TP OpenVPN

Le but de ce TP est de mettre en place OpenVPN entre un serveur VPN d'une entreprise (avec réseau interne), et des clients sur PC nomade.

Travail demandé Pendant la séance, prenez des notes sur votre travail technique (montage du réseau, fichiers de configuration, problèmes rencontrés et solutions trouvées, etc.) et faites des captures réseau dès qu'une connexion marche.

Pour le 17 novembre, déposez sur Universitice un unique compte-rendu de TP au format PDF explicitant votre travail, les différentes configurations (montage réseau + config OpenVPN) et dé crivant les captures réseau effectuées durant le TP.

Site officiel d'OpenVPN : http://openvpn.net/index.php/open-source

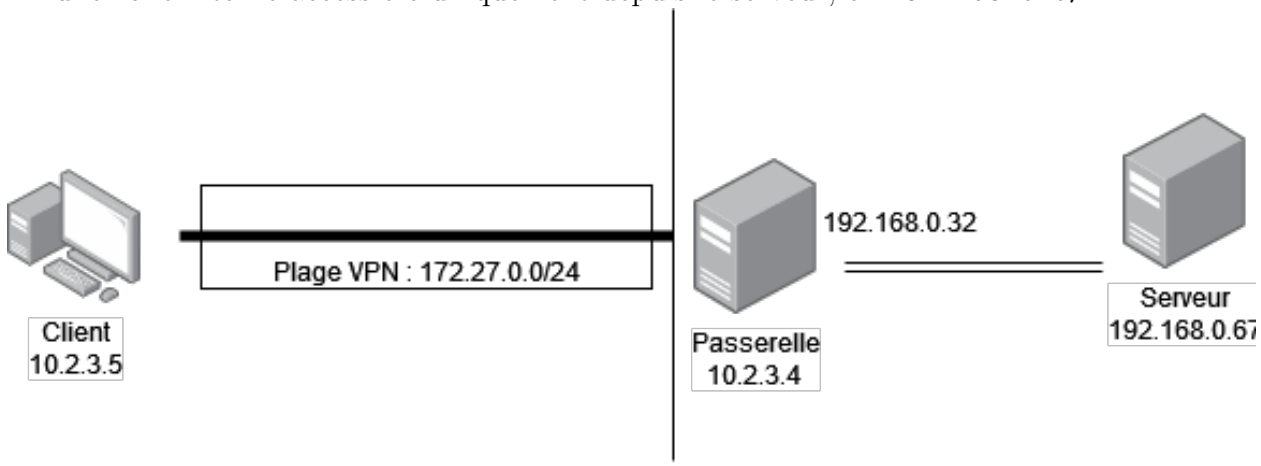
1 Mise en route

Le réseau que l'on cherche à monter consiste en :

— un serveur OpenVPN et un client OpenVPN qui sont sur une plage d'adresses publiques, adresse `10.2.3.4` pour le serveur et `10.2.3.5` pour le client ;

— une zone VPN, de plage d'adressage IP virtuelle `172.27.0.0/24` ;

— une zone interne accessible uniquement depuis le serveur, en `192.168.0.0/24`.



Montez une configuration client-serveur-machine interne.

Installez `openvpn`. Donnez votre version.

`Openvpn` peut se configurer de deux manières : par clefs partagées (mode manuel, statique), ou en utilisant TLS pour l'authentification et l'échange de clefs (mode automatique, dynamique).

2 Cryptographie

Exercice 1 A l'aide de votre autorité de certification personnelle [1], générez des certificats serveur et client(s) TLS pour votre VPN, ainsi que des paramètres (EC)DH pour l'échange de clefs. Attention, il faut que le certificat serveur puisse effectuer les usages `keyEncipherment`, `digitalSignature` (`keyUsage`) et `serverAuth` (`extendedKeyUsage`). Vous pouvez regarder le fichier de configuration `openssl.cnf` du fichier d'exemples `/usr/share/doc/openvpn/examples/sample-keys/openssl.cnf`.

Vous pouvez aussi vous inspirer du script compressé `/usr/share/doc/openvpn/examples/sample-keys/gen-sample-keys.sh.gz` qui propose une configuration intégrale des éléments cryptographiques [2]. Expliciter chaque étape avec les choix effectués.

3 Fichiers de configuration simple

Créer un fichier de configuration pour le serveur et pour le client. Vous pouvez partir du fichier d'exemple compressé ou non : `/usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz` et `client.conf`.

— Choix du protocole : `proto tcp`
— Configurez les clefs, certificats et paramètres (EC)DH générés précédemment.
— Définir le mode routeur : `dev tun` [3]
— Pour le début, augmenter la verbosité : `verb 5`

Pour le serveur :

— Configurer le mode serveur : `server 172.27.0.0 255.255.255.0`

Pour le client :

— Configurer l'adresse du serveur : `remote 10.2.3.4 1194`

4 Lancement du VPN

On lance le serveur : `# openvpn server.conf`

Là normalement, le fichier `openvpn-status.log` vous indique que le serveur est à l'écoute.

Puis on lance le client : `# openvpn client.conf` et on teste la connexion !

On peut ensuite ajouter le réseau interne, et configurer les routes (sur le serveur : la commande `push "route 192.168.0.0 255.255.255.0"` indiquera aux clients que le réseau `192.168.0.0/24` sera joignable au travers de la connexion VPN).

Vérifiez les fichiers de log. Sur le client, vérifiez les routes avec `netstat -r` ou `ip route`.

Vous pouvez créer un nouveau client (avec un nouveau certificat, normalement si vous utilisez le même certificat cela doit déconnecter le premier). Regardez la directive `client-to-client`.

1. Si vous n'en avez pas, créez-en une, elle resservira !
2. On trouve cela de temps en temps en plus du `man` !
3. Avec LXC, le conteneur n'a pas forcément les droits d'utiliser cette interface, cela peut nécessiter une configuration comme `lxc.mount.entry = /dev/net/tun dev/net/tun none bind,create=file`

5 Sécurisation de l'installation

Pour sécuriser un peu plus l'installation, appliquer les configurations suivantes :

Déni de Service Pour éviter des DoS, on peut définir sur chaque client une clef statique partagée. Pour générer la clef : `openvpn --genkey --secret ta.key`
Fichier de configuration : `tlsauth ta.key 0` sur le serveur et `1` sur les clients. On peut également ajouter un script de vérification `tls-verify script.sh` pour filtrer les certificats présentés.

Abaisser les privilèges du serveur Il faut créer un utilisateur `openvpn`[4] et chrooter le démon OpenVPN.

Cryptographie Vous pouvez tester d'autres types de configuration.
Pour voir les algorithmes cryptographiques disponibles : `openvpn --show-ciphers`, `openvpn --show-digests`. On peut fixer les algos dans le fichier de configuration avec `cipher` et `auth`.

Passerelle par défaut Vous pouvez indiquer depuis le serveur au client que le VPN doit être configuré comme passerelle par défaut : `push "redirect-gateway def1 bypass-dhcp"`. Que cela change-t-il ?

6 Guide ANSSI

De même que pour les TP SSH et IPsec, lisez le guide de l'ANSSI sur le nomadisme numérique (`https://www.ssi.gouv.fr/uploads/2018/10/guide_nomadisme_anssi_pa_054_v1.pdf` (il n'existe pas de guide spécifique à OpenVPN car ce n'est qu'une solution parmi d'autres - là où SSH et IPsec sont de l'ordre du standard). Proposez des configurations système à effectuer pour sécuriser un peu plus l'architecture VPN.

4. Pourquoi n'est-ce pas une bonne idée d'utiliser l'utilisateur et le groupe `nobody` par défaut ?