Référence technique du protocole Tentacle

Référence technique du protocole Tentacle

Qu'est-ce que Tentacle ?

Tentacle, un outil de transfert de fichiers client/serveur, est :

- Sécurisé par conception.
- Facile à utiliser et à intégrer à d'autres outils.
- Polyvalent, flexible et multiplateforme.

Tentacle a été créé pour remplacer des outils plus complexes tels que SCP/SSH et FTP afin d'effectuer des transferts de fichiers simples et cesser d'utiliser des mécanismes d'authentification faibles tels que .netrc, ainsi que des connexions interactives automatisées avec expect et le mécanisme de clé SSH, pour passer à l'authentification basée sur la norme X.509, à l'aide de certificats.

Le client et le serveur ont été conçus pour être exécutés depuis la ligne de commande ou appelés depuis un shellscript. Tentacle est la méthode de transfert par défaut pour Pandora FMS depuis 2008, remplaçant SCP.

Tentacle est implémenté en Perl et ANSI C (les deux plates-formes incluses dans MS Windows®).

Vous pouvez le télécharger et obtenir plus d'informations sur le Site officiel du projet sur SourceForge.

Guide de l'utilisateur de Tentacle sur GNU Linux

Installation de la version Perl

Installation à partir de Source Forge Net

Pour installer le serveur Tentacle, vous devez disposer de droits équivalents à l'utilisateur root, après l'avoir installé, vous pouvez l'exécuter en tant qu'utilisateur standard.

Récupérez le fichier `tentacle_server-762.tar.gz` depuis Source Forge Net :

https://sourceforge.net/projects/pandora/files/Tools%20and%20dependencies%20%28Toutes%20versions%29/

Par exemple (vous devez avoir wget installé) :

```
wget
https://sourceforge.net/projects/pandora/files/Tools%20and%20dependencie
s%20%28All%20versions%29/tentacle_server-762.tar.gz
```

Installation sur Rocky Linux 8

- Décompressez le fichier téléchargé avec `tar xzvf tentacle_server-762.tar.gz` .
- Installez le langage Perl avec `dnf install perl` .
- Entrez dans le répertoire avec `cd tentacle` .
- Installez avec `./tentacle_server_installer –install` .

Installation sur CentOS 7

- Décompressez le fichier téléchargé avec `tar xzvf tentacle_server-762.tar.gz` .
- Installez le langage Perl avec `yum install perl perl-IO-Compress zlib` .
- Entrez dans le répertoire avec `cd tentacle` .
- Installez avec `./tentacle_server_installer –install` .

Installation depuis SVN

Le processus consiste à télécharger le code source à l'aide de Apache® Subversion® (svn) et à le compiler. Pour ce faire, vous devrez disposer des droits d'administrateur ou root (dans cette documentation, ce sont les lignes qui commencent par le caractère numérique #). Vous êtes seul responsable de ladite clé.

Pour installer les versions client et serveur, exécutez :

```
$ svn co http://svn.code.sf.net/p/tentacled/code/trunk/perl/ tentacle
$ cd tentacle
$ perl Makefile.PL
$ make
# make install
```

Pour installer uniquement la partie client, exécutez :

```
$ svn co http://svn.code.sf.net/p/tentacled/code/trunk/perl/client
$ cd client
$ perl Makefile.PL
$ make
# make install
```

Pour installer uniquement la partie serveur, exécutez :

```
$ svn co http://svn.code.sf.net/p/tentacled/code/trunk/trunk/perl/server
$ cd server
$ perl Makefile.PL
$ make
```

```
# make install
```

Si vous souhaitez installer dans un répertoire spécifique, remplacez :

```
$ perl Makefile.PL
```

par :

```
$ perl Makefile.PL PREFIX=/location
```

Installation manuelle

Si make n'est pas disponible sur votre système, vous pouvez effectuer l'installation manuellement en copiant les fichiers `tentacle_client` et `tentacle_server` dans le répertoire approprié (par exemple, `/usr/local/bin`).

Dans ce cas, si le binaire Perl ne se trouve pas dans `/usr/bin/perl`, éditez les deux fichiers Tentacle et modifiez la première ligne afin qu'elle pointe vers le chemin correct où se trouve votre binaire Perl. Ainsi, par exemple, remplacez `location` par l'emplacement de Perl sur le système à installer :

```
#!/location/perl
```

Installation de la version C

Installation à partir de SVN

En tenant compte du préambule d'installation dans la section précédente, pour installer le client Tentacle, exécutez :

```
$ svn co http://svn.code.sf.net/p/tentacled/code/trunk/c/ tentacle
$ cd tentacle
$ ./configure
$ make
# make install
```

Assurez-vous que le résultat de la commande configure ne génère aucune erreur, aucune dépendance d'en-tête incomplète, etc.

Pour désactiver le support OpenSSL, activé par défaut, remplacez :

```
$ ./configure
```

par :

```
 $ ./configure --disable-ssl
```

Exemples d'utilisation de Tentacle

Pour afficher les optionss'il est disponible, exécutez avec le paramètre `-h`, aussi bien dans la version client que dans la version serveur :

```
$ tentacle_client -h
Usage: tentacle_client [options] [file] [file] ...

Tentacle client v0.4.0.

Options:
       -a address      Server address (default 127.0.0.1).
       -b localaddress Local address to bind.
       -c              Enable SSL without a client certificate.
       -e cert         OpenSSL certificate file. Enables SSL.
       -f ca           Verify that the peer certificate is signed by a ca.
       -g              Get files from the server.
       -h              Show help.
       -k key          OpenSSL private key file.
       -p port         Server port (default 41121).
       -q              Quiet. Do now print error messages.
       -r number       Number of retries for network operations (default 3).
       -t time         Time-out for network operations in seconds (default 1s).
       -v              Be verbose.
       -w              Prompt for OpenSSL private key password.
       -x pwd          Server password.
       -y proxy        Proxy server string (user:password@address:port).
```

```
$ tentacle_server -h
Usage: /usr/local/bin/tentacle_server -s <storage directory> [options]

Tentacle server v0.6.2. See https://pandorafms.com/docs/ for protocol
description.

Options:
        -a ip_addresses IP addresses to listen on (default 0,0.0.0.0).
                        (Multiple addresses separated by comma can be defined.)
        -c number       Maximum number of simultaneous connections (default 10).
        -d              Run as daemon.
        -e cert         OpenSSL certificate file. Enables SSL.
        -f ca_cert      Verify that the peer certificate is signed by a ca.
        -F config_file  Configuration file full path.
        -h              Show help.
        -I              Enable insecure operations (file listing and moving).
        -i              Filters.
        -k key          OpenSSL private key file.
```

```
        -l log_file             File to write logs.
        -m size         Maximum file size in bytes (default 2000000b).
        -o              Enable file overwrite.
        -p port         Port to listen on (default 41121).
        -q              Quiet. Do now print error messages.
        -r number       Number of retries for network operations (default 3).
        -s Storage directory
        -S (install|uninstall|run) Manage the win32 service.
        -t time         Time-out for network operations in seconds (default 1s).
        -v              Be verbose (display errors).
        -V              Be verbose on hard way (display errors and other info).
        -w              Prompt for OpenSSL private key password.
        -x pwd          Server password.
        -b ip_address   Proxy requests to the given address.
        -g port         Proxy requests to the given port.
        -T              Enable tcpwrappers support.
                        (To use this option, 'Authen::Libwrap' should be
installed.)
```

Les valeurs par défaut de toutes les options seront également affichées dans l'aide.

Pour tous les exemples présentés ci-dessous, le serveur est situé à l'adresse `192.168.1.1` et la clé privée du client n'est pas protégée par mot de passe.

- Transfert simple d'un fichier limité à une taille maximale de 1 Mo et placé sur `/tmp` :

```
$ tentacle_server -m 1048576 -s /tmp -v
$ tentacle_client -a 192.168.1.1 -v /home/user/myfile.dat
```

- Transfert simple sur le port 65000 avec mode écrasement activé :

```
$ tentacle_server -o -p 65000 -s /tmp -v
$ tentacle_client -a 192.168.1.1 -p 65000 -v /home/user/myfile.dat
```

- Transfert simple avec authentification par mot de passe :

```
$ tentacle_server -x password -s /tmp -v
$ tentacle_client -a 192.168.1.1 -x password -v /home/user/myfile.dat
```

- Transfert sécurisé, sans certificat client :

```
$ tentacle_server -e cert.pem -k key.pem -w -s /tmp -v
$ tentacle_client -a 192.168.1.1 -c -v /home/user/myfile.dat
```

- Transfert sécurisé avec certificat client :

```
$ tentacle_server -e cert.pem -k key.pem -f cacert.pem -w -s /tmp -v
$ tentacle_client -a 192.168.1.1 -e cert.pem -k key.pem -v /home/user/myfile.dat
```

- Transfert sécurisé avec certificat client et authentification par mot de passe supplémentaire (notez l'utilisation du connecteur pour faciliter l'écriture de divers paramètres) :

```
$ tentacle_server -x password -e cert.pem -k key.pem -f cacert.pem -w -s /tmp -v
$ tentacle_client \
  -a 192.168.1.1 \
  -x password \
  -e cert.pem \
  -k key.pem \
  -v /home/user/myfile.dat
```

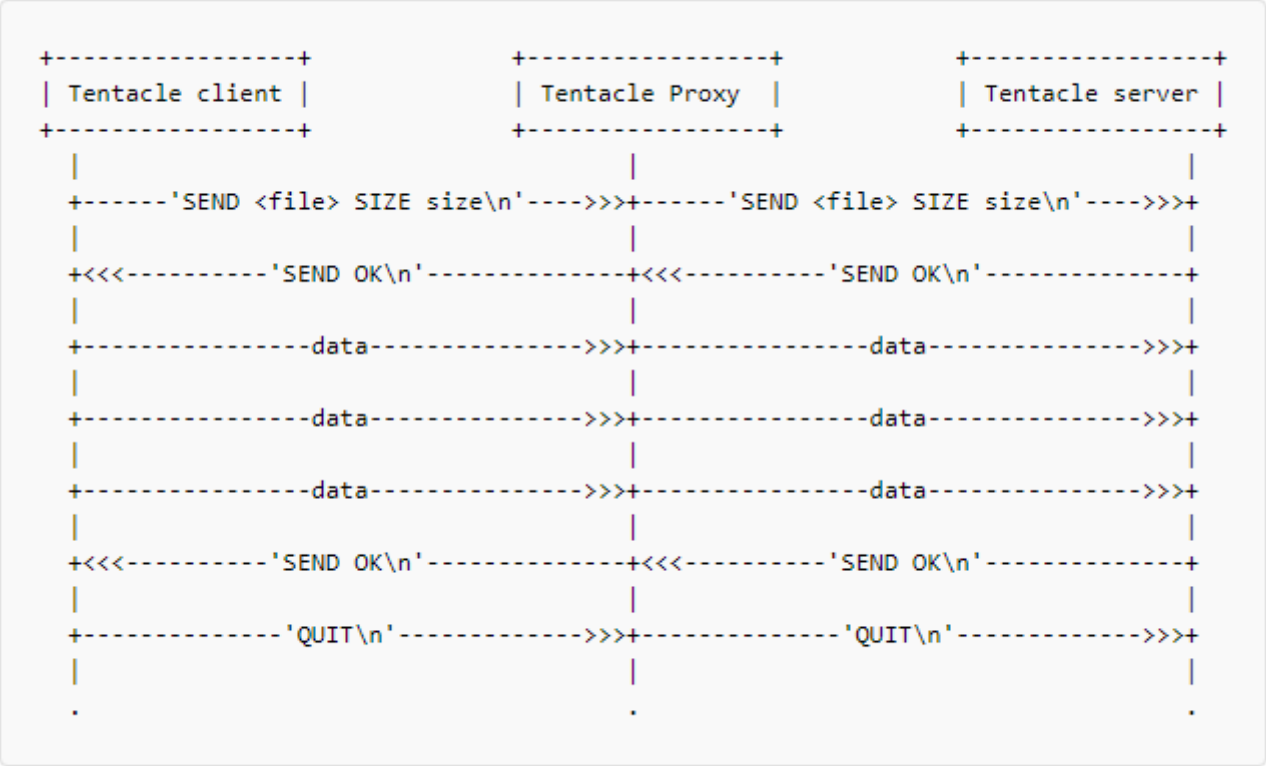
Le serveur Tentacle permet sa configuration à l'aide d'un fichier texte brut. Toutes les options de ligne de commande sont disponibles via ce fichier. Si la même option de configuration est spécifiée dans le fichier et sur la ligne de commande, la valeur ind sera prioricée dans ce dernier. Le chemin complet du fichier de configuration est indiqué avec l'option `-F`.

```
$ tentacle_server -F /etc/tentacle/tentacle_server.conf
```

Proxy Tentacle

Le serveur Tentacle peut fonctionner comme un proxy communiquant de nombreux clients Tentacle à un serveur Tentacle inaccessible.

Le diagramme suivant montre le fonctionnement du serveur proxy Tentacle :



Le proxy ne dispose d'aucune information, mais envoie uniquement les informations des clients au serveur Tentacle. Par exemple, pour lancer le serveur Tentacle en mode proxy utilisez les paramètres suivants :

```
$ tentacle_server -b 192.168.200.200 -g 65000
```

Ces paramètres sont l'adresse IP (`-b`) et le port (`-g`) du serveur Tentacle inaccessible. Ajoutez également les paramètres normaux sur une seule ligne :

```
$ tentacle_server -a 192.168.100.100 -p 45000 -b 192.168.200.200 -g 65000
```

Le protocole Tentacle en mode proxy supporte également
les paramètres d'authentification et de chiffrement.

Guide Tentacle pour MS Windows

Configurez et exécutez le client et le serveur Tentacle sur MS Windows®.

Installation de la version Perl

Installation de l'environnement Perl

À l'aide d'ActiveState®, téléchargez ActivePerl 5.8 à l'aide du lien suivant et exécutez le programme d'installation avec les options par défaut :

https://www.activestate.com/products/downloads/

Installation du module IO-Socket-SSL

Téléchargez et installez OpenSSL depuis :

http://gnuwin32.sourceforge.net/packages/openssl.htm

Téléchargez les modules Perl suivants :

- http://archive.apache.org/dist/perl/win32-bin/ppms/Net_SSLeay.pm.ppd
- http://archive.apache.org/dist/perl/win32-bin/ppms/IO-Socket-SSL.ppd

Exécutez depuis la ligne de commande dans le répertoire où se trouvent les fichiers `.ppd` :

```
> ppm install Net_SSLeay.pm.ppd> ppm install IO-Socket-SSL.ppd
```

Exécution du client et du serveur Tentacle

L'exécution est similaire à celle des systèmes Unix/Linux, il suffit d'enter d'abord la commande Perl, suivie de la syntaxe complète, par exemple :

```
> perl tentacle_client -v c:\file> perl tentacle_server -q -s c:\tmp
```

Définition du protocole Tentacle

Le protocole Tentacle lui-même est très simple et direct. Certaines caractéristiques de conception importantes sont :

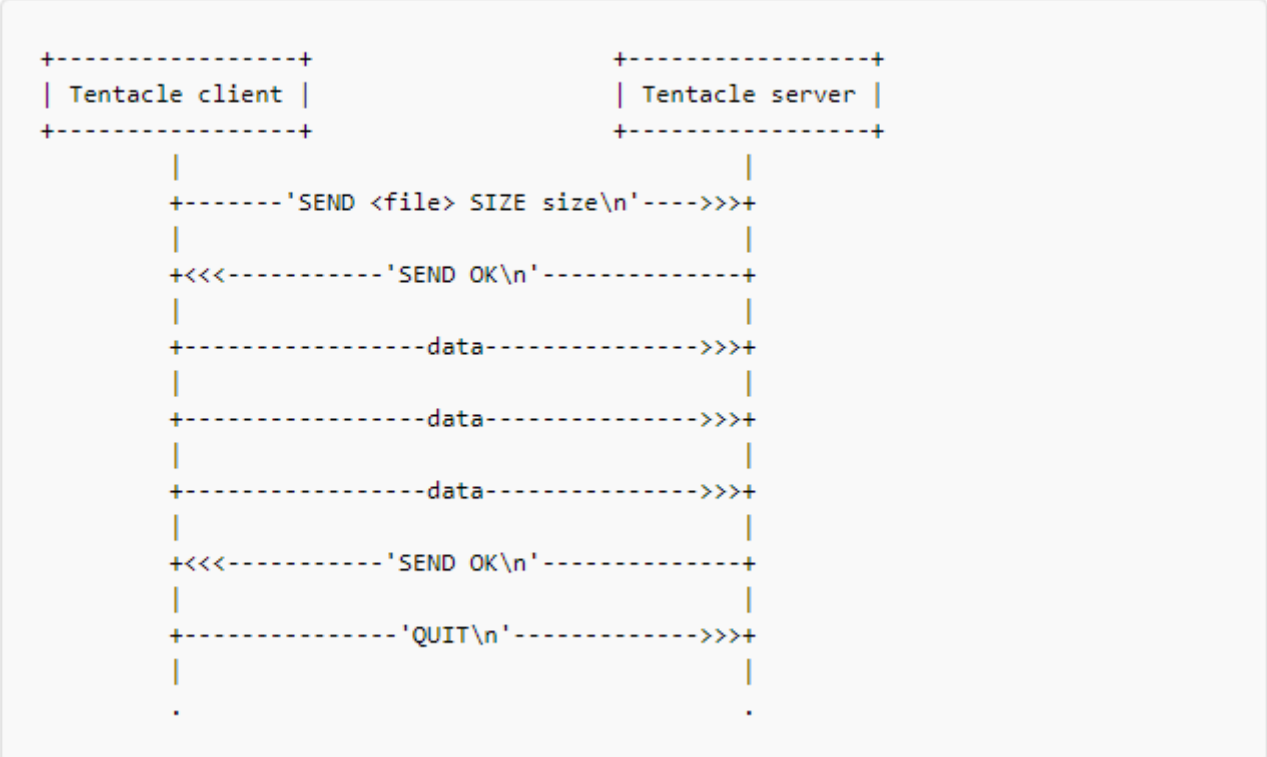
- La communication est toujours établie par le client.
- Les commandes se terminent toujours par un caractère de fin de ligne.
- Les caractères suivants ne peuvent pas faire partie d'un nom de fichier :

```
'?[]/\=+<>:;',*~'
```

Des diagrammes de séquence ASCII seront utilisés pour illustrer les cas possibles. Les commandes sont affichées entre guillemets simples.

Envoyer le(s) fichier(s)

Un transfert de fichier réussi s'affiche en premier



Pour permettre plusieurs transferts de fichiers au sein d'une même session, une nouvelle commande « SEND » doit être envoyée, après un transfert réussi, et avant une commande « QUIT ».

Si le serveur rejette un fichier, un message d'erreur générique est renvoyé au client. Pour des raisons de sécurité, les détails de l'échec de la commande ne sont pas affichés. Cela se produit lorsque :

- Le fichier a un nom de fichier invalide ou un chemin d'accès est spécifié.
- Est vide ou dépasse la taille maximale spécifiée par le serveur.
- Il existe déjà sur le serveur et l'écrasement des fichiers n'est pas activé.

Réception du fichier

Tentacle prend également en charge les demandes de fichiers du client.

Le client a la possibilité de rejeter le fichier après que le serveur l'a informé de sa taille.

Comme pour la commande « SEND », une nouvelle commande « RECV » peut être envoyée après un transfert réussi (même si le fichier a été rejeté par le client) et toujours avant le Commande « QUITTER ». Une erreur générique sera envoyée si le serveur refuse d'envoyer le fichier. Cette dernière peut survenir lorsque :

- A un nom de fichier invalide ou un chemin a été spécifié.
- N'existe pas sur le serveur.

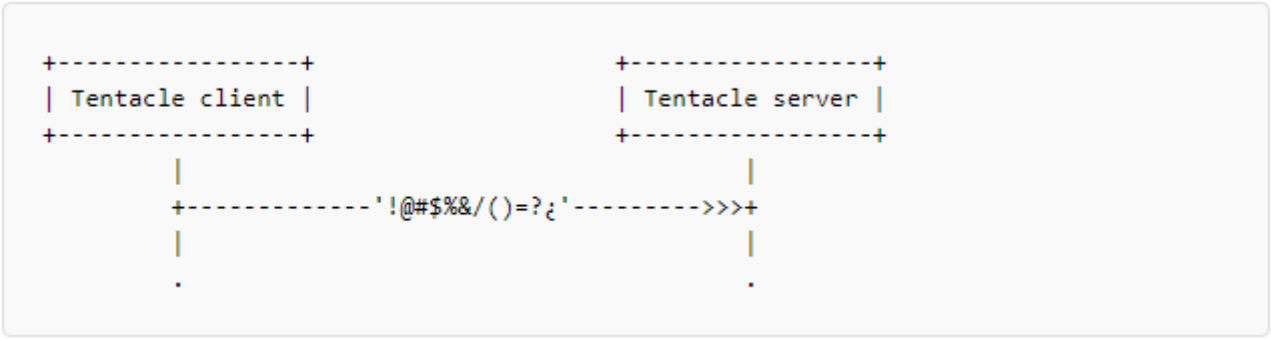
Authentification par mot de passe

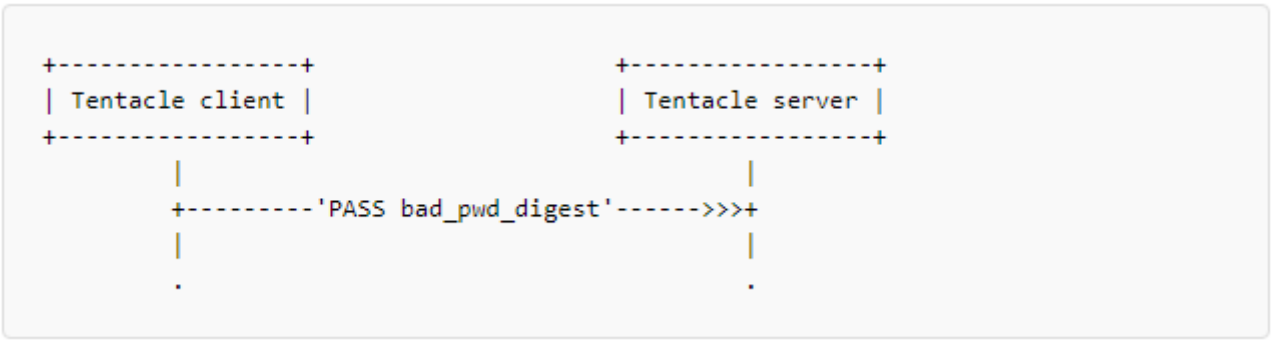
Si le serveur nécessite un mot de passe, le client doit s'authentifier avant d'envoyer toute autre commande.

Un double MD5 du mot de passe sera envoyé pour obscurcir. Si vous travaillez sur une connexion non cryptée, cela N'implémente PAS ni n'ajoute de sécurité. Si tu nSi vous avez besoin de sécurité, utilisez des connections chiffrées avec SSL .

Gestion des erreurs

En cas d'erreur, le serveur fermera la connexion sans donner aucune explication. Cela peut être dû à une commande incorrecte, à un mot de passe incorrect, à plus de données envoyées que prévu ou à toute autre raison qui amène le serveur à fonctionner en dehors de ce qui est établi ou considéré comme « normal ».





Par défaut, le journal Tentacle est défini sur `/dev/null`.

Guide rapide des certificats OpenSSL

Il s'agit d'un guide de démarrage rapide des certificats OpenSSL à utiliser avec Tentacle ou d'autres applications. Pour plus d'informations vous pouvez consulter le site officiel du projet OpenSSL :

https://www.openssl.org/docs/

Création d'un certificat

Préparation de l'environnement :

```
$ mkdir democA
$ mkdir democA/newcerts
$ mkdir democA/private
```

N'oubliez pas d'établir, pour des raisons de sécurité, les autorisations d'écriture et de lecture des différents utilisateurs de votre système sur les dossiers nouvellement créés.

L'étape suivante consiste à créer un certificat CA auto-signé et à le déplacer vers les répertoires créés :

```
$ openssl req -new -x509 -keyout cakey.pem -out cacert.pem
$ mv cakey.pem demoCA/private/
$ mv cacert.pem demoCA/
```

Remplissez les champs demandés pour le certificat et mémorisez-les bien car ils seront à nouveau nécessaires plus tard, exactement de la même manière. Vous devez maintenant créer une demande de certificat :

```
$ openssl req -new -keyout tentaclekey.pem -out tentaclereq.pem -days 360
```

Signez la demande de certificat, en établissant également une série consécutive de certificats comme mécanisme de contrôle et d'audit :

```
$ cat tentaclereq.pem tentaclekey.pem> tentaclenew.pem
$ touch demoCA/index.txt
$ echo "01">> demoCA/serial
$ openssl ca -out tentaclecert.pem -in tentaclenew.pem
```

Notez que si le random seed file présente des problèmes, vous pouvez le supprimer avec les droits d'utilisateur root : `sudo rm ~/.rnd`. De cette façon, il peut être recréé avec ses propres droits de lecture et d'écriture. Vous êtes seul responsable de ladite clé racine.

Créer un certificat auto-signé

```
$ openssl req -new -x509 -keyout tentaclekey.pem -out tentaclecert.pem -days 360
```

Générer une clé privée RSA

Ceci est très utile pour éviter d'avoir à saisir un mot de passe côté client à l'aide de Tentacle.

Générez la clé :

```
$ openssl genrsa -out tentaclekey.pem
```

Et remplacez `-keyout` par `-key` dans les sections précédentes.

Exporter le certificat vers un autre format

Des certificats peuvent être requis au format DER au lieu de PEM pour certains systèmes d'exploitation (tels que Ubuntu® ou Windows®). Si tel est le cas, vous pouvez obtenir le certificat dans ce format à partir du PEM généré :

```
openssl x509 -outform der -in tentaclecert.pem -out tentaclecert.der
```

Configuration d'une communication sécurisée avec Tentacle

Il explique étape par étape comment configurer les agents logiciels et les serveurs Tentacle pour une communication sécurisée.

Tout d'abord, il est fortement recommandé d'effectuer des tests manuels depuis les terminaux pour s'assurer que la configuration, les paramètres et les certificats sont corrects.

Ensuite, une configuration permanente peut être effectuée dans les fichiers de configuration respectifs :

Serveurs Tentacles

```
/etc/tentacle/tentacle_server.conf
```

Agents logiciels sous Unix/Linux

```
/etc/pandora/pandora_agent.conf
```

Agents logiciels sur MS Windows®

```
%ProgramFiles%\pandora_agent\pandora_agent.conf
```

Serveurs satellites

```
/etc/pandora/satellite_server.conf
```

Serveurs proxy Tentacle

```
/etc/tentacle/tentacle_server.conf
```

Pensez à redémarrer les services correspondants après toute modification. Dans le cas d'Unix/Linux vous pouvez également utiliser l'option `TENTACLE_EXT_OPTS` située dans `/etc/init.d/tentacle_serverd` (vous pouvez consulter le reste des options dudit démon dans ce lien).

Cryptage des communications

Pour crypter la communication entre les clients et le serveur Tentacle, il sera nécessaire de disposer au préalable de certificats et de clés SSL. Dans ce guide, nous verrons toutes les options de configuration possibles, les certificats peuvent donc être à la fois self-signed et signés par une autorité de certification valide.

Pour éviter toute confusion dans cet article, les certificats et les clés de chaque côté sont identifiés désignés par les noms suivants :

- `ca_cert` : Certificat de l'AC utilisée pour signer les certificats.
- `tentacle_key` : Clé générée pour le serveur Tentacle.
- `tentacle_cert` : Certificat généré pour le serveur Tentacle.
- `tentacle_client_key` : Clé générée pour le client Tentacle.
- `tentacle_client_cert` : Certificat généré pour le client Tentacle.

Il est TOUJOURS nécessaire d'indiquer dans les paramètres les chemins absolus où se trouvent les certificats, par exemple `/etc/ssl/tentaclecert.pem`

Pour utiliser les options sécurisées de Tentacle, veuillez vérifier que le package `perl(IO::Socket::SSL)` est installé sur votre système.

Configuration des certificats sur le serveur Tentacle acceptant n'importe quel certificat sur le client

Pour cette configuration vous devez indiquer le certificat et la clé utilisés pour le chiffrement dans la configuration du serveur Tentacle.

Exécuter manuellement sur le serveur avec les paramètres `-e` et `-k`

```
$ su - pandora -s /bin/bash
# tentacle_server -v -e tentacle_cert -k tentacle_key -s /tmp
```

Exécuter manuellement sur le client avec le paramètre `-c` :

```
$ echo test> file.txt
$ tentacle_client -v -c -a 192.168.70.125 file.txt
```

Si cette exécution manuelle fonctionne correctement, vous pouvez rendre la configuration

permanente dans le fichier correspondant :

- Pour un serveur Tentacle :

```
ssl_cert tentacle_cert
ssl_key tentacle_key
```

- Pour un agent logiciel :

```
server_opts -c
```

- Pour un serveur satellite :

```
server_opts -c
```

Configuration des certificats sur le serveur Tentacle et sur le client en vérifiant le certificat avec une autorité de certification spécifique sur le client

Pour cette configuration vous devez indiquer le certificat et la clé utilisés pour le chiffrement dans la configuration du serveur Tentacle ainsi que les certificats utilisés pour le chiffrement sur les clients.

Exécuter manuellement sur le serveur avec les paramètres `-e` et `-k`

```
# su - pandora -s /bin/bash
# tentacle_server -v -e tentacle_cert -k tentacle_key -s /tmp
```

Exécuter manuellement sur le client avec les paramètres `-e` et `-f` :

```
# echo test> file.txt
# tentacle_client -v -e tentacle_client_cert -f ca_cert -a 192.168.70.125
file.txt
```

Si cette exécution manuelle fonctionne correctement, vous pouvez rendre la configuration permanente dans le fichier correspondant :

- Pour un serveur Tentacle :

```
ssl_cert tentacle_cert
ssl_key tentacle_key
```

- Pour un agent logiciel :

```
server_opts -e tentacle_client_cert -f ca_cert
```

- Pour un serveur satellite :

```
server_opts -e tentacle_client_cert -f ca_cert
```

Configuration des certificats sur le serveur Tentacle et sur le client en vérifiant le certificat auprès d'une autorité de certification spécifique sur le serveur

Pour cette configuration vous devez indiquer les certificats et clés utilisés pour le chiffrement dans la configuration du serveur et des clients Tentacle.

Exécuter manuellement sur le serveur avec les paramètres `-e`, `-k` et `-f`

```
# su - pandora -s /bin/bash
# tentacle_server -v -e tentacle_cert -k tentacle_key -f ca_cert -s /tmp
```

Exécuté manuellement sur le client avec les paramètres `-e` et `-k` (notez l'utilisation du connecteur de ligne `\`) :

```
# echo test> file.txt
# tentacle_client -v \
           -e tentacle_client_cert \
           -k tentacle_client_key \
           -a 192.168.70.125 file.txt
```

Si cette exécution manuelle fonctionne correctement, vous pouvez rendre la configuration permanente dans le fichier correspondant :

- Pour un serveur Tentacle :

```
ssl_cert tentacle_cert
ssl_ca ca_cert
ssl_key tentacle_key
```

- Pour un agent logiciel :

```
server_opts -e tentacle_client_cert -k tentacle_client_key
```

- Pour un serveur satellite :

```
server_opts -e tentacle_client_cert -k tentacle_client_key
```

Configuration des certificats sur le serveur Tentacle et le client en vérifiant le certificat avec une autorité de certification spécifique sur les deux

Pour cette configuration vous devez indiquer les certificats et clés utilisés pour le chiffrement dans la configuration du serveur et des clients Tentacle.

Exécuter manuellement sur le serveur avec les paramètres `-e`, `-k` et `-f` :

```
# su - pandora -s /bin/bash
# tentacle_server -v -e tentacle_cert -k tentacle_key -f ca_cert -s /tmp
```

Exécuter manuellement sur le client avec les paramètres `-e`, `-k` et `-f` :

```
# echo test> file.txt
# tentacle_client -v \
         -e tentacle_client_cert \
         -k tentacle_client_key \
         -f ca_cert \
         -a 192.168.70.125 file.txt
```

Si cette exécution manuelle fonctionne correctement, vous pouvez rendre la configuration permanente dans le fichier correspondant :

- Pour un serveur Tentacle :

```
ssl_cert tentacle_cert
ssl_ca ca_cert
ssl_key tentacle_key
```

- Pour un agent logiciel :

```
server_opts -e tentacle_client_cert -k tentacle_client_key -f ca_cert
```

- Pour un serveur satellite :

```
server_opts -e tentacle_client_cert -k tentacle_client_key -f ca_cert
```

Configuration sécurisée de Tentacles

Le serveur Tentacle et les agents logiciels peuvent utiliser une communication sécurisée avec des certificats et un mot de passe, soit par communication directe entre les deux, soit via un serveur proxy Tentacle.

Il est TOUJOURS nécessaire d'indiquer dans les paramètres les chemins absolus où se trouvent les certificats, par exemple `/etc/ssl/tentaclecert.pem`

Pour utiliser les options sécurisées de Tentacle, veuillez vérifier que le package `perl(IO::Socket::SSL)` est installé sur votre système.

Dans les sections précédentes, les différentes combinaisons sont expliquées en détail ; Cette section ajoute des options de mot de passe, le serveur proxy Tentacle et l'utilisation de `TENTACLE_EXT_OPTS` pour définir les paramètres. Consultez également cette section ci-dessus pour connaître les noms de certificat et les clés de chaque côté. Une syntaxe simplifiée est utilisée

à des fins pédagogiques uniquement :

Transfert simple avec authentification par mot de passe :

Paramètre supplémentaire sur le serveur pour le mot de passe :

```
-x password
```

Paramètre supplémentaire sur le client pour le mot de passe (`TENTACLE_EXT_OPTS`) :

```
-x password
```

Transfert sécurisé, sans certificat client :

Paramètres supplémentaires sur le serveur :

```
-e tentacle_cert -k tentacle_key
```

Transfert sécurisé avec certificat client

Paramètres supplémentaires sur le serveur :

```
-e tentacle_cert -k tentacle_key -f ca_cert
```

Paramètres supplémentaires sur le client (`TENTACLE_EXT_OPTS`) :

```
-e tentacle_client_cert -k tentacle_client_key
```

Transfert sécurisé avec certificat client et authentification par mot de passe supplémentaire :

Paramètres supplémentaires sur le serveur :

```
-x password -e tentacle_cert -k tentacle_key -f ca_cert
```

Paramètres supplémentaires sur le client (`TENTACLE_EXT_OPTS`) :

```
-x password -e tentacle_client_cert -k tentacle_client_key
```

Cas d'utilisation de la configuration sécurisée avec le proxy Tentacle

Il explique étape par étape comment configurer les agents logiciels et le serveur Tentacle pour une communication sécurisée, également à l'aide d'un serveur proxy Tentacle.

Tests manuels :

1. Démarrez `tentacle_server` manuellement :

```
sudo -u //user//  tentacle_server \
          -x password \
          -e tentacle_cert \
          -k tentacle_key \
          -f ca_cert -s /tmp -v
```

2. Démarrez le proxy manuellement :

```
sudo -u //user//  tentacle_server -b //ip_server//
 -g 41124
```

3. Démarrez `tentacle_client` manuellement :

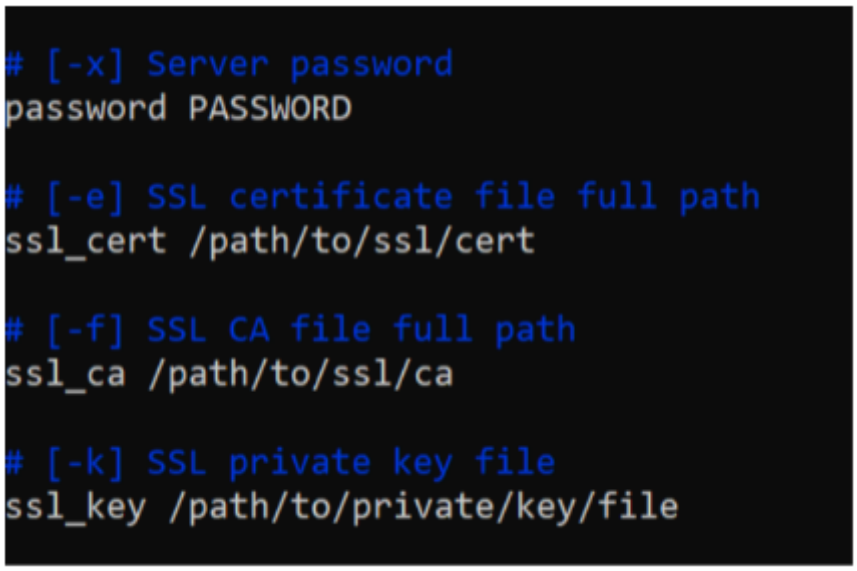
```
sudo -u //user//  tentacle_client \
           -a //ip_proxy/ip_server//  \
           -x password \
           -e tentaclecert.pem \
           -k tentaclekey.pem \
           -v //file//
```

Lorsque vous avez vérifié que la soumission du fichier a réussi, vous pouvez procéder à la configuration permanente du `tentacle_server` et des clients.

Pour configurer le `tentacle_server` avec les options de certificat, vous devez éditer le fichier de configuration du service tentacle_serverd, généralement situé dans `/etc/tentacle/tentacle_server.conf`, de même pour configurer un point intermédiaire pour agir en qualité de mandataire. Pour configurer les agents logiciels afin d'utiliser la communication sécurisée Tentacle, vous devez modifier les fichiers de configuration pandora_agent.conf, généralement situés dans `/etc/pandora/pandora_agent.conf`.

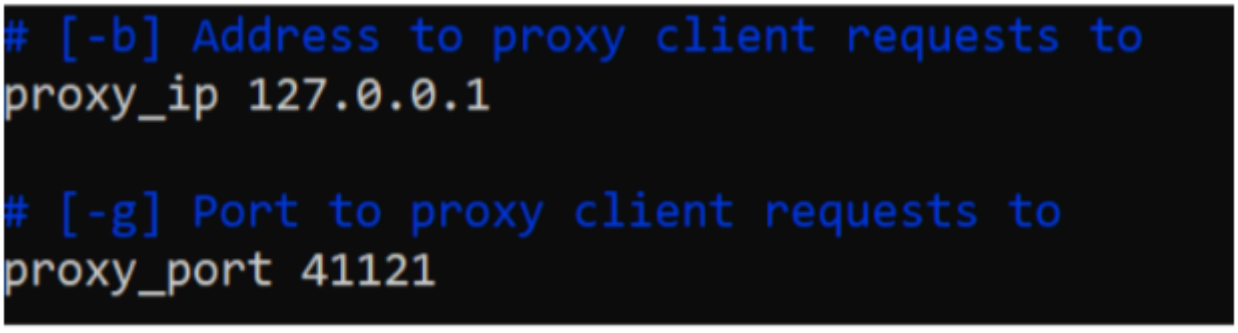
Paramètres permanents :

1. Démarrez le serveur avec SSL. Modifiez le fichier de configuration `/etc/tentacle/tentacle_server.conf` et décommentez et complétez les lignes `password`, `ssl_cert`, `ssl_key`, `ssl_ca` avec les valeurs ou les chemins valides pour votre certificat :



N'oubliez pas qu'à chaque fois que vous apportez des modifications au fichier de configuration de Tentacle, il est nécessaire de redémarrer le service pour que les modifications prennent effet : `/etc/init.d/tentacle_serverd start` .

2. Démarrez le proxy. Comme au point précédent numéro 1, modifiez le fichier de configuration `/etc/tentacle/tentacle_server.conf` de la machine qui fera office de proxy. De même, décommentez et complétez les lignes `proxy_ip` et `proxy_port` avec la configuration valide dans votre environnement :



N'oubliez pas qu'à chaque fois que vous apportez des modifications au fichier de configuration de Tentacle, il est nécessaire de redémarrer le service pour que les modifications prennent effet : `/etc/init.d/tentacle_serverd start` .

3. Démarrez l'agent logiciel avec lles options correspondantes. Modifiez le fichier `pandora_agent.conf`, recherchez la ligne `server_opts` et ajoutez :

```
-x password -e tentacle_client_cert -k tentacle_client_key
```

N'oubliez pas que le jeton `server_ip` doit être défini pour pointer vers l'adresse IP du proxy plutôt que vers celle du serveur principal. Cela ressemblerait à ceci :

```
server_opts -x password -e tentacle_client_cert -k tentacle_client_key
```

Si vous ne souhaitez utiliser aucune des options, comme le mot de passe, n'utilisez simplement pas le paramètre correspondant.

Compression des données dans Tentacle

Version NG 725 ou supérieure.

Tentacle vous permet d'activer la compression des données en transit avec l'option de ligne de commande `-z`, réduisant ainsi la taille des données transférées au détriment de la charge du processeur.

Agent Pandora FMS

Editez le fichier `/etc/pandora/pandora_agent.conf` et ajoutez `-z` à `server_opts` :

```
server_opts -z
```

Serveur satellite

Editez le fichier `/etc/pandora/satellite_server.conf` et ajoutez `-z` à `server_opts` :

```
server_opts -z
```

Éléments du fichier de configuration

Par défaut, le fichier de configuration de Tentacle se trouve dans `/etc/tentacle/tentacle_server.conf`.

N'oubliez pas que chaque fois que vous apportez des modifications au fichier de configuration de Tentacle, vous devez redémarrer le service pour que les modifications

prennent effet :

`/etc/init.d/tentacle_serverd start` .

adresses

```
# [-a] IPv4 address to listen on. Several IP address can be selected separating
it by comma.
addresses 0.0.0.0
```

- Adresse IPv4 où le serveur Tentacle écoutera. Plusieurs adresses IP peuvent être séparées par des virgules.
- Paramètre équivalent en ligne de commande : `-a`.

port

```
# [-p] Port number to listen on
port 41121
```

- Numéro de port où le serveur Tentacle écoutera.
- Paramètre équivalent en ligne de commande : `-p`.

max_connections

```
# [-c] Maximum number of simultaneous connections
max_connections 10
```

- Nombre maximum de connexions simultanées.
- Paramètre équivalent en ligne de commande : `-c`.

daemon

```
# [-d] Run as daemon. 1 true, 0 false
daemon 1
```

- Exécuter en tant que daemon `1`, sinon `0`.
- Paramètre équivalent en ligne de commande : `-d`.

insecure

```

# [-I] Enable insecure mode
insecure 0
```

- Activer le mode non sécurisé `1`, sinon `0` (fait référence à des opérations telles que la liste des fichiers, etc.).
- Paramètre équivalent en ligne de commande : `-I`.

filters

```
# Filters (regexp:dir;regexp:dir...)
filters
.*\.conf:conf;.*\.md5:md5;.*\.zip:collections;.*\.lock:trans;.*\.rcmd:commands
```

- Il vous permet de définir des filtres pour les types de fichiers dans des répertoires spécifiques. Mettez une expression régulière (filtre en tant que tel) séparée par `:` et le répertoire correspondant. Pour ajouter un autre filtre, séparez-le par `;`.
- Paramètre équivalent en ligne de commande : `-i`.

max_size

```
# [-m] Maximum file size allowed by the server in bytes
max_size 2000000
```

- Taille de fichier maximale autorisée (en octets).
- Paramètre équivalent en ligne de commande : `-m`.

overwrite

```
# [-o] Accept files with a repeated name. 1 true, 0 false.
overwrite 0
```

- Il permet l'écrasement si le fichier reçu porte le même nom et existe déjà, désactivé par défaut (`0`), pour l'activer entrez `1`.
- Paramètre équivalent en ligne de commande : `-o`.

quiet

```
# [-q] Do not output error messages.
quiet 0
```

- Évitez d'afficher des messages d'erreur ; activé `1`, désactivé `0`.
- Paramètre équivalent en ligne de commande : `-q`.

retries

```
# [-r] Number of retries for socket read/write operations
retries 3
```

- Nombre de tentatives pour les opérations de lecture et d'écriture.
- Paramètre équivalent en ligne de commande : `-r`.

directory

```
# [-s] Storage directory
```

```
directory /var/spool/pandora/data_in
```

- Il vous permet de définir le répertoire de stockage.
- Paramètre équivalent en ligne de commande : `-s`.

proxy_ip

```
# [-b] IP address to proxy client requests to
proxy_ip 127.0.0.1
```

- Il vous permet de définir l'adresse IP d'un périphérique intermédiaire (client proxy).
- Paramètre équivalent en ligne de commande : `-b`.

proxy_port

```
# [-g] Port number to proxy client requests to
proxy_port 41121
```

- Il vous permet de définir le numéro de port d'un périphérique intermédiaire (client proxy).
- Paramètre équivalent en ligne de commande : `-g`.

timeout

```
# [-t] Timeout for socket read/write operations in seconds
timeout 1
```

- Date d'expirationnes, en secondes, pour les opérations de lecture et d'écriture.
- Paramètre équivalent en ligne de commande : `-t`.

verbose

```
# [-v and -V] Verbose level
#   0: Do not display any informative messages
#   1: Display only important messages [-v]
#   2: Display all messages [-V]
verbose 0
```

- Il définit la quantité d'informations à afficher à des fins de débogage.
    - `-v 0` : Aucun message d'information.
    - `-v 1` ou `-v` : Afficher uniquement les messages importants.
    - `-v 2` ou `-V` : Afficher tous les messages.

log_file

```
# [-l] Log file
log_file /dev/null
```

- Il vous permet d'établir un journal ou un fichier pour enregistrer les événements.
- Paramètre équivalent en ligne de commande : `-l`.

password

```
# [-x] Server password
# password PASSWORD
```

- Définissez le mot de passe du serveur Tentacle.
- Paramètre équivalent en ligne de commande : `-x`.

ssl_cert

```
# [-e] SSL certificate file full path
# ssl_cert /path/to/ssl/cert
```

- Il vus permet de définir le chemin complet du fichier contenant le certificat SSL.
- Paramètre équivalent en ligne de commande : `-e`.

ssl_ca

```
# [-f] SSL CA file full path
# ssl_ca /path/to/ssl/ca
```

- Il permet de définir le chemin complet du fichier qui contient l'Autorité de Certification (CA) du certificat SSL.
- Paramètre équivalent en ligne de commande : `-f`.

ssl_key

```

# [-k] SSL private key file
# ssl_key /path/to/private/key/file
```

- Localisation du fichier avec la clé privée du certificat SSL.
- Paramètre équivalent en ligne de commande : `-k`.

ssl_password

```
# [-w] SSL password. Set to 1 to ask for password by command line
# ssl_password 0
```

- Si le certificat SSL contient un mot de passe, autorisez sa demande (1) sur la ligne de commande.
- Paramètre équivalent en ligne de commande : `-w`.

use_libwrap

```
# [-T] Use libwrap library (Authen::Libwrap perl module). 1 true, 0 false
# use_libwrap 0
```

- Pour le langage Perl, il permet d'utiliser le module `Authen::Libwrap`. Activé `1`, désactivé `0`.
- Paramètre équivalent en ligne de commande : `-T`.

ssl_version

```
# [-z] Restrict to a specific ssl version
# ssl_version TLSv1_3
```

- Il définit une ou plusieurs versions SSL autorisées séparées par deux points (`:`), par exemple : `SSLv3:TLSv1:TLSv1.1:TLSv1.2:TLSv1.3`.
- Chaque version peut être explicitement exclue au moyen d'un point d'exclamation fermant, par exemple pour autoriser SSL2 et SSL3 et exclure l'utilisation des autres versions : `SSLv23:!TLSv1:!TLSv1_1:!SSLv3:!SSLv2`.
- Paramètre équivalent en ligne de commande : `-z`.

ssl_cipher

```
# [-u] Restrict to a specific ssl cipher
#ssl_cipher AES256-SHA
```

- Il définit un ou plusieurs mécanismes de chiffrement autorisés séparés par deux points (`:`), par exemple : `ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384`.
- Les mécanismes de chiffrement peuvent être exclus au moyen d'un point d'exclamation fermant, par exemple `HIGH:!aNULL:!MD5:!3DES` signifie que les chiffrements de haute sécurité sont autorisés, à l'exclusion de ceux qui sont nuls, basés sur MD5 ou 3DES.
- Paramètre équivalent en ligne de commande : `-u`.

Revenir à l'index de la documentation Pandora FMS