

Mise en place d'un Proxy Socks

- [Introduction](#)
- [Installation et configuration](#)
- [Configuration du/des Clients](#)
- [Annexe](#)

Introduction

I – Contexte

Motivations : Afin de renforcer ma confidentialité, ma sécurité et mon anonymat quand je navigue sur le web, j'ai décidé de mettre mes compétences à profit pour me construire mon propre serveur proxy.

C'est quoi un serveur proxy : Un serveur proxy est un serveur informatique qui agit comme un intermédiaire entre un utilisateur et un serveur de destination. Il sert d'interface entre les demandes provenant des utilisateurs et les ressources disponibles sur d'autres serveurs. Lorsqu'un utilisateur effectue une demande (par exemple, accéder à un site web), le serveur proxy intercepte la requête et la transmet au serveur de destination au nom de l'utilisateur. Il peut être utilisé pour des fonctions telles que la sécurité, la confidentialité, le filtrage du contenu, l'accélération du trafic et la gestion du réseau.

Différences avec un VPN : Un VPN chiffre la connexion alors que le serveur proxy sert d'intermédiaire pour naviguer. Selon le site de FORTINET, Bien que les proxys et les VPN offrent tous deux une confidentialité, ils le font de manière différente. Lorsqu'on compare les capacités des proxys et des VPN, la différence réside dans le fait que les proxy agissent strictement comme une passerelle entre Internet et les utilisateurs. En revanche, le trafic d'un VPN passe par un tunnel chiffré et l'appareil de l'utilisateur, ce qui fait des VPN une solution efficace pour garantir la sécurité du réseau.

Source : <https://www.fortinet.com/resources/cyberglossary/proxy-vs-vpn#:~:text=When%20comparing%20proxy%20vs.,solution%20for%20ensuring%20network%20security>.

II – Prérequis

Le serveur que nous allons mettre en place ne demande pas beaucoup de ressources. Dans mon cas voici ma configuration :

- 1vCPU

- 1Go de RAM
- Debian 12 (stable)

Pour les autres prérequis, il vous faut si possible un VPS chez l'hébergeur de votre choix, j'ai choisi les 3 mois gratuit sur AWS, et mon VPS est situé en virginie.

Installation et configuration

III - Installation des prérequis

Snapd

Snap est un gestionnaire de paquet, il va nous permettre d'installer simplement le serveur.

```
sudo apt-get install snapd
```

Shadowsocks

Shadowsocks est le serveur proxy que l'on va utiliser pour mettre en place notre propre serveur. Pour l'installer c'est assez simple coller la ligne ci-dessous dans votre terminal

```
sudo snap install shadowsocks-libev
```

Ufw (*facultatif*)

Ufw est un firewall local qui peut être installer sur linux, je l'installe dans mon cas pour renforcer la sécurité de ma machine. Je ne vais pas détailler la sécurité de la machine dans cette procédure. Pour installer ufw, coller la ligne ci-dessous

```
sudo apt-get install ufw
```

IV - Modification de la configuration de shadowsocks

Nous allons dans un premier temps nous rendre dans le répertoire shadowsocks-libev qui a dû être directement crée lors de l'installation. Utiliser la commande ci-dessous pour vous y rendre.

```
cd /etc/shadowsocks-libev/
```

Une fois cela fait, il va falloir modifier le fichier de configuration « config.json ». Je vais utiliser nano pour faire la modification mais libre à vous d'utiliser l'éditeur qui vous convient. Mais d'abord on commence par une sauvegarde du fichier original.

```
mv config.json config.json.bak
```

Puis on passe à l'édition

```
sudo nano config.json
```

Copier le code ci-dessous et inséré le dans votre fichier

```
{  
  "server": "",           #Adresse ip de votre carte réseaux (Pas loopback)  
  "server_port": 8000,    #Port externe  
  "local_port": 1080,     #Port interne  
  "password": "",        #Mot de passe pour la connexion  
  "timeout": 60,         #Ne pas toucher ou mettre <500  
  "method": "aes-256-gcm", #le changer a votre envie  
  "nameserver": "1.1.1.1" #DNS (Ligne facultative)  
}
```

On va maintenant activer shadowsocks au démarrage et le redémarré.

```
sudo systemctl enable shadowsocks-libev
```

Puis on redémarre le service

```
sudo systemctl restart shadowsocks-libev
```

Ensuite nous allons faire l'ouverture de port avec ufw

```
sudo ufw allow 1080
```

```
sudo ufw allow 8000
```

V – Ouverture des ports (NAT/PAT)

Il va maintenant falloir ouvrir le port 8000 sur votre box, je vous laisse vous renseigner car cette opération varie en fonction du F.A.I de votre BOX. Il faut ouvrir le port 8000 en TCP et en UDP. Car par défaut shadowsocks est configuré pour marcher avec les deux mais libre à vous de le

restreindre en modifiant le fichier de configuration. Je vais détailler l'ouverture de port sur Amazon AWS ci-dessous, car je n'ai pas trouvé la procédure simple. Et l'interface n'est pas très « user-friendly ».

Configuration du/des Clients

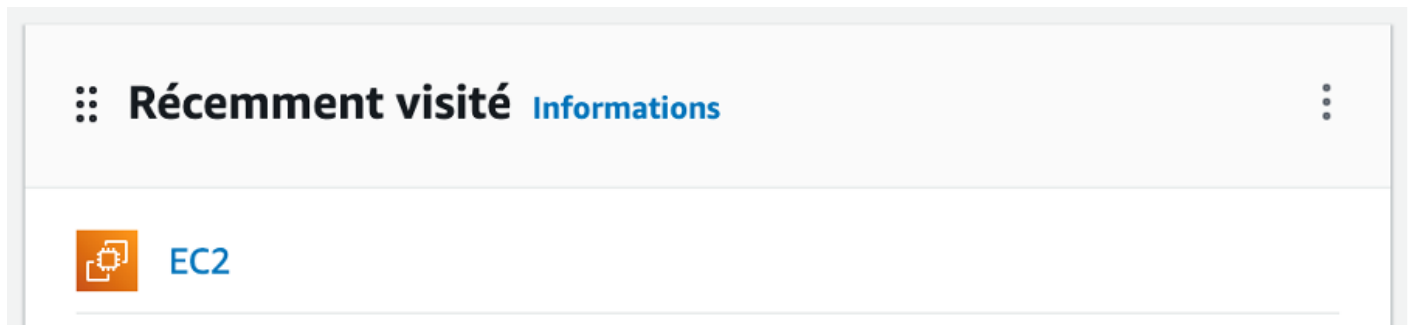
VI – Configuration Clients

Windows

1. Rendez-vous dans les release Github de shadowsocks-windows a ce lien :
<https://github.com/shadowsocks/shadowsocks-windows/releases>
2. Descendez un peu et trouver sous « Assets » Shadowsocks-X.X.X.X.zip (les X correspondent a la version.
3. Télécharger le fichier.
4. Décompressez l'archive et double clic sur shadowsocks.exe
5. Dans la petite flèche dans la barre des tâches faites un clic droit sur l'icône d'avion, puis serveur > éditer serveur. Dans cette fenêtre configurer votre serveur avec les mêmes informations que dans le fichier de configuration de shadowsocks. Si des champs sont vide et vous ne savez pas quoi mettre c'est qu'il ne faut pas y toucher.
6. Pour activer le proxy, clic droit sur l'avion puis proxy et enfin sur Global.
7. Pour vérifier que le proxy fonctionne bien, ouvrez un navigateur et rendez-vous sur ce site et regarder votre localisation. Elle doit normalement correspondre à la région sélectionner dans AWS et avoir l'IP publique de votre machine.

Annexe

1. Rendez-vous sur le lien suivant : <https://console.aws.amazon.com>
2. Puis cliquez sur EC2



3. Et enfin sur instances dans le menu latéral

▼ Instances

Instances

4. Puis sur l'id de votre instance

☐	Name 	▼	ID d'instance
☐	Proxy		i-0710b7e2e8d15775b

5. Sur Sécurité

Détails	Status and alarms New	Surveillance	Sécurité	Mise en réseau	Stockage	Balises
---------	--------------------------------------	--------------	----------	----------------	----------	---------

6. Sous règles entrante, trouver le bouton « launch-wizard-1 » et cliquez dessus.

▼ Règles entrantes

Filtrer les règles < 1 >					
	ID de règle du groupe de s...	Plage de ports	Protocole	Source	Groupes de sécurité
	sgr-08a933347087a75cf	1080	TCP	0.0.0.0/0	launch-wizard-1
	sgr-0a7572f0e52449388	1080	UDP	0.0.0.0/0	launch-wizard-1
	sgr-068870e0324376d86	8000	TCP	0.0.0.0/0	launch-wizard-1
	sgr-01310a4e50d4cbcd	1194	UDP	0.0.0.0/0	launch-wizard-1
	sgr-032726f171ace3cda	22	TCP	0.0.0.0/0	launch-wizard-1
	sgr-080082a460b1140b5	8000	UDP	0.0.0.0/0	launch-wizard-1

7. Cliquez ensuite sur l'id du groupe de sécurité

☐	Name	Security group ID
☐	—	sg-0ff4e61732362306f

8. Puis modifié les règles entrantes

9. Dans mon cas j'ai préféré ouvrir et le port 8000 et le port 1080 même si il correspond au port interne.

Règles entrantes (6) ↻ Gérer les balises Modifier les règles entrantes							
Search < 1 > ⚙							
☐	Name	ID de règle de grou...	Version...	Type	Protocole	Plage d...	Source
☐	—	sgr-08a933347087a75cf	IPv4	TCP personnalisé	TCP	1080	0.0.0.0/0
☐	—	sgr-0a7572f0e52449388	IPv4	UDP personnalisé	UDP	1080	0.0.0.0/0
☐	—	sgr-068870e0324376...	IPv4	TCP personnalisé	TCP	8000	0.0.0.0/0
☐	—	sgr-01310a4e50d4cbcd	IPv4	UDP personnalisé	UDP	1194	0.0.0.0/0
☐	—	sgr-032726f171ace3cda	IPv4	SSH	TCP	22	0.0.0.0/0
☐	—	sgr-080082a460b114...	IPv4	UDP personnalisé	UDP	8000	0.0.0.0/0