Cible de sécurité CSPN

OpenVPN 2.4.6

Réf.	18-02-403-LIV
Version	1.3
Date	03 août 2018
Preparé pour	ANSSI
Réalisé par	Quarkslab

Quarkslab SAS
13 rue Saint-Ambroise
75011 Paris
France

Table des matières

1. Introduction

1.1 Identification du document

Nom du projet d'évaluation	FIXME ID fournie par ANSSI
Référence de la cible de sécurité	18-02-403-LIV
Auteur	Philippe Teuwen <pteuwen@quarkslab.com>
Approbateur	Marion Videau
Date de création de la cible de sécurité	10 avril 2018
Date de mise à jour de la cible de sécurité	03 août 2018
N° de version de la cible de sécurité	1.3
Divers	

1.2 Identification du produit

Nom de l'éditeur	OpenVPN Technologies, Inc. <sales@openvpn.net>
Nom du produit	OpenVPN™
N° de version analysée	2.4.6
Correctifs éventuels appliqués	
Domaine technique CSPN	Réseau privé virtuel (VPN) logiciel

1.3 Aperçu du produit

La cible d'évaluation considérée est une passerelle VPN logicielle libre sous license GPL.

OpenVPN permet d'établir des connexions point à point ou site à site selon des configurations de type route (explicite) ou pont (transparent). Il utilise un protocole qui lui est propre et qui repose largement sur OpenSSL. L'authentification est possible soit grâce à une clé secrète partagée à l'avance, soit grâce à SSLv3/TLSv1 et des certificats électroniques. OpenVPN est disponible sous tous les environnements *nix tels que Solaris, OpenBSD, FreeBSD, NetBSD, Linux, mais aussi Mac OS X, Windows 2000, XP, Vista, 7, 8 et 10, Android et iOS.

Son développeur principal est James Yonan, co-fondateur d'OpenVPN Technologies, Inc. Selon www.openvpn.net, sa communauté comporte 5 millions d'utilisateurs.

2. Description de la cible d'évaluation

2.1 Type de produit

OpenVPN est une passerelle VPN logicielle destinée à être déployée dans divers environnements professionnels, voire privés, la plupart du temps pour connecter un poste client distant à un réseau privé ou à une passerelle Internet (et protéger le client d'une connexion Internet locale peu sûre).

L'environnement précis dans lequel OpenVPN va être déployé n'est pas connu de son développeur, puisque le produit peut être incorporé dans différents systèmes spécifiques et différents environnements système.

Notons que OpenVPN Technologies propose également à la vente des solutions clé en main, mais la cible d'évaluation considérée dans ce document est la version libre, "communautaire", du logiciel [1].

Cette version telle que distribuée sur le site officiel est clairement destinée à des administrateurs système aguerris, car sous forme de code source à compiler. En outre un installateur pour les environnements Windows est fourni.

Le cas d'usage typique sera donc celui d'un administrateur réseau qui installera un serveur OpenVPN sur une machine *nix (probablement même une version packagée pour sa distribution ainsi que les dépendances, plutôt que les sources officielles), en choisira la configuration et transmettra les quelques paramètres de configuration à ses utilisateurs qui installeront par eux-mêmes la version cliente sur leur ordinateur portable. Il est également possible que ce soit directement l'administrateur qui procède au déploiement sur les machines clientes, mais il est intéressant d'observer comment l'installation du produit se passe face à des utilisateurs peu avertis.

Dans la suite de ce document, l'acronyme "ToE" (Target of Evaluation) et la locution "cible d'évaluation" seront utilisés de manière interchangeable. De même "OpenVPN" et "le produit" seront des synonymes désignant le produit dans sa globalité, sans être limités par un périmètre d'évaluation.

2.2 Fonctions de sécurité de la cible d'évaluation

La ToE comprend les fonctions suivantes :

F1 : Fonction de VPN

La ToE permet d'établir des tunnels VPN avec une passerelle distante ou avec un client nomade. La passerelle, construite sur un protocole qui lui est propre, repose sur des protocoles standard (TLS) et implémente, entre autres, des mécanismes cryptographiques conformes aux recommandations de l'ANSSI en vigueur. La fonction de VPN apporte confidentialité et intégrité du trafic utilisateur au travers d'un réseau non maîtrisé (lire : Internet).

F2 : Mécanismes d'authentification

[1] https://openvpn.net/index.php/download/community-downloads.html

La ToE offre la possibilité de configurer différents moyens d'authentification entre ToE serveur et ToE client. Ces mécanismes sont en œuvre lors de l'établissement de la fonction de VPN.

F3 : Fonctions d'administration

La ToE dispose de fonctions permettant de configurer l'ensemble des autres fonctionnalités. La configuration se fait au moyen de fichiers et éventuellement de scripts. Il existe également une option d'administration distante via TCP ou socket unix. Cette option est complètement non sécurisée et doit être protégée par d'autres moyens si elle est utilisée.

F4 : Journalisation locale d'événements

La ToE permet de définir une politique de journalisation locale d'événements (au niveau du serveur et au niveau du client) notamment de sécurité et d'administration.

2.3 Périmètre de la cible d'évaluation et configuration évaluée

2.3.1 Éléments de la ToE

D'un point de vue fonctionnel, l'usage de la ToE est double.

La ToE permet de connecter un client nomade en VPN pour accéder au réseau protégé.

La ToE permet également d'instaurer un tunnel VPN entre le réseau à protéger et un réseau distant protégé par une passerelle similaire. Dans ce cas, l'une des deux passerelles jouera le rôle d'un client et l'autre du serveur.

Les deux parties assurent l'authenticité, l'intégrité et la confidentialité des communications.

Il n'y a pas de distinction entre client et serveur au niveau du logiciel fourni, seule la configuration (option `client`) déterminera le rôle de chaque partie.

L'évaluation portera sur les composants suivants :

- un serveur OpenVPN sous Linux (distribution au choix, architecture x64), installé depuis les sources disponibles sur le site officiel, en version 2.4.6 [2].
- un client OpenVPN sous Linux (distribution au choix, architecture x64), installé depuis les sources disponibles sur le site officiel, en version 2.4.6 (le même exécutable fournit les fonctions client et serveur).
- un client OpenVPN sous Windows 10 x64, installé depuis l'installateur disponible sur le site officiel, en version 2.4.6 [3]. Ce dernier inclut quelques briques supplémentaires :
  - TAP-windows NDIS 6, un logiciel pilote chargé de créer et gérer des équipements virtuels de type "TAP", en version 9.21.2. En raison de son rôle critique, TAP-windows fait partie du périmètre à évaluer, donc de la ToE.
  - OpenVPN-GUI, une interface graphique sans grand intérêt du point de vue de l'analyse de sécurité, elle sera donc exclue du périmètre.
  - EasyRSA 2, un logiciel qui permet de générer sa propre autorité de certification. EasyRSA est exclu du périmètre tout comme la gestion de clés de manière générale.

[2] https://swupdate.openvpn.org/community/releases/openvpn-2.4.6.tar.xz
[3] https://swupdate.openvpn.org/community/releases/openvpn-install-2.4.6-I602.exe

Les dépendances, même si incluses dans l'installateur Windows, sont considérés comme hors du périmètre d'évaluation, notamment la bibliothèque OpenSSL en version 1.1.0.

Dans le cadre de l'évaluation, la version d'OpenSSL considérée est la 1.1.0h.

Les différentes configurations considérées comme sûres et donc à évaluer sont dérivées de celles données en exemple [4] et celles préconisées sur le site officiel [5], en particulier :

Un résumé des options de configuration retenues est fourni en annexe.

2.3.2 Paramètres généraux

Le site officiel recommande les éléments de configuration suivants :

- UDP (`proto udp`) car il offre une meilleure protection contre le *port scanning* et le déni de service.
- L'abaissement automatique des privilèges administrateur de la ToE serveur dès son démarrage (`user nobody` et `group nobody`). Il existe une possibilité, sous Linux uniquement, de faire tourner OpenVPN complètement en mode non privilégié, mais la configuration est spécifique et compliquée à mettre en œuvre et on ne peut donc pas considérer raisonnablement qu'elle soit déployée par défaut (quoique certaines distributions Linux l'appliquent peut-être par défaut). Il en va de même pour l'option `chroot`, non triviale à mettre en œuvre.
- Concernant la journalisation, un court fichier donnant le statut courant est mis à jour chaque minute (`status openvpn-status.log`) et les messages de journalisation (niveau `verb 3` par défaut) sont envoyés par défaut au *syslog* ou, sous Windows en tant que service, dans le répertoire `\Program Files\OpenVPN\log`.
- `topology subnet`, pourtant le choix officiellement non recommandé `net30` est celui actif par défaut.

La protection anti-rejeu est active par défaut. La documentation officielle déconseille explicitement de la désactiver et l'option de désactivation est étiquetée comme étant obsolète avant de disparaître complètement d'OpenVPN v2.5. En environnement dynamique où de multiples sessions se succèdent, il est recommandé d'activer l'option `replay-persist file` pour garder une trace de l'état des protections anti-rejeu d'une session à l'autre dans un fichier.

2.3.3 Méthodes d'authentification

OpenVPN permet plusieurs méthodes d'authentification, cependant la seule configuration retenue pour la ToE sera le **mode TLS**, les deux autres méthodes sont écartées pour les raisons détaillées ci-dessous.

Clé secrète partagée (configuration non retenue)

Cette configuration n'est possible que pour établir une liaison point à point entre deux machines. Dans cette configuration, l'authentification mutuelle repose sur une clé secrète partagée. Cette clé statique contient quatre clés indépendantes : pour le HMAC d'émission, le HMAC de réception, le chiffrement et le déchiffrement. Par défaut, les clés de HMAC seront identiques des

[4] https://github.com/OpenVPN/openvpn/blob/master/sample/sample-config-files/server.conf et https://github.com/OpenVPN/openvpn/blob/master/sample/sample-config-files/client.conf
[5] https://openvpn.net/index.php/open-source/documentation/howto.html

deux côtés ainsi que les clés de chiffrement, mais il est possible de changer ce fait via l'option `secret` comme recommandé dans le mini-howto [6].

Dans ce mode, la suite cryptographique doit être spécifiée statiquement et celle présentée par défaut dans la configuration donnée en exemple est `cipher AES-256-CBC`. Seul le mode CBC est possible dans ce mode.

Soulignons l'absence de *perfect forward secrecy* de cette configuration et le stockage en clair de la clé statique dans la ToE. C'est pourquoi *cette configuration est considérée comme peu sûre et est écartée*, car incapable de satisfaire pleinement les objectifs de securité.

Mode TLS (configuration retenue)

Dans cette configuration, une session TLS est établie avec authentification mutuelle reposant sur deux certificats. Une fois l'authentification réalisée, les clés de session de chiffrement et de HMAC sont calculées à l'aide de la fonction PRF de TLS (`key-method 2`) à partir de valeurs générées aléatoirement (OpenSSL `RAND_bytes`) par les deux parties et échangées dans la session TLS. Le certificat de la ToE est spécifié via `cert mycert.crt` et `key mykey.key` et le certificat de l'autorité de certification via `ca myca.crt`. Les quatre clés de session sont indépendantes et différentes : pour le HMAC d'émission, le HMAC de réception, le chiffrement et le déchiffrement. Pendant la période de renouvellement des clés, un paramètre de fenêtre de transition autorise un passage progressif des anciennes aux nouvelles clés de session, évitant tout ralentissement du trafic pendant la renégociation TLS.

Les éléments de configuration propres au mode TLS sont détaillés plus loin.

Mode TLS augmenté d'un utilisateur/mot de passe (configuration non retenue)

Cette configuration est identique au mode TLS, augmenté d'une authentification par un couple utilisateur/mot de passe (`auth-user-pass`) dont le mécanisme repose sur un plug-in `openvpn-auth-pam.so` (l'utilisation de l'alternative `auth-pam.pl` n'étant pas recommandée). Cette configuration apporte donc une double authentification, combinant ce que l'utilisateur possède (son certificat) à ce qu'il connaît (son mot de passe) et évitant ainsi la compromission du réseau protégé par VPN en cas de compromission de la configuration cliente et de son certificat. Le site officiel déconseille explicitement l'usage seul de `auth-user-pass` sans certificat client (`client-cert-not-required`).

Cependant, cette configuration n'apporte rien de plus au mode TLS simple, pour autant qu'un veille bien à utiliser des clés privées chiffrées, protégées par mot de passe, cf *H10*. C'est pourquoi *cette configuration est écartée*.

2.3.4 Configuration du mode TLS

La configuration recommandée en exemple côté client préconise l'utilisation de paramètres de Diffie-Hellman générés par OpenSSL et de 2048 bits : `dh dh2048.pem`.

La configuration recommandée en exemple côté client ajoute la vérification sur l'usage du certificat du serveur (`remote-cert-tls server`) pour éviter une attaque où un autre client ferait

[6] https://openvpn.net/index.php/open-source/documentation/miscellaneous/78-static-key-mini-howto.html

passer son propre certificat (signé par la même autorité) comme étant un certificat de serveur. On considère donc qu'elle est **toujours active** sur la ToE conjointement à l'utilisation du mode TLS. Il est également possible d'utiliser des autorités différentes pour les certificats serveur et clients. Rappelons que `remote-cert-tls server` est équivalent à `remote-cert-ku --remote-cert-eku "TLS Web Server Authentication"`.

Le mode TLS peut être rendu plus robuste par deux modes exclusifs spécifiques à OpenVPN :

— *pare-feu* HMAC : l'utilisation d'une clé secrète partagée (PSK) entre le serveur et tous les clients, activant l'ajout d'un authentifiant HMAC de tous les paquets de la négociation TLS, ce qui prévient une série d'attaques visant la phase de négociation (déni de service, port scanning, vulnérabilités, etc.), cf. l'option `tls-auth ta.key {0,1}`.

— HMAC et sur-chiffrement : depuis OpenVPN 2.4.0, l'option `tls-crypt tc.key` peut remplacer l'option `tls-auth` pour apporter, outre le MAC des paquets de la négociation TLS, une couche supplémentaire de chiffrement de ces paquets, utile pour se prémunir d'une défaillance de la couche TLS mais également cacher les certificats utilisés et rendre le trafic OpenVPN plus difficile à reconnaître.

Le mode `tls-auth` est présent dans les fichiers donnés en exemple et recommandé par le site officiel, néanmoins le mode `tls-crypt`, plus récent et encore absent de la documentation du site officiel, est pressenti comme le successeur du mode `tls-auth`. On considère donc qu'une des deux options, `tls-auth` ou `tls-crypt`, est **toujours active** sur la ToE.

Il est possible d'utiliser une liste de révocation (CRL) via `crl-verify crl.pem`. Elle peut être complétée à la volée et les changements sont pris en compte immédiatement pour toutes les nouvelles connexions ainsi que les connexions existantes lors de leur re-négociation (une fois par heure par défaut). Pour forcer la révocation immédiate d'un client déjà connecté, il est possible de terminer sa connexion, par exemple en forçant une reconnexion de tous les clients.

Il est possible de configurer OpenVPN pour utiliser des cartes à puce pour le stockage de la clé privée du client, mais cette option n'est pas considérée dans cette évaluation.

Par défaut, OpenVPN supporte la négociation de la version de TLS. Les clients plus anciens que la v2.3.3 ne supportent que TLS 1.0. Un document spécifique au durcissement d'OpenVPN [7] préconise l'usage de `tls-version-min 1.2` mais on ne retrouve pas cette recommandation dans la configuration d'exemple ni dans le *howto*. Pour une utilisation sûre du produit, nous considérons que l'option spécifiant l'usage de TLS en version 1.2 est **toujours active** sur la ToE. OpenVPN supporte également TLS 1.3 depuis sa version 2.4.5. Cette configuration n'est pas considérée dans le cadre de l'évaluation. Les suites supportées sont détaillées dans le prochain chapitre.

2.3.5 Autres éléments de configuration

Les configurations explicitement déconseillées sur le site officiel ne seront pas considérées, et ce y compris celles déjà mentionnées précédemment et celles documentées comme étant obsolètes [8].

Les éléments de configuration non spécifiés doivent être envisagés dans les diverses combinaisons possibles, notamment (liste non exhaustive) :

— IPv4 / IPv6

— `dev tap` et mode *bridging*, `dev tun` et mode *routing* (défaut), `dev tap` et mode *routing*

— configurations spécifiques à certains clients, cf. `client-config-dir`

[7] https://community.openvpn.net/openvpn/wiki/Hardening

[8] https://community.openvpn.net/openvpn/wiki/DeprecatedOptions

— règles de routage spécifiques, cf. `route`, `iroute`, `ifconfig-push` et la manière dont le client accepte ou non par défaut des directives peut-être intrusives
— Tous les moyens de pousser des éléments du serveur vers le client (`push`, `client-config-dir`, `route`, `iroute`, `ifconfig-push`, `pull-filter`, `push-remove`, etc.)
— `http-proxy`
— Keying Material Exporter [RFC-5705]
— `client-to-client` (par défaut les clients ne voient que le serveur)
— `float` : Seamless client IP/port floating
— Authentication tokens

2.4 Utilisateurs typiques

La liste des types d'utilisateurs susceptibles d'interagir avec la ToE est la suivante :

U1a : Administrateur serveur et client

Utilisateur ayant les droits de modifier la configuration de la ToE côté serveur et côté client. C'est un utilisateur ayant une connaissance fine des principaux concepts de l'informatique et des réseaux, une capacité à configurer et administrer un parc d'ordinateurs reliés en réseau, mais pas nécessairement une connaissance des moyens cryptographiques mis en œuvre.

U1b : Administrateur serveur uniquement

Utilisateur ayant les droits de modifier la configuration de la ToE côté serveur. C'est un utilisateur ayant une connaissance fine des principaux concepts de l'informatique et des réseaux, une capacité à configurer et administrer des serveurs, mais pas nécessairement une connaissance des moyens cryptographiques mis en œuvre. Les utilisateurs restent maîtres de leur poste de travail (*U3a*).

U2 : Auditeur

Utilisateur ayant le droit de consulter tout ou partie des journaux d'événements produits par la ToE côté serveur et client.

U3a : Utilisateur avec droit d'administration locale

Utilisateur grand public, sans compétence particulière en informatique. Il est chargé de l'utilisation journalière du poste client nomade et vraisemblablement de l'activation et de la désactivation du tunnel configuré dans la ToE client. Il est amené à installer lui-même la ToE client sur sa machine et à la configurer selon les prescriptions de son Adminstrateur *U1b* et de la documentation de la ToE client. Il est administrateur local de son poste de travail.

U3b : Utilisateur sans droit d'administration locale

Utilisateur grand public, sans compétence particulière en informatique. Il est chargé de l'utilisation journalière du poste client nomade et vraisemblablement de l'activation et de la désacti-

vation du tunnel configuré dans la ToE client. Il n'est pas administrateur local de son poste de travail, seul son Administrateur *U1a* l'est.

U4 : Utilisateur tiers sans droit d'administration locale (ToE serveur)

Utilisateur tiers ayant un compte non privilégié sur la machine hébergeant la ToE serveur.

U5 : Utilisateur tiers sans droit d'administration locale (ToE client)

Utilisateur tiers ayant un compte non privilégié sur la machine hébergeant la ToE client.

3. Description des éléments cryptographiques disponibles

De nombreuses options et sous-options sont en rapport avec les choix des éléments cryptographiques, certaines n'étant là que pour assurer une rétrocompatibilité. Afin de limiter les choix et donc l'analyse de la ToE, nous avons suivi ces principes :

- Utiliser exclusivement TLS 1.2 ;
- Écarter les options de rétrocompatibilité qui désactivent les mécanismes récents ou spécifient des algorithmes obsolètes ;
- Restreindre volontairement les suites cryptographiques en écartant SHA1 et le mode CBC.

3.1 Chiffrement et intégrité des flux du canal de contrôle

Déterminés par la suite TLS négociée, voir plus bas.

L'intégrité est assurée selon le modèle *encrypt-then-MAC* ou par chiffrement AEAD (*GCM*) selon la suite négociée.

L'intégrité est assurée également pour l'authentification de l'échange TLS par l'usage de l'option `tls-auth` ou de l'option `tls-crypt`.

Un chiffrement supplémentaire assure la confidentialité de l'échange TLS si l'option `tls-crypt` est activée.

3.2 Chiffrement et intégrité des flux du canal de données

Liste à négocier configurable via l'option `ncp-ciphers`.

Négociée, défaut : la première disponible.

Liste par défaut : `AES-256-GCM:AES-128-GCM`.

Il est nécessaire que client et serveur soient configurés avec la négociation activée (donc ne pas utiliser l'option `ncp-disable`), sans quoi les options `cipher` et `auth` dicteront les mécanismes utilisés.

La valeur par défaut de `cipher` est `BF-CBC`, et celle de `auth` est `SHA1`. Il vaut mieux spécifier des valeurs plus sûres au cas où, pour une raison quelconque, la ToE est mise en communication avec une autre entité OpenVPN qui n'a pas de négociation activée : `cipher AES-256-GCM` et `auth SHA256`.

3.3 Suite cryptographique TLS

Configurable via l'option `tls-cipher`.

Négociée, défaut : la première disponible.

Valeurs possibles :

```
$ openvpn --show-tls
Available TLS Ciphers,
listed in order of preference:

TLS-ECDHE-ECDSA-WITH-AES-256-GCM-SHA384
TLS-ECDHE-RSA-WITH-AES-256-GCM-SHA384
TLS-DHE-RSA-WITH-AES-256-GCM-SHA384
TLS-ECDHE-ECDSA-WITH-CHACHA20-POLY1305-SHA256
TLS-ECDHE-RSA-WITH-CHACHA20-POLY1305-SHA256
TLS-DHE-RSA-WITH-CHACHA20-POLY1305-SHA256
TLS-ECDHE-ECDSA-WITH-AES-128-GCM-SHA256
TLS-ECDHE-RSA-WITH-AES-128-GCM-SHA256
TLS-DHE-RSA-WITH-AES-128-GCM-SHA256
TLS-ECDHE-ECDSA-WITH-AES-256-CBC-SHA384
TLS-ECDHE-RSA-WITH-AES-256-CBC-SHA384
TLS-DHE-RSA-WITH-AES-256-CBC-SHA256
TLS-ECDHE-ECDSA-WITH-AES-128-CBC-SHA256
TLS-ECDHE-RSA-WITH-AES-128-CBC-SHA256
TLS-DHE-RSA-WITH-AES-128-CBC-SHA256
TLS-ECDHE-ECDSA-WITH-AES-256-CBC-SHA
TLS-ECDHE-RSA-WITH-AES-256-CBC-SHA
TLS-DHE-RSA-WITH-AES-256-CBC-SHA
TLS-ECDHE-ECDSA-WITH-AES-128-CBC-SHA
TLS-ECDHE-RSA-WITH-AES-128-CBC-SHA
TLS-DHE-RSA-WITH-AES-128-CBC-SHA
```

En écartant SHA1, le mode CBC et ChaCha20/Poly1305, la liste se réduit à :

```
TLS-{ECDHE-ECDSA,ECDHE-RSA,DHE-RSA}-WITH-AES-256-GCM-SHA384
TLS-{ECDHE-ECDSA,ECDHE-RSA,DHE-RSA}-WITH-AES-128-GCM-SHA256
```

Il existe une option `tls-cert-profile` avec trois niveaux : `legacy` (default), `preferred` (défaut dans une version future) et `suiteb`. Cependant, nous ne ferons pas usage de cette option car les suites et courbes retenues dans la cible forment un ensemble à mi-chemin entre les profils `preferred` et `suiteb` et l'implémentation de l'option `tls-cert-profile` vise principalement mbed TLS, son application à OpenSSL est une suite d'approximations difficilement maîtrisable.

3.4 Diffie Hellman

La documentation officielle d'OpenVPN conseille de générer et d'utiliser des clés de 2048 bits : `dh dh2048.pem`.

3.5 Diffie Hellman sur courbe elliptique

Configurable via l'option `ecdh-curve`.

Défaut : La courbe utilisée par défaut par OpenSSL (`SSL_CTX_set_ecdh_auto` pour OpenSSL >= 1.0.2).

Valeurs possibles : 81 courbes des familles `secp`, `prime`, `sect`, `c2xnb`, `wap-wsg-idm-ecid-wtls`, `Oakley-EC2N` et `brainpool`.

Nous en retiendrons les courbes suivantes, selon les recommendations de sécurité relatives à TLS publiées par l'ANSSI :

```
prime256v1 (secp256r1)
secp384r1
secp521r1
brainpoolP256r1
brainpoolP384r1
brainpoolP512r1
```

3.6 Génération d'aléa

OpenVPN implémente son propre PRNG par-dessus OpenSSL `RAND_bytes` pour des raisons de performances. L'option `prng` permet de le paramétrer, voire de le désactiver pour s'en remettre complètement à `RAND_bytes`.

Par défaut, on a `prng sha1`. Il est acceptable de le désactiver (`prng none`) mais pas de l'affaiblir (par ex. `prng md4`).

4. Environnement de sécurité de la cible d'évaluation

4.1 Biens sensibles de l'environnement

B1 : Flots de données dans les tunnels (intégrité, authenticité, confidentialité)

La ToE protège en intégrité, en authenticité et en confidentialité les flots de données transitant dans un tunnel VPN dont elle est une des extrémités.

B2a : Authenticité des clients (authenticité)

La ToE serveur assure l'authenticité des clients avec lesquels elle établit un tunnel VPN.

B2b : Authenticité des serveurs (authenticité)

La ToE client assure l'authenticité du serveur avec lequel elle établit un tunnel VPN.

B3 : Politique de sécurité VPN (disponibilité, intégrité)

La ToE permet de définir une politique de sécurité pour les flux de données avec les clients nomades ou les serveurs. Cette politique permet notamment de

- définir la liste des pairs pour les tunnels ;
- définir la liste des flux de données pouvant ou devant transiter dans les tunnels pour chaque pair ;
- définir les propriétés de sécurité requises et les algorithmes cryptographiques à utiliser pour chaque tunnel.

Les réseaux locaux des ToE doivent donc rester imperméables à toute personne non autorisée par la politique de sécurité.

4.2 Biens sensibles de la ToE

B4 : Code (intégrité, authenticité)

Afin d'assurer correctement ses fonctions, le code de la ToE doit être intègre et authentique.

B5 : Configuration (intégrité, confidentialité)

La configuration de la ToE doit être confidentielle et intègre. L'attaquant ne doit pas pouvoir découvrir cette configuration autrement que par l'observation de l'activité de la ToE.

B6 : Mécanisme d'authentification des clients (intégrité, authenticité)

Ce mécanisme s'appuie sur une PKI (donc un certificat d'une autorité de certification, la confidentialité n'est alors pas requise). La ToE doit protéger l'intégrité et l'authenticité du mécanisme.

B7a : Secrets de connexion des clients (intégrité, confidentialité)

Une clé privée asymétrique et une clé PSK `tls-hmac` sont contenues dans la ToE client et un mot de passe protège le stockage de la clé privée. La ToE client doit garantir l'intégrité et la confidentialité de ces éléments.

B7b : Secrets de connexion du serveur (intégrité, confidentialité)

Une clé privée asymétrique et une clé PSK `tls-hmac` sont contenues dans la ToE serveur et un mot de passe protège le stockage de la clé privée. La ToE serveur doit garantir l'intégrité et la confidentialité de ces éléments.

B7c : Éléments publics (intégrité)

Les différents éléments publics utiles pour les tunnels VPN doivent être protégés en intégrité. Il s'agit des éléments publics permanents utilisés lors de l'établissement du tunnel (certificats client et serveur, paramètres Diffie-Hellman).

B8 : Éléments secrets VPN (intégrité, confidentialité)

Outre les éléments mentionnés en *B7a* et *B7b*, les différents éléments secrets utiles pour les tunnels VPN doivent être protégés en confidentialité et en intégrité, notamment les éléments temporaires comme une clé de session par exemple.

B9 : Politique de gestion des droits (intégrité)

Cette politique peut être contenue en local sur la ToE. La ToE doit garantir l'intégrité de cette politique de gestion des droits.

B10 : Fonction de journalisation locale (disponibilité, intégrité)

La ToE dispose d'une fonction de journalisation locale qui, une fois configurée, doit rester opérationnelle (disponibilité). La ToE est responsable de la génération des messages destinés aux journaux des événements (intégrité) jusqu'à leur envoi au système de journalisation de l'OS. Ces messages sortent alors du périmètre d'évaluation.

4.3 Hypothèses d'utilisation sécurisée

H1 : Documentation de sécurité

La ToE est fournie avec une documentation détaillée sur l'utilisation sécurisée de l'équipement. L'ensemble des préconisations issues de cette documentation ont été appliquées en vue de l'évaluation.

H2 : Activation des journaux

Les fonctions de journalisation locale sont supposées fonctionnelles. Les journaux locaux sont supposés intègres et authentiques.

H3 : Consultation des journaux

Il est considéré que les administrateurs consultent régulièrement les journaux de la ToE serveur. Il n'est pas réaliste de faire la même hypothèse concernant les journaux de la ToE client.

H4 : Administrateurs du serveur

Les administrateurs de la ToE serveur sont compétents, formés et non hostiles. Il n'est pas réaliste de faire la même hypothèse concernant les administrateurs locaux *U3a* de la ToE client vis-à-vis de la ToE serveur.

H5 : Environnement du serveur

La ToE serveur est installée sur un serveur protégé physiquement, réputé mis à jour régulièrement, en particulier en ce qui concerne les dépendances logicielles de la ToE. L'accès physique est limité aux seules personnes autorisées considérées comme non hostiles. Les éventuels utilisateurs non privilégiés du serveur *U5* sont considérés comme potentiellement hostiles.

H6 : Administrateurs du client

Les administrateurs de la ToE client sont compétents et formés (*U1a*) et non hostiles vis-à-vis de la ToE client dans tous les cas (*U1a* et *U3a*). On ne peut pas faire la même hypothèse concernant les administrateurs serveur *U1b* vis-à-vis de la ToE client administrée par un *U3a*.

H7 : Environnement du client

Le poste du client nomade est supposé sain, mis à jour régulièrement, en particulier en ce qui concerne les dépendances logicielles de la ToE. Il ne peut être compromis par un attaquant durant l'évaluation sauf du fait d'une défaillance de la ToE. Les éventuels utilisateurs non privilégiés du client *U3b* ou *U4* sont considérés comme potentiellement hostiles et pourraient

profiter d'une faiblesse de la ToE client.

H8 : Politique de filtrage

Aucune supposition n'est faite sur une éventuelle politique de filtrage autour de la ToE.

H9 : Dimensionnement

Il est supposé que la ToE est dimensionnée correctement pour les traitements qu'elle doit effectuer.

H10 : Gestion des clés et certificats

Il est supposé que les clés sont gérées correctement par les administrateurs et leur gestion n'entre pas dans le périmètre à évaluer. Les clés en elles-mêmes restent bien évidemment des biens à protéger (*B7a* ou *B7b*) au sein de la ToE.

Une gestion correcte implique notamment :

- D'utiliser des clés de taille suffisante. Pour RSA, OpenVPN se réfère aux recommandations d'ENISA : minimum 2048 bits et 3072 bits ou plus si on souhaite une assurance au-delà de dix ans.
- De protéger les clés privées des certificats par chiffrement et mot de passe de qualité suffisante.
- D'émettre des certificats avec des *usages de clé* et *usages étendus de clé* en accord avec la RFC3280 : les certificats clients doivent avoir comme usages `digitalSignature + keyAgreement` et comme usage étendu `TLS Web Client Authentication`, tandis que les certificats serveurs doivent avoir comme usage `digitalSignature + keyAgreement` et comme usage étendu `TLS Web Server Authentication`.
- De choisir des suites cryptographiques compatibles avec TLS 1.2 ou supérieur.
- De ne pas laisser la clé privée de l'autorité de certification sur la ToE.

H11 : Services non évalués désactivés par défaut

L'ensemble des services présents dans la ToE, mais hors de la cible de sécurité sont désactivés dans la configuration par défaut.

4.4 Menaces sur la sécurité

Les menaces censément couvertes sont élargies aux attaquants ayant le plus de privilèges admis qui arriveraient à toucher des biens qui ne leur sont pas accessibles. Lister les différentes capacités de l'attaquant dans chaque cas a pour but de permettre aux évaluateurs de vérifier qu'aucun chemin d'attaque n'a été oublié et de pouvoir compléter le cas échéant.

Les menaces suivantes ont été retenues :

M1 : Déni de service

L'attaquant parvient à effectuer un déni de service sur la ToE, donc une compromission de la disponibilité de la politique de sécurité *B3*, en effectuant une action imprévue ou en exploitant une vulnérabilité (envoi d'une requête malformée, utilisation d'un fichier de configuration corrompu...). Ce déni de service peut concerner toute la ToE ou seulement certaines de ses fonctions.

L'attaquant a (une combinaison de) certaines capacités :
- un tiers placé en coupure sur le réseau non maîtrisé.
- un utilisateur local ayant des privilèges limités (*U4* ou *U5*).
- un utilisateur client sans droits administratifs (*U3b*).
- un utilisateur client (*U3a*) peut tenter un déni de service envers la ToE serveur et les autres clients.

M2 : Violation de la politique de sécurité VPN

L'attaquant parvient à contourner la politique de sécurité VPN *B3* en effectuant, par exemple, une des actions suivantes :
- faire transiter du trafic non prévu dans un tunnel ;
- faire transiter du trafic devant être protégé hors d'un tunnel ;
- modifier les caractéristiques d'un tunnel.

Cela inclut le cas trivial où un déni de service sur l'acheminement du trafic chiffré entraîne l'arrêt du VPN et l'envoi du trafic client à protéger en clair sur le réseau Internet local (cas d'un VPN vers une passerelle Internet).

L'attaquant a (une combinaison de) certaines capacités :
- un tiers placé en coupure sur le réseau non maîtrisé.
- un utilisateur local ayant des privilèges limités (*U4* ou *U5*).
- un utilisateur client sans droits administratifs (*U3b*).
- un utilisateur client (*U3a*) peut tenter de modifier les caractéristiques de son tunnel (par exemple l'affaiblir volontairement).
- un utilisateur client (*U3a*) vis-à-vis des autres clients.

M3 : Corruption du code de la ToE

L'attaquant parvient à injecter et faire exécuter un code malveillant sur la ToE. L'injection de code peut être temporaire ou permanente et ceci inclut donc toute exécution de code non prévue ou non autorisée (*B4*).

L'attaquant a (une combinaison de) certaines capacités :
- un tiers placé en coupure sur le réseau non maîtrisé.
- un utilisateur local ayant des privilèges limités (*U4* ou *U5*).
- un utilisateur client sans droits administratifs (*U3b*).
- un utilisateur client (*U3a*) peut tenter de modifier le code de la ToE serveur ou d'autres clients.
- un administrateur serveur (*U1b*), sans être administrateur des postes clients, peut tenter de modifier le code de la ToE client.

M4 : Corruption de la configuration

L'attaquant parvient à modifier, de façon temporaire ou permanente, la configuration *B5* de la ToE.

L'attaquant a (une combinaison de) certaines capacités :
- un tiers placé en coupure sur le réseau non maîtrisé.
- un utilisateur local ayant des privilèges limités (*U4* ou *U5*).
- un utilisateur client sans droits administratifs (*U3b*).
- un utilisateur client (*U3a*) peut tenter de modifier la configuration de la ToE serveur ou d'autres clients.
- un administrateur serveur (*U1b*), sans être administrateur des postes clients, peut tenter de reconfigurer une ToE client (par exemple pour configurer des flux supplémentaires et pénétrer le réseau local du client).

M5 : Compromission de la configuration

L'attaquant parvient à récupérer tout ou partie de la configuration *B5* de la ToE de manière illégitime, outre ce qu'il est permis de deviner de la configuration par observation externe des flux.

L'attaquant a (une combinaison de) certaines capacités :
- un tiers placé en coupure sur le réseau non maîtrisé.
- un utilisateur local ayant des privilèges limités (*U4* ou *U5*).
- un utilisateur client (*U3a*) peut tenter d'accéder à la configuration de la ToE serveur ou d'autres clients. Il est entendu qu'il a un accès légitime à sa propre configuration.

M6a : Vol d'identifiants client

L'attaquant parvient à récupérer les secrets de connexion *B7a* et/ou *B8* d'un utilisateur.

L'attaquant a (une combinaison de) certaines capacités :
- un tiers placé en coupure sur le réseau non maîtrisé.
- un utilisateur local ayant des privilèges limités (*U4* ou *U5*).
- un administrateur de ToE serveur (*U1b*), s'il n'est pas chargé de la gestion des clés, peut tenter de récupérer une clé privée d'un utilisateur.
- un utilisateur client (*U3a*) vis-à-vis d'autres clients.

M6b : Vol des secrets côté serveur

L'attaquant parvient à récupérer les secrets de connexion *B7b* et/ou *B8* d'un serveur.

L'attaquant a (une combinaison de) certaines capacités :
- un tiers placé en coupure sur le réseau non maîtrisé.
- un utilisateur local ayant des privilèges limités (*U4*).
- un utilisateur client (*U3a*).

M7a : Contournement de l'authentification côté serveur

L'attaquant parvient à s'authentifier sans avoir les secrets de connexion, mettant à mal l'idée-même de client authentifié *B2a* et les mécanismes d'authentification *B6*, éventuellement par la corruption (par ex. injection) d'éléments d'authentification *B7b* ou *B8* ou encore le certificat de l'AC *B7c* sur le serveur.

L'attaquant a (une combinaison de) certaines capacités :
- un tiers placé en coupure sur le réseau non maîtrisé.
- un utilisateur local ayant des privilèges limités (*U4* ou *U5*).
- un utilisateur client (*U3a*) vis-à-vis d'autres clients (usurpation d'identité).

M7b : Contournement de l'authentification côté client

L'attaquant parvient à se faire passer pour un serveur légitime auprès d'un client sans avoir les secrets de connexion, mettant à mal l'idée-même de serveur authentifié *B2b* et les mécanismes d'authentification *B6*, éventuellement par la corruption (par ex. injection) d'éléments d'authentification *B7a* ou *B8* ou encore le certificat de l'AC *B7c* sur le client.

L'attaquant a (une combinaison de) certaines capacités :
- un tiers placé en coupure sur le réseau non maîtrisé.
- un utilisateur local ayant des privilèges limités (*U4* ou *U5*).
- un utilisateur client (*U3a*) vis-à-vis d'autres clients (usurpation d'identité).

M8 : Compromission des flux

Pour les flux *B1* requérant la confidentialité, l'attaquant parvient à récupérer des informations en interceptant des échanges entre la ToE et un composant externe.

L'attaquant a (une combinaison de) certaines capacités :
- un tiers placé en coupure sur le réseau non maîtrisé.
- un utilisateur local ayant des privilèges limités (*U4* ou *U5*).
- un utilisateur client (*U3a*) vis-à-vis d'autres clients.

M9 : Altération des flux

L'attaquant parvient à modifier des échanges *B1* entre la ToE et un composant externe sans que cela ne soit détecté.

L'attaquant a (une combinaison de) certaines capacités :
- un tiers placé en coupure sur le réseau non maîtrisé.
- un utilisateur local ayant des privilèges limités (*U4* ou *U5*).
- un utilisateur client (*U3a*) vis-à-vis d'autres clients.

M10 : Contournement de la politique de droits

L'attaquant parvient à obtenir des droits *B9* qui ne lui sont pas normalement dévolus.

L'attaquant a (une combinaison de) certaines capacités :
- un tiers placé en coupure sur le réseau non maîtrisé.
- un utilisateur local ayant des privilèges limités (*U3b*, *U4* ou *U5*).
- un utilisateur client (*U3a*) vis-à-vis d'autres clients.
- un administrateur de ToE serveur (*U1b*) vis-à-vis des clients.

M11 : Corruption des journaux d'événements locaux

L'attaquant parvient à corrompre (injecter, effacer ou modifier) une entrée dans les journaux d'événements locaux de la ToE *B10* sans y avoir été autorisé par la politique de droits de la ToE.

L'attaquant a (une combinaison de) certaines capacités :
- un tiers placé en coupure sur le réseau non maîtrisé.
- un utilisateur local ayant des privilèges limités (*U4* ou *U5*).
- un utilisateur client (*U3a*) peut tenter de récupérer les journaux de la ToE serveur ou d'autres clients.

Remarque :

Il est entendu que si l'administrateur configure la ToE serveur ou client pour tolérer une interopérabilité avec des clients ou serveurs qui utiliseraient des versions antérieures et des moyens cryptographiques dépassés (par exemple via les options *cipher* et *ncp-ciphers*), il assume les risques encourus suite à l'utilisation de moyens cryptographiques non recommandés. Néanmoins, dans ce cadre, l'exploitation de failles présentes dans le support de ces moyens qui seraient propres à l'implémentation qui est faite dans la ToE reste une menace à prendre en compte.

5. Objectifs de sécurité

Les objectifs de sécurité retenus sont les suivants :

O1 : Gestion des entrées malformées

La ToE a été développée de manière à gérer correctement les entrées malformées, en particulier en provenance du réseau.

O2 : Politique de sécurité VPN

La ToE veille au respect de la politique de sécurité configurée.

O3 : Connexion sécurisée entre le serveur et le client

La ToE permet une connexion sécurisée entre le serveur et le client en assurant l'authenticité des deux extrémités, l'intégrité et la confidentialité des échanges, ainsi que le non-rejeu.

O4 : Politique de droits

La politique de gestion des droits est gérée de manière extrêmement stricte. L'implémentation de cette politique permet en particulier de garantir l'authenticité des opérations critiques, c'est-à-dire pouvant porter atteinte aux biens sensibles identifiés.

O5 : Intégrité et confidentialité de la configuration

La politique de gestion des utilisateurs ne permet à une personne non autorisée, ni de consulter, ni de modifier tout ou partie de la configuration de la ToE.

O6 : Intégrité des journaux

Les journaux d'événements générés par la ToE sont intègres et seul l'administrateur peut les modifier.

Cette liste ne comporte pas d'objectif *Stockage sécurisé des secrets* car la ToE n'est pas maître du stockage des secrets et ne peut garantir seule que la compromission d'un fichier ne permet pas de les récupérer. Pour une utilisation sûre du produit, l'administrateur veillera à ce que le déploiement garantisse cet objectif en émettant exclusivement des certificats avec des clés privées chiffrées, protégées par mot de passe, cf *H10*. L'évaluateur vérifiera que la ToE permet bien l'utilisation de clés privées chiffrées et que les secrets de déchiffrement ne sont pas exposés pendant leur manipulation par la ToE.

6. Couvertures

6.1 Couverture des biens par les menaces

Biens	B1	B2a	B2b	B3	B4	B5	B6	B7a	B7b	B7c	B8	B9	B10
M1				D									
M2				I									
M3					IA								
M4						I							
M5						C							
M6a								C			C		
M6b									C		C		
M7a		A					IA		I	I	I		
M7b			A				IA	I		I	I		
M8	C												
M9	IA												
M10												I	
M11													I

6.2 Couverture des menaces par les objectifs de sécurité

Menaces	M1	M2	M3	M4	M5	M6a	M6b	M7a	M7b	M8	M9	M10	M11
O1	X		X										X
O2		X											
O3								X	X	X	X		
O4		X				X	X					X	
O5				X	X					X	X		
O6													X

7. Références

[CEM] : Common Methodology for Information Technology Security Evaluation : Evaluation Methodology, version en vigueur.

[RGS_B] : Référentiel général de sécurité, annexes B :

[RGS_B1] : Règles et recommandations concernant le choix et le dimensionnement des mécanismes cryptographiques.

[RGS_B2] : Règles et recommandations concernant la gestion des clés utilisées dans des mécanismes cryptographiques.

[RGS_B3] : Règles et recommandations concernant les mécanismes d'authentification.

[CRITERES] : Critères pour l'évaluation en vue d'une certification de sécurité de premier niveau, référence ANSSI-CSPN-CER-I-02, version en vigueur.

[PP_VPNa] : Profil de protection d'une passerelle VPN industrielle, Version 1.0 court-terme, GTCSI, 13 juillet 2015

[PP_VPNb] : Profil de protection d'une passerelle VPN industrielle, Version 1.0 moyen-terme, GTCSI, 13 juillet 2015

[TLS_ANSSI] : Recommandations de sécurité relatives à TLS, SDE-NT-35/ANSSI/SDE/NP v1.1, 19 août 2016

[RFC3280] : Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, April 2002

8. Résumé des configurations minimales retenues

8.1 Configuration minimale côté serveur

```
local a.b.c.d
port 1194
proto udp
dev tun / dev tap
ca ca.crt
cert server.crt
key server.key
dh dh2048.pem
topology subnet
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt
keepalive 10 120
# activer soit tls-auth, soit tls-crypt :
tls-auth ta.key 0
#tls-crypt tc.key
user nobody
group nobody
persist-key
persist-tun
status openvpn-status.log
verb 3
explicit-exit-notify 1
# anti-rejeu entre sessions
replay-persist file
# spécifications cryptographiques
tls-version-min 1.2
tls-version-max 1.2
tls-cipher TLS-ECDHE-ECDSA-WITH-AES-256-GCM-SHA384:TLS-ECDHE-RSA-WITH-AES-256-GCM-
↪SHA384:TLS-DHE-RSA-WITH-AES-256-GCM-SHA384:TLS-ECDHE-ECDSA-WITH-AES-128-GCM-
↪SHA256:TLS-ECDHE-RSA-WITH-AES-128-GCM-SHA256:TLS-DHE-RSA-WITH-AES-128-GCM-SHA256
ncp-ciphers AES-256-GCM:AES-128-GCM
cipher AES-256-GCM
auth SHA256
# choix parmi prime256v1, secp{384,521}r1, brainpoolP{256,384,512}r1
ecdh-curve brainpoolP256r1
# PRNG : none, sha1 ou supérieur
prng sha1
# rejet des clients ayant des options incompatibles :
opt-verify
# compression optionnelle :
compress lz4-v2
push "compress lz4-v2"
# CRL optionnelle :
crl-verify crl.pem
```

8.2 Configuration minimale côté client

```
client
proto udp
dev tun / dev tap
remote my-server-1 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca ca.crt
cert client.crt
key client.key
remote-cert-tls server
# activer soit tls-auth, soit tls-crypt :
tls-auth ta.key 1
#tls-crypt tc.key
verb 3
# anti-rejeu entre sessions
replay-persist file
# spécifications cryptographiques
tls-version-min 1.2
tls-version-max 1.2
tls-cipher TLS-ECDHE-ECDSA-WITH-AES-256-GCM-SHA384:TLS-ECDHE-RSA-WITH-AES-256-GCM-
↪SHA384:TLS-DHE-RSA-WITH-AES-256-GCM-SHA384:TLS-ECDHE-ECDSA-WITH-AES-128-GCM-
↪SHA256:TLS-ECDHE-RSA-WITH-AES-128-GCM-SHA256:TLS-DHE-RSA-WITH-AES-128-GCM-SHA256
ncp-ciphers AES-256-GCM:AES-128-GCM
cipher AES-256-GCM
auth SHA256
# choix parmi prime256v1, secp{384,521}r1, brainpoolP{256,384,512}r1
ecdh-curve brainpoolP256r1
# PRNG : none, sha1 ou supérieur
prng sha1
```

8.3 Options proscrites

```
secret
ncp-disable
client-cert-not-required
setenv FORWARD_COMPATIBLE 1
ignore-unknown-option
script-security >= 2
disable-occ
keysize
no-replay
no-iv
key-method
compat-names
no-name-remapping
reneg-bytes 0
ns-cert-type
```

8.4 Options hors cible

```
auth-user-pass*
pkcs11*
tls-cert-profile
```