

OpenVPN

OpenVPN est un protocole et un logiciel permettant d'établir une connexion distante "VPN". Il s'agit d'un projet mûr et bénéficiant d'une solide réputation, il est l'oeuvre de James Yonan qui l'a créé le 13 mai 2001. Grâce à lui, vous allez pouvoir vous connecter à distance à votre réseau d'entreprise ou votre réseau local de votre domicile, le tout de façon sécurisée grâce au chiffrement.

- [Présentation](#)
- [Installation](#)

Présentation



Qu'est-ce qu'OpenVPN ?

OpenVPN est un logiciel open source qui permet de créer un réseau privé virtuel, appelé VPN. Ce réseau sécurisé permet de connecter des ordinateurs distants comme s'ils étaient sur le même réseau local, même s'ils sont éloignés géographiquement.

OpenVPN chiffre les données échangées, ce qui protège votre connexion contre les écoutes et les interceptions. C'est une solution très populaire pour sécuriser les accès à distance, notamment dans les entreprises ou pour les particuliers soucieux de leur vie privée.

À quoi sert OpenVPN ?

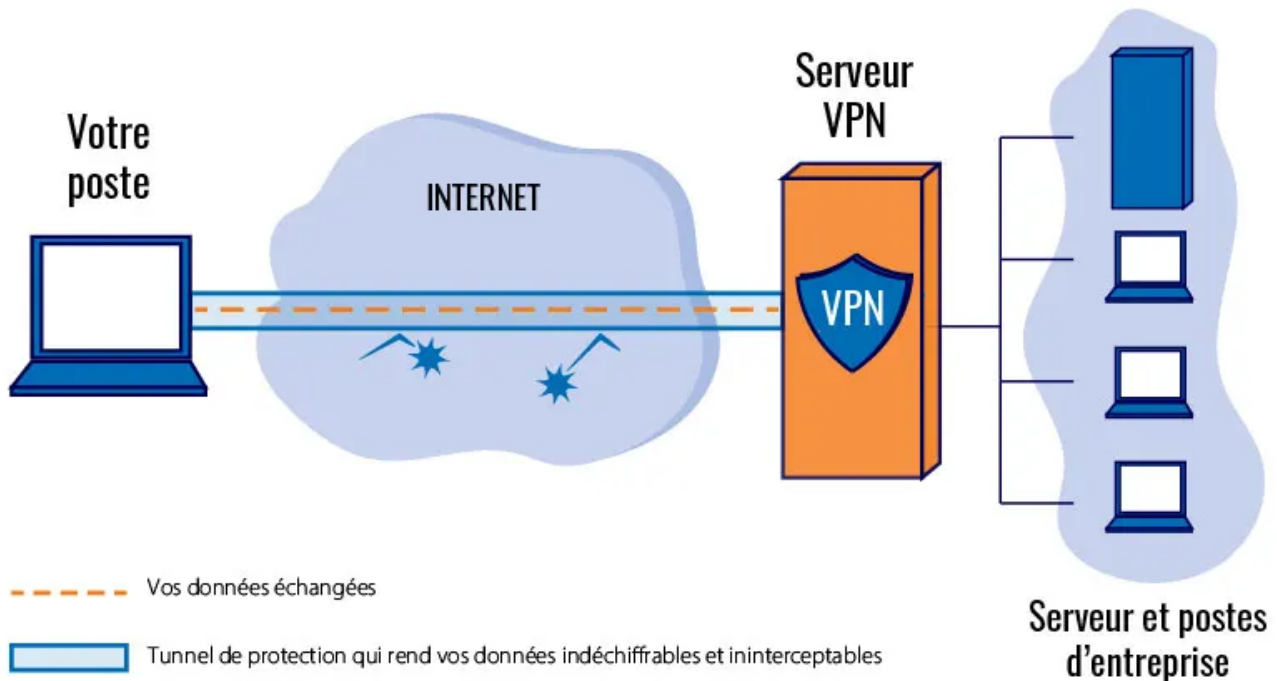
OpenVPN sert à établir une connexion sécurisée entre plusieurs sites ou utilisateurs. Par exemple, un salarié peut accéder aux ressources de son entreprise depuis chez lui, comme s'il était au bureau.

Il permet aussi de masquer son adresse IP, de contourner des restrictions géographiques, ou de protéger sa navigation sur des réseaux publics.

Comment fonctionne OpenVPN ?

OpenVPN fonctionne selon un principe client-serveur. Le serveur VPN est installé sur une machine accessible, souvent dans un datacenter ou une entreprise. Le client OpenVPN, installé sur votre ordinateur ou smartphone, se connecte au serveur.

Les échanges entre client et serveur sont chiffrés grâce à des certificats et des clés, garantissant la confidentialité des données. Une fois connecté, le client obtient une adresse IP virtuelle qui lui permet d'accéder au réseau distant.



Pourquoi utiliser OpenVPN ?

OpenVPN est reconnu pour sa robustesse, sa flexibilité, et sa sécurité. Il fonctionne sur de nombreuses plateformes (Linux, Windows, macOS, Android, iOS) et supporte différents modes de chiffrement.

Utiliser OpenVPN permet d'assurer une connexion sécurisée, même sur des réseaux publics non fiables, et de protéger les données sensibles en transit.

Cas d'usage courants

OpenVPN est souvent utilisé pour :

- Permettre un accès sécurisé au réseau d'entreprise à distance,
- Protéger la connexion internet sur les réseaux publics,

- Contourner des restrictions géographiques sur certains services en ligne,
- Connecter plusieurs sites distants entre eux,
- Assurer la confidentialité des échanges sur internet.

En résumé

OpenVPN est une solution efficace pour créer un réseau privé sécurisé entre plusieurs machines. Elle garantit la confidentialité et l'intégrité des données échangées, et permet un accès sécurisé à distance.

Installation



OPENVPN

Téléchargement

Sur le serveur VPN nouvellement créé et sur lequel nous sommes connectés :

```
apt update  
apt install curl
```

On peut à présent télécharger le script d'installation :

```
curl -O https://raw.githubusercontent.com/Angristan/openvpn-install/master/openvpn-install.sh
```

Il faut ensuite le rendre exécutable :

```
chmod +x openvpn-install.sh
```

Et enfin l'exécuter :

```
./openvpn-install.sh
```

Configuration

IPv4

Première étape, définir l'IPv4 local du serveur :

```
Welcome to the OpenVPN installer!  
The git repository is available at: https://github.com/angristan/openvpn-install
```

```
I need to ask you a few questions before starting the setup.  
You can leave the default options and just press enter if you are ok with them.
```

```
I need to know the IPv4 address of the network interface you want OpenVPN listening to.  
Unless your server is behind NAT, it should be your public IPv4 address.  
IP address: "IP"
```

Domaine

Vient ensuite l'IP publique, ou le nom de domaine par lequel le serveur est accessible :

```
It seems this server is behind NAT. What is its public IPv4 address or hostname?  
We need it for the clients to connect to the server.  
Public IPv4 address or hostname: "domaine.tld"
```

IPv6

Dans mon cas, je refuse l'utilisation de l'IPv6, simplement car je n'en ai pas l'utilité :

```
Checking for IPv6 connectivity...  
  
Your host appears to have IPv6 connectivity.  
  
Do you want to enable IPv6 support (NAT)? [y/n]: n
```

Port

Nous allons ensuite nous attaquer au port d'écoute du serveur :

```
What port do you want OpenVPN to listen to?  
  1) Default: 1194  
  2) Custom  
  3) Random [49152-65535]  
Port choice [1-3]: 3  
Random Port: "Random port"
```

J'ai préféré l'option du hasard, car peu de port sont utilisés dans cette étendue, et pour des raisons de sécurité, moins le port est évident, mieux c'est.

Protocole

Il va ensuite nous falloir choisir entre les protocoles UDP et TCP :

What protocol do you want OpenVPN to use?

UDP is faster. Unless it is not available, you shouldn't use TCP.

- 1) UDP
- 2) TCP

Protocol [1-2]: 1

Le protocole UDP étant plus fluide, il est recommandé pour l'utilisation d'un VPN.

DNS

Vient le choix du DNS utilisé avec le VPN :

What DNS resolvers do you want to use with the VPN?

- 1) Current system resolvers (from /etc/resolv.conf)
- 2) Self-hosted DNS Resolver (Unbound)
- 3) Cloudflare (Anycast: worldwide)
- 4) Quad9 (Anycast: worldwide)
- 5) Quad9 uncensored (Anycast: worldwide)
- 6) FDN (France)
- 7) DNS.WATCH (Germany)
- 8) OpenDNS (Anycast: worldwide)
- 9) Google (Anycast: worldwide)
- 10) Yandex Basic (Russia)
- 11) AdGuard DNS (Anycast: worldwide)
- 12) NextDNS (Anycast: worldwide)
- 13) Custom

DNS [1-12]: 11

Aucune préférence ici pour ma part.

Chiffrement

Je vais utiliser du chiffrement pour sécurisé au mieux mon VPN :

Do you want to customize encryption settings?

Unless you know what you're doing, you should stick with the default parameters provided by the script.

Note that whatever you choose, all the choices presented in the script are safe. (Unlike OpenVPN's defaults)

See <https://github.com/angristan/openvpn-install#security-and-encryption> to learn more.

Customize encryption settings? [y/n]: y

Chiffrement des données

Choose which cipher you want to use for the data channel:

- 1) AES-128-GCM (recommended)
- 2) AES-192-GCM
- 3) AES-256-GCM
- 4) AES-128-CBC
- 5) AES-192-CBC
- 6) AES-256-CBC

Cipher [1-6]: 3

Le cypher définit les bases de la connexion sécurisée :

- Comment les clés seront échangées
- Quel algorithme de chiffrement sera utilisé
- Quelle fonction de hachage sera utilisée pour vérifier l'intégrité
- Quel algorithme de signature est attendu sur le certificat

Type de certificat

Choose what kind of certificate you want to use:

- 1) ECDSA (recommended)
- 2) RSA

Certificate key type [1-2]: 1

Ce sont deux types de chiffrement, alors que RSA utilise la factorisation de grands nombres premiers, ECDSA utilise des courbes elliptiques, ce sont deux méthodes mathématiques totalement différentes.

Aujourd'hui une clé ECDSA 256bits est aussi sécurisée qu'une clé RSA 2048bits pour faire une comparaison.

ECDSA est plus récent, plus rapide, plus sécurisé que RSA, qui est de plus en plus abandonné.

Courbe de certificat

Choose which curve you want to use for the certificate's key:

- 1) prime256v1 (recommended)
- 2) secp384r1
- 3) secp521r1

Curve [1-3]: 1

Voici les courbes mentionnées plus haut, le choix de ECDSA ayant été fait, il faut choisir quelle courbe elliptique utiliser. Elles sont proposées dans l'ordre croissant de niveau de sécurité, mais aussi décroissant de rapidité. Ce type de clés étant déjà très sécurisées pour l'usage du cas présent, la première option est largement suffisante.

Pour continuer le parallèle avec un certificat RSA, la courbe ferait office de la taille de la clé, uniquement d'un point de vue fonctionnel, d'un point de vue mathématique, cela est beaucoup plus complexe.

Cryptage du control channel

Choose which cipher you want to use for the control channel:

- 1) ECDHE-ECDSA-AES-128-GCM-SHA256 (recommended)
- 2) ECDHE-ECDSA-AES-256-GCM-SHA384

Control channel cipher [1-2]: 1

Le control channel est comme son nom l'indique un canal, de communication dédié à l'échange des informations de gestions et de contrôle entre les deux extrémités du tunnel VPN.

Il a plusieurs rôles primordiaux :

- Echange des clés de chiffrements (mentionnées juste au dessus)
- Authentification des pairs
- Négociation des paramètres de connexion
- Signalisation et gestion du tunnel VPN

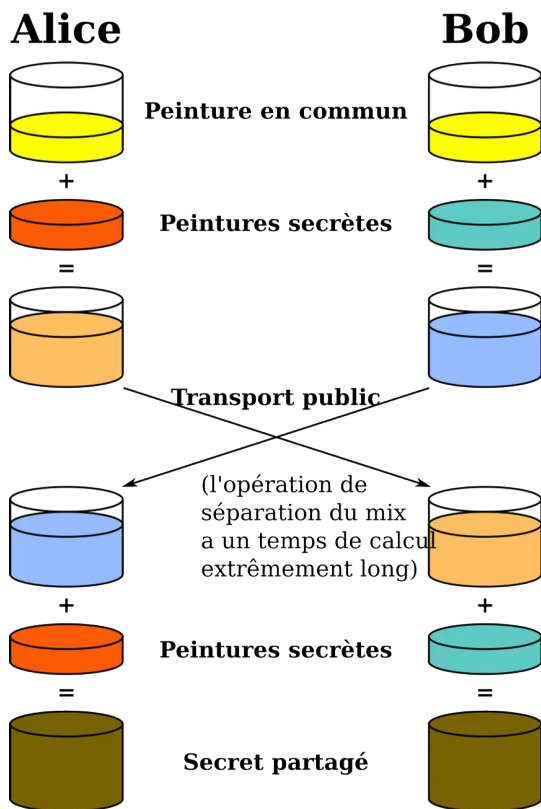
Type de clé Diffie-Hellman

Choose what kind of Diffie-Hellman key you want to use:

- 1) ECDH (recommended)
- 2) DH

DH key type [1-2]: 1

Diffie-Hellman est un algorithme qui permet de sécuriser un échange, sans jamais échanger de clé privée :



Ensuite, la différence entre les deux options est la même que pour RSA ou ECDSA, RSA=DH et ECDSA=ECDH, c'est à dire que ECDH est plus sécurisé et rapide.

Courbe pour les clés ECDH

Choose which curve you want to use for the ECDH key:

- 1) prime256v1 (recommended)
- 2) secp384r1
- 3) secp521r1

Curve [1-3]: 1

Il faut alors, la aussi, choisir la courbe pour la génération des clés, et le raisonnement pour le choix est identique.

Algorithme de hachage pour HMAC

The digest algorithm authenticates tls-auth packets from the control channel.

Which digest algorithm do you want to use for HMAC?

- 1) SHA-256 (recommended)
- 2) SHA-384
- 3) SHA-512

Digest algorithm [1-3]: 1

Le hachage est ce qui garantie l'intégrité de la donnée, ainsi que sa source. Cela sert notamment pour des signatures numériques, ou de l'authentification.

Quand aux choix proposés, cela est toujours la même question, il faut définir la priorité entre rapidité et sécurité.

Dans le cas présent, SHA-256 est sur a 99%.

Sécurité supplémentaire

```
You can add an additional layer of security to the control channel with tls-auth and tls-crypt
tls-auth authenticates the packets, while tls-crypt authenticate and encrypt them.
```

```
1) tls-crypt (recommended)
```

```
2) tls-auth
```

```
Control channel additional security mechanism [1-2]: 1
```

TLS-Auth permet d'authentifier les paquets à l'intérieur du control channel, et les rejette en cas de non conformité.

TLS-Crypt lui va plus loin en chiffrant les paquets du control channel en plus de les authentifiés, ce qui accroît la sécurité.

Finalisation

```
Okay, that was all I needed. We are ready to setup your OpenVPN server now.
```

```
You will be able to generate a client at the end of the installation.
```

```
Press any key to continue...
```

Une fois qu'on appuie sur une touche quelconque cela lance l'installation à proprement parlé.

Client

Nom

Vous êtes invité à créer le premier compte client :

```
Tell me a name for the client.
```

```
The name must consist of alphanumeric character. It may also include an underscore or a dash.
```

```
Client name: "NOM-PC"
```

Mot de passe

Vous pouvez choisir d'utiliser ou non un mot de passe pour sécurisé l'accès.

Do you want to protect the configuration file with a password?

(e.g. encrypt the private key with a password)

1) Add a passwordless client

2) Use a password for the client

Select an option [1-2]: 2

⚠ You will be asked for the client password below ⚠

J'ai fait le choix d'utiliser un mot de passe, ce qui lance ceci :

* Using SSL: openssl OpenSSL 3.0.15 3 Sep 2024 (Library: OpenSSL 3.0.15 3 Sep 2024)

* Using Easy-RSA configuration: /etc/openvpn/easy-rsa/vars

* The preferred location for 'vars' is within the PKI folder.

To silence this message move your 'vars' file to your PKI

or declare your 'vars' file with option: --vars=<FILE>

Enter PEM pass phrase:

Verifying - Enter PEM pass phrase:

Le mot de passe est demandé deux fois par soucis de vérification et éviter toutes erreur, puis OpenVPN génère les fichiers nécessaire, notamment le fameux .ovpn qu'il faut importer dans son client OpenVPN pour se connecter.

Notice

Keypair and certificate request completed. Your files are:

req: /etc/openvpn/easy-rsa/pki/reqs/NOM-PC.req

key: /etc/openvpn/easy-rsa/pki/private/NOM-PC.key

Using configuration from /etc/openvpn/easy-rsa/pki/acb207c7/temp.904fd4dd

Check that the request matches the signature

Signature ok

The Subject's Distinguished Name is as follows

commonName :ASN.1 12:'NOM-PC'

Certificate is to be certified until Apr 11 07:06:08 2035 GMT (3650 days)

Write out database with 1 new entries

Database updated

Notice

Certificate created at:

```
* /etc/openvpn/easy-rsa/pki/issued/NOM-PC.crt
```

Notice

Inline file created:

```
* /etc/openvpn/easy-rsa/pki/inline/NOM-PC.inline
```

Client NOM-PC added.

The configuration file has been written to /root/NOM-PC.ovpn.

Download the .ovpn file and import it in your OpenVPN client.

L'installation du serveur est terminée !

Ajouter de nouveaux clients

Il suffit de relancer le script principal

```
./openvpn-install.sh
```

Et de faire le choix 1

Welcome to OpenVPN-install!

The git repository is available at: <https://github.com/angristan/openvpn-install>

It looks like OpenVPN is already installed.

What do you want to do?

- 1) Add a new user
- 2) Revoke existing user
- 3) Remove OpenVPN
- 4) Exit

Select an option [1-4]: 1

Et de suivre la même procédure plus haut.