.329 180300 172.20.10.3 TCP_TUNNEL/200 5191 CONNECT firefox-settings-attachments.cdn.mozilla.net:443 - HIER_DIRECT/151.101.121.91
768315268.329 180416 172.20.10.3 TCP_TUNNEL/200 5891 CONNECT firefox.settings.services.mozilla.com:443 - HIER_DIRECT/151.101.121.91 -
768315269.329 181295 172.20.10.3 TCP_TUNNEL/200 1221 CONNECT ads-img.mozilla.org:443 - HIER_DIRECT/34.36.54.80 -
768315305.340 176515 172.20.10.3 TCP_TUNNEL/200 9392 CONNECT google.com:443 - HIER_DIRECT/172.217.18.206 -
768315400.729 1200469 172.20.10.3 TCP_TUNNEL/200 4322 CONNECT push.services.mozilla.com:443 - HIER_DIRECT/34.107.243.93 -
768315410.944     19 172.20.10.3 TCP_TUNNEL/200 39 CONNECT push.services.mozilla.com:443 - HIER_DIRECT/34.107.243.93 -
768315559.535   5385 172.20.10.3 TCP_TUNNEL/200 4269 CONNECT safebrowsing.googleapis.com:443 - HIER_DIRECT/74.125.206.95 -
768315566.631 176268 172.20.10.3 TCP_TUNNEL/200 21540 CONNECT ads.mozilla.org:443 - HIER_DIRECT/34.36.137.203 -
768315566.632 176267 172.20.10.3 TCP_TUNNEL/200 34940 CONNECT merino.services.mozilla.com:443 - HIER_DIRECT/151.101.121.91 -
768315581.343 175608 172.20.10.3 TCP_TUNNEL/200 978 CONNECT push.services.mozilla.com:443 - HIER_DIRECT/34.107.243.93 -
768315597.153 175476 172.20.10.3 TCP_TUNNEL/200 4090 CONNECT firefox.settings.services.mozilla.com:443 - HIER_DIRECT/151.101.121.91 -
```

Dans la capture ci-dessus, on voit la requête vers google.com, j’ai fait la requête vers youtube en même temps (avec un second onglet) mais on ne l’a voit pas car elle ne passe pas par le proxy.