Procédure proxy Squid

I) Préparation du système et du réseau

Configuration du réseau :

```
sudo nano /etc/network/interfaces
```

Modifier ce fichier comme suit :

```
auto ens38
iface ens38 inet dhcp
auto ens33
iface ens33 inet static
    address 10.11.29.21
    netmask 255.255.255.240
    gateway 10.11.29.17
    dns-nameserver 10.11.29.18
```

Puis appliquer les changements avec :

```
sudo systemctl restart networking
```

Configuration des sources :

```
sudo nano /etc/apt/sources.list
```

Remplacer le contenu de ce fichier par les nouvelles sources ci-dessous :

```
deb http://deb.debian.org/debian bookworm main contrib non-free non-free-firmware
deb http://deb.debian.org/debian bookworm-updates main contrib non-free non-free-firmware
deb http://security.debian.org/debian-security bookworm-security main contrib non-free non-free-firmware
```

Modifier le fichier du noyau Linux pour permettre à notre serveur de transférer des paquets entre son interface LAN et WAN :

```
sudo nano /etc/sysctl.conf
```

Puis décommentez la ligne suivante :

```
net.ipv4.ip_forward=1
```

Appliquez les modifications sans redémarrer :

```
sudo sysctl –p
```

II) Configuration du pare-feu nftables pour la redirection transparente

Modifier le fichier de configuration de nftables :

```
sudo nano /etc/nftables.conf
```

Remplacer son contenu par les règles suivantes :

```
#!/usr/sbin/nft -f

flush ruleset

table inet filter {
    chain input {
        type filter hook input priority 0; policy drop;
        # Accepter le trafic de la boucle locale iifname "lo" accept
        # Accepter les requêtes DHCP et DNS venant du LAN
        iifname "ens33" udp dport {67, 53} accept
        iifname "ens33" tcp dport 53 accept
        # Accepter le trafic déjà établi ct state { established, related } accept
    }
    chain forward {
        type filter hook forward priority 0; policy accept;
        # Autoriser le trafic du LAN vers le WAN à condition qu'il soit "natté"
        iif "ens38" oif "ens33" accept
        # Accepter le trafic déjà établi ct state { established, related } accept
    }
```

```
    chain output {
        type filter hook output priority 0; policy accept;
    }
}
table nat {
    chain prerouting {
        type nat hook prerouting priority -100;
        # Rediriger le trafic HTTP (port 80) vers le port de Squid 3129
        iifname "ens38" tcp dport 80 redirect to :3129
        # Rediriger le trafic HTTPS (port 443) vers le port de Squid 3130
        iifname "ens38" tcp dport 443 redirect to :3130
    }
    chain postrouting {
        type nat hook postrouting priority 100;
        # Masquer le trafic sortant du LAN derrière l'IP du WAN
        oifname "ens33" masquerade
    }
}
```

Appliquez les nouvelles règles de pare-feu :

```
sudo systemctl enable nftables.service
sudo systemctl restart nftables.service
```

III) Mise en place du filtrage automatisé

Créez le répertoire qui contiendra nos fichiers de listes noires et donnez les bonnes permissions à l'utilisateur proxy (l'utilisateur sous lequel tourne Squid).

```
sudo mkdir /etc/squid/blacklist
sudo chown -R proxy:proxy /etc/squid/blacklist
```

Créez le fichier de script :

```
sudo nano /opt/refresh-blacklist.sh
```

Collez le contenu suivant dans le script

```
#!/bin/bash

# Création d'un répertoire temporaire

TMP_DIR=$(mktemp -d)

echo "Téléchargement de la blacklist depuis GitHub..."

git clone https://github.com/grosskurth/ut-capitole-blacklists.git "$TMP_DIR"

echo "Nettoyage des anciennes listes..."

sudo rm -rf /etc/squid/blacklist/*

echo "Déplacement des nouvelles listes utiles..."

sudo mkdir -p /etc/squid/blacklist

sudo mv "$TMP_DIR/ut-capitole-blacklists/blacklists/adult" /etc/squid/blacklist/

sudo mv "$TMP_DIR/ut-capitole-blacklists/blacklists/ads" /etc/squid/blacklist/

echo "Application des permissions..."

sudo chown -R proxy:proxy /etc/squid/blacklist

# Nettoyage

rm -rf "$TMP_DIR"

echo "Rechargement de la configuration de Squid..."

sudo systemctl reload squid.service
```

Rendez le script exécutable :

```
sudo chmod +x /opt/refresh-blacklist.sh
```

Nous allons planifier l'exécution de ce script tous les jours à 2h du matin avec Cron

Exécutez cette commande pour ouvrir l'éditeur de tâches planifiées :

```
Sudo crontab -e
```

Puis ajoutez cette ligne à la fin du fichier :

```
0 2 * * * /opt/refresh-blacklist.sh
```

Signification : à la minute 0, de la 2ème heure, chaque jour, chaque mois, chaque jour de la semaine, exécute ce script.

Maintenant, exécutez le script manuellement pour peupler vos listes :

```
sudo /opt/refresh-blacklist.sh
```

IV) Configuration de Squid

Pour pouvoir filtrer le trafic HTTPS, Squid doit se comporter comme un « homme du milieu » (Man-in-the-Middle) : il déchiffre le trafic, l'analyse, puis le rechiffre avant de l'envoyer au client. Pour cela, il doit générer des certificats à la volée, en se faisant passer pour le site demandé. Cela n'est possible que si les clients font confiance à l'autorité de certification (CA) de Squid.

Créez la CA de Squid :

```
sudo mkdir -p /etc/squid/cert/
```

```
cd /etc/squid/cert/
```

```
sudo openssl req -new -newkey rsa:4096 -sha256 -days 3650 -nodes -x509 -keyout squid_proxyCA.pem -out squid_proxyCA.pem
```

Remplissez les informations demandées. Pour le « Common Name », mettez quelque chose d'explicite comme « CA Proxy Squid Entreprise ».

```
sudo chown -R proxy:proxy /etc/squid/cert/ sudo chmod 0400 /etc/squid/cert/squid_proxyCA.pem
```

Initialisez la base de données des certificats SSL générés.

Note : Vérifiez le chemin avec `which security_file_certgen` s'il y a une erreur :

```
sudo /usr/lib/squid/security_file_certgen -c -s /var/spool/squid/ssl_db -M 4MB
```

```
sudo chown -R proxy:proxy /var/spool/squid/ssl_db
```

Rédaction des permissions du fichier squid.conf :

Faites une sauvegarde de la configuration par défaut et créez un fichier vide.

```
sudo mv /etc/squid/squid.conf /etc/squid/squid.conf.bak

sudo touch /etc/squid/squid.conf

sudo nano /etc/squid/squid.conf
```

Puis coller dans le fichier la configuration suivante :

```
# -- PORTS D'ECOUTE --

# Port HTTP transparent

http_port 3129 transparent

# Port HTTPS transparent avec interception SSL

https_port 3130 intercept ssl-bump generate-host-
certificates=on dynamic_cert_mem_cache_size=4MB
cert=/etc/squid/cert/squid_proxyCA.pem
key=/etc/squid/cert/squid_proxyCA.pem

# -- CONFIGURATION DE L'INTERCEPTION SSL (SSL-BUMP) --

# Programme helper pour générer les certificats
sslcrtd_program /usr/lib/squid/security_file_certgen -s
/var/spool/squid/ssl_db -M 4MB

# Configuration du "bumping"

ssl_bump peek all

ssl_bump bump all

# -- LISTES DE CONTROLE D'ACCES (ACL) --

# ACL pour les ports sécurisés

acl Safe_ports port 80 # http

acl Safe_ports port 443 # https

acl CONNECT method CONNECT

# ACL pour nos listes noires

acl adult dstdomain

"/etc/squid/blacklist/adult/domains"

acl adult url_regex "/etc/squid/blacklist/adult/urls"
```

```
acl ads dstdomain "/etc/squid/blacklist/ads"

# -- REGLES D'ACCES -- #

Les règles sont lues dans l'ordre. La première qui correspond est appliquée.

http_access deny !Safe_ports

http_access deny CONNECT !Safe_ports

# Autoriser l'administration locale (non utilisé ici, mais bonne pratique)

http_access allow localhost manager

http_access deny manager

# Blocage basé sur nos listes noires

http_access deny adult

http_access deny ads

# Autoriser tout le reste

http_access allow localhost

http_access allow all

# -- AUTRES PARAMETRES --

coredump_dir /var/spool/squid

refresh_pattern . 0 20% 4320
```

Ensuite nous pourrons déployer le certificat de la CA sur le serveur Squid

Convertissez le certificat .pem au format .crt et placez le dans le bon répertoire :

```
sudo openssl x509 -inform PEM -in

/etc/squid/cert/squid_proxyCA.pem -out

/usr/local/share/ca-certificates/squid_proxyCA.crt
```

Mettre à jour le magasin de certificats :

```
sudo update-ca-certificates
```

Puis redémarrez Squid pour appliquer les modifications

```
sudo systemctl restart squid
```

V) Configuration des postes clients

Récupérez le fichier du certificat qui se trouve dans le serveur, il se trouve à cet emplacement : `/usr/local/share/ca-certificates/` et est nommé `squid_proxyCA.crt`

Sur un poste Debian/Ubuntu :

Copiez le fichier `squid_proxyCA.crt` dans le répertoire `/usr/local/share/ca-certificates/` du client.

Sur le client, exécutez la commande :

```
sudo update-ca-certificates
```

Sur un poste client Windows :

Récupérez le fichier `squid_proxyCA.crt`.

Double-cliquez dessus. Cliquez sur « Installer le certificat ».

Choisissez « Machine locale ».

Important : Sélectionnez « Placer tous les certificats dans le magasin suivant » et choisissez le magasin « Autorités de certification racines de confiance ».

Validez l’assistant.