Tentacle プロトコル仕様

Tentacle プロトコル仕様

Pandora FMS ドキュメント一覧に戻る

Tentacle について

Tentacle は、以下を意識したクライアント/サーバのファイル転送プロトコルです。

- 安全な設計
- 使いやすい
- 汎用性とクロスプラットフォーム

Tentacle は、SCP / SSH および FTP のようなより複雑なツールの置き換えとして、認証の仕組みも .netrc のようなものや、expect を使ったインタラクティブなログインおよび SSH キーなど、X.509 を使ったものから、簡単にファイルを転送できるツールとして作成されました。

クライアントとサーバは、コマンドラインやシェルスクリプトから呼び出されて動くように設計されています。2008年以降Tentacle は SCP に代わって Pandora FMS のデフォルトのファイル転送方式です。

Tentacle は、Perl および ANSI C (Windows プラットフォームも含む)で実装されています。

ダウンロードおよび詳細情報の確認は、Sourceforge 上の公式のウェブサイト でできます。

目次:

- Tentacle ユーザガイド]
- Tentacle Windows ガイド
- Tentacle プロトコル定義
- OpenSSL 証明書クイックガイド
- tentacle での暗号化通信

Tentacle ユーザガイド

PERL 版のインストール

SourceForge からのインストール

Tentacle サーバをインストールするには、root ユーザ権限が必要です。インストール後は標準ユーザとして実行できます。

SourceForge から tentacle_server-762.tar.gz というファイルを取得します。

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

例 (wget がインストールされている必要があります):

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

Rocky Linux 8 でのインストール

- tar xzvf tentacle_server-762.tar.gz にてダウンロードしたファイルを展開します。
- dnf install perl で perl をインストールします。
- cd tentacle でディレクトリに入ります。
- ./tentacle_server_installer –install にてインストールします。

CentOS 7 でのインストール

- tar xzvf tentacle_server-762.tar.gz にてダウンロードしたファイルを展開します。
- yum install perl perl-IO-Compress zlib で perl をインストールします。
- cd tentacle でディレクトリに入ります。
- ./tentacle_server_installer –install にてインストールします。

SVN からのインストール

この処理は、Apache® Subversion®（svn ）を用いてソースコードをダウンロードしコンパイルすることで行います。そのためには admin または *root* 権限が必要です(このドキュメントでは、# で始まる行です)。

クライアントとサーバの *両方* をインストールするには次のようにします。

```
$ svn co http://svn.code.sf.net/p/tentacled/code/trunk/perl/ tentacle
```

```
$ cd tentacle
$ perl Makefile.PL
$ make
# make install
```

クライアントのみインストールする場合は以下の通りです。

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

サーバのみインストールする場合は以下の通りです。

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

特定の場所にインストールしたい場合は、以下の代わりに

```
$ perl Makefile.PL
```

次のようにします。

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

手動インストール

システムに make が無い場合は、tentacle_client と tentacle_server を手動で適切な場所(例えば /usr/local/bin)にコピーすることもできます。

このとき Perl のバイナリが /usr/bin/perl にない場合は、双方のファイルの 1行目を Perl バイナリが置かれている適切なパスに修正します。例えば Perl がインストールされている場所に ubication を変更します。

```
#!/ubication/perl
```

C言語版のインストール

SVN からのインストール

Tentacle クライアントをインストールするには、次のようにします。

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

configure の出力エラー、ヘッダーファイルの不足などを確認します。

OpenSSL 開発ライブラリがあるとデフォルトで OpenSSL 対応が有効化されますが、無効化したい場合は、以下を

```
$ ./configure
```

次のようにします。

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

Tentacle 利用例

To see the available options, execute -h parameter, both in the client and server version:

存在する全オプションを表示するには、クライアント、サーバ共に -h パラメータを付けて実行します。

```
$ 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 opertions (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.)
```

すべてのオプションで事前定義された値もヘルプに表示されます。

以降の例では、サーバアドレスは 192.168.1.1 で、クライアントの秘密鍵はパスワードで保護されていません。

- 最大 1MB で /tmp にあるファイルの単純な転送:

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

- 上書きモード有効にして 65000 番ポートでの単純な転送:

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

- パスワード認証による単純な転送:

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

- クライアント証明書無しの暗号化転送:

```
$ 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
```

- クライアント証明書有りの暗号化転送:

```
$ 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
```

- クライアント証明書有りでパスワード認証を付けた暗号化転送(パラメータをわかりやすくするために、を利用して改行していることに注意してください)

```
$ 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
```

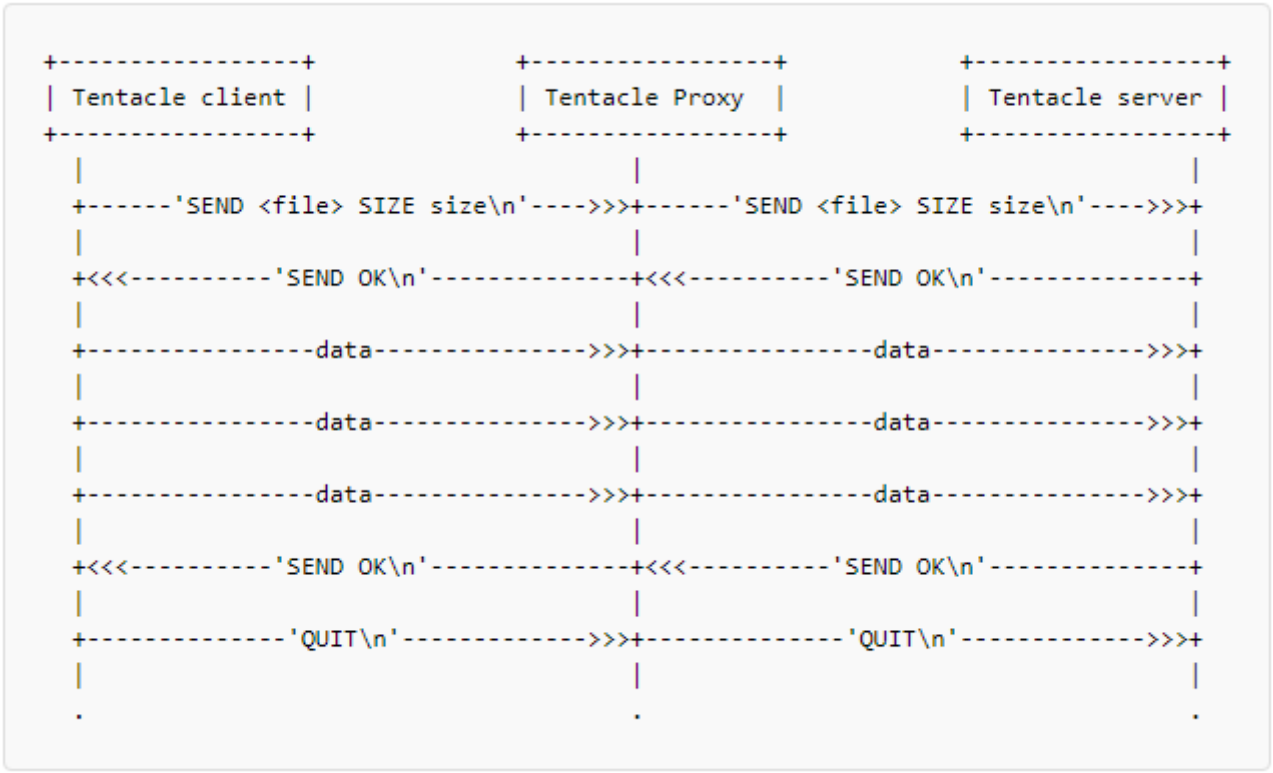
Tentacle サーバでは、プレーンテキストファイルで設定できます。 すべてのコマンドラインオプションは設定ファイルでも指定できます。ファイルとコマンドラインの両方で同じ設定オプションが指定されている場合、後から設定された値が優先されます。設定ファイルへのフルパスは、オプション -F で指定します。

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

Tentacle プロキシ

Tentacle サーバは、多くの tentacle クライアントが直接通信できない tentacle サーバとの通信を中継するプロキシとしても動かすことができます。

以下は、どのように tentacle プロキシが動作するかを示した図です。



プロキシは情報を持たず、データをクライアントから tentacle サーバへ転送するだけです。例えば tentacle サーバをプロキシモードで起動するには、次のようなパラメータを利用します。

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

これらのパラメータは、クライアントから直接アクセスできない tentacle サーバの IP アドレス (-b) および ポート (-g) です。通常のパラメータも一緒に指定できます。

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

Tentacle プロキシも、認証 および 暗号化 に対応しています。

Windows 版 PERL のインストール

この簡易ガイドは、MS Windows® で Tentacle クライアントとサーバを設定および実行するためのものです。

Perl 版のインストール

Perl 環境のインストール

ActiveState® の https://www.activestate.com/products/downloads/ から ActivePerl 5.8 をダウンロー

ドし、デフォルトオプションでインストーラを実行します。

モジュール IO-Socket-SSL のインストール

OpenSSL を以下からダウンロードします。

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

以下の perl モジュールをダウンロードします。

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

.ppd ファイルがあるディレクトリでコマンドプロンプトから以下を実行します。

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

Tentacle クライアントとサーバの実行

実行は、Unix/Linux システムと似ており、最初に perl コマンド、その後に実行プログラムを指定します。例:

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

Tentacle プロトコル定義

Tentacle プロトコル自体は非常に簡単です。いくつかの重要な特徴は次の通りです。

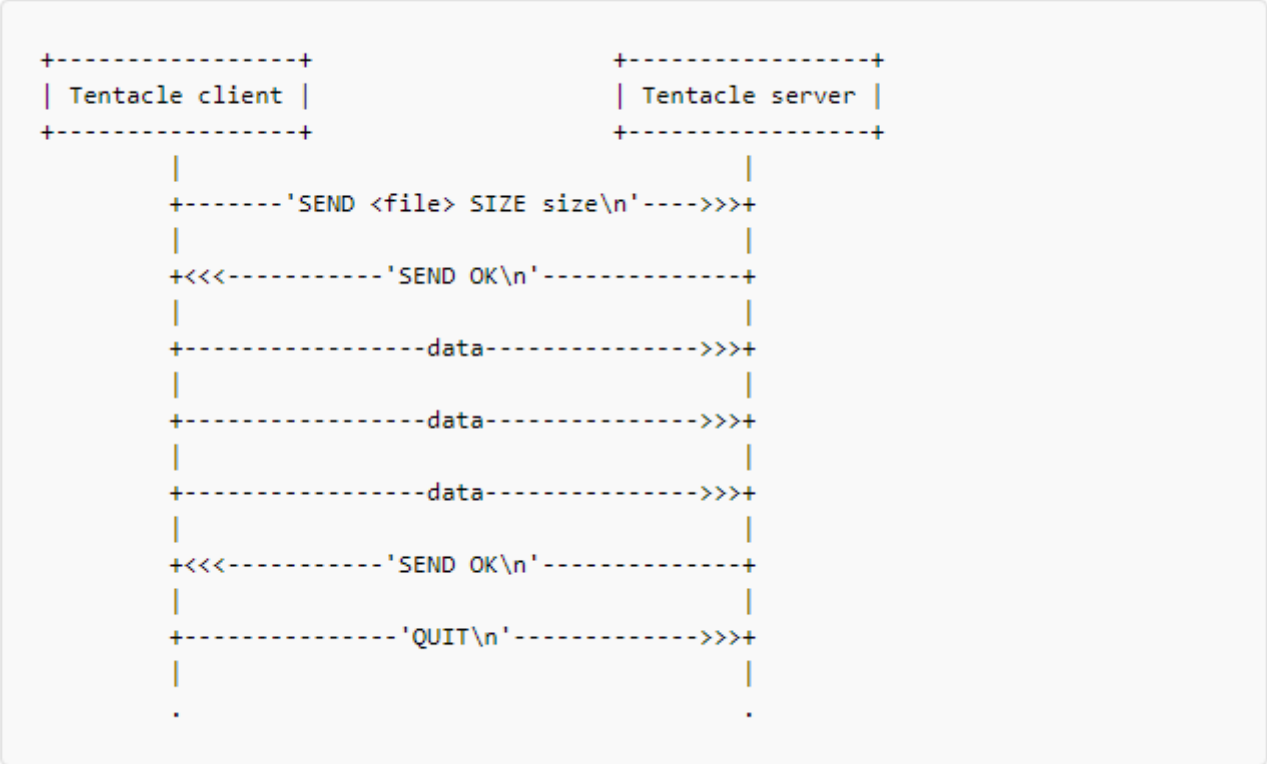
- 通信は、常にクライアント側から開始します。
- コマンドは、常に改行コードで終了します。
- 次の文字はファイル名には利用できません:

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

ASCII 文字でのシーケンス図でユースケースを説明します。コマンドはシングルクォートでくくります。

ファイル送信

まず最初に、正しくファイル転送できた場合を示します。



一つのセッションで複数のファイル転送ができるように、一つのファイル転送が完了したあと QUIT の前に新たな SEND コマンドを送ることができます。

もし、サーバがファイルの受け取りを拒否する場合は、クライアントにエラーメッセージが返されます。セキュリティ上の理由により詳細は示しませんが、ファイルが以下の場合に発生します。

- 不正なファイル名やパスが指定された。
- 空もしくはサーバが受け取れる最大サイズを超過した。
- すでにサーバ上にファイルがあり、ファイルの上書きが許可されていない。

ファイル受信

単一のファイルをサーバから要求できます。

クライアントは、サーバがサイズを報告してきた後にファイル受け取り拒否が可能です。

SEND と同様に、一つのファイルの受け取りが完了した後に、QUIT の前に(クライアントがファイル受信を拒否した場合でも)新たな RECV コマンドを送信することができます。サーバがファイル送信を拒否した場合は、エラーメッセージが送られます。ファイルが次のような場合に発生します。

- 不正なファイル名やパスが指定された。
- サーバにファイルが存在しない。

パスワード認証

サーバがパスワードを要求した場合、クライアントはコマンド送信前に認証を行う必要があります。

パスワードを隠蔽するためにパスワードの 2段階の md5 が送信されます。しかし、非暗号化接続では、それ以上のセキュリティは ない ことに注意してください。セキュアなファイル転送が必要な場合には、常に SSL を有効化してください。

エラー処理

エラー状態になると、サーバは何らかの説明を行うことなく接続を切ります。不正なコマンド、不正なパスワード、送信すると通知したものよりも多くのデータ送信などの場合です。

デフォルトでは Tentacle のログ出力先は /dev/null になっています。

OpenSSL 証明書クイックガイド

これは OpenSSL 証明書を使う場合のクイックガイドです。http://www.openssl.org/docs/ もあわせて参照してください。

証明書の作成

環境の準備をします。

```
$ mkdir demoCA
$ mkdir demoCA/newcerts
$ mkdir demoCA/private
```

セキュリティ上の理由から、作成したフォルダに必要なユーザの書き込みと読み取りのアクセス権限を設定することを忘れないでください。

次の手順では、自己署名 CA 証明書を作成し、作成したディレクトリに移動します。

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

```
$ mv cacert.pem demoCA/
```

証明書の要求されたフィールドに入力し、後で同じ情報を使うため、それを記録しておいてください。 次に、証明書署名要求を作成します。

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

証明書署名要求に署名し、連続したシリアルを設定して制御および監査システムに設定します。

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

ランダムファイル に不便な点がある場合は、*root* ユーザで削除できます: `sudo rm ~/.rnd` そうすれば、書き込みおよび読み取り権限のあるユーザで再度作成できます。 あなた が、上記の *root* キーの唯一の責任者です。

自己証明書の作成

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

RSA 秘密鍵の作成

これは、クライアント側で Tentacle 利用の際にパスワードの入力をする必要が無く便利です。

鍵生成:

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

上記の手順で `-keyout` を `-key` に置き換えます。

証明書の他のフォーマットへのエクスポート

一部のオペレーティングシステム(Ubuntu® や Windows® など)では PEM ではなく DER 形式の証明書が必要になる場合があります。 その場合は、生成された PEM を介して上記の形式の証明書を取得できます。

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

セキュリティオプション付での Tentacle 設定ガイド

このガイドでは、安全な通信を確保するために エージェントソフトウェア と Tentacle サーバ の両方を設定する方法を段階的に説明します。

まず、デバイスから手動テストを実行して、設定、パラメータ、および証明書が正しいことを確認することをお勧めします。

次に、それに応じた次の設定ファイルを永続的に設定します。

Tentacle サーバ

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

Unix/Linux ソフトウエアエージェント

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

MS Windows® ソフトウエアエージェント

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

サテライトサーバ

```
ect/pandora/satellite_server.conf
```

Tentacle プロキシサーバ

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

変更後は関連するサービスを再起動することを忘れないようにしてください Unix/Linux の場合は `/etc/init.d/tentacle_serverd` の TENTACLE_EXT_OPTS オプションを使うこともできます。(その他のオプションについては こちらのリンク を確認してください。)

通信の暗号化

Tentacleサーバーとソフトウェアエージェントの両方が、証明書とパスワードを使用した安全な通信を使用できます。両方の間で直接通信するか Tentacle プロキシサーバを介して通信します。

常に 証明書がある場所は絶対パスで指定する必要があります。例: `/etc/ssl/tentaclecert.pem`

Tentacle の暗号化オプションを利用するには、システムに `perl(IO::Socket::SSL)` パッケージがインストールされていることを確認してください。

前の章では、さまざまな組み合わせについて詳しく説明しました。この章では、パスワードオプション、Tentacle プロキシサーバ、および設定を追加するための TENTACLE_EXT_OPTS パラメータについて説明します。また、前述の 、証明書名とキーを確認してください。説明の目的で簡略化した表現をしています。

パスワード認証での単純な転送:

パスワード認証のためのサーバのパラメータ:

```
-x password
```

パスワード認証のためのクライアントのパラメータ(TENTACLE_EXT_OPTS):

```
-x password
```

クライアント証明無しでの暗号化転送:

サーバのパラメータ:

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

クライアント証明有りでの暗号化転送:

サーバのパラメータ:

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

クライアントのパラメータ (TENTACLE_EXT_OPTS):

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

クライアント証明書およびパスワード認証での暗号化転送:

サーバのパラメータ:

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

クライアントのパラメータ (TENTACLE_EXT_OPTS):

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

証明書を用いて Tentacle サーバが任意のクライアントからの暗号化接続を受け付ける設定

この設定では、Tentacle サーバの設定で暗号化に使用する証明書とキーを入力するだけです。

手動で サーバ を起動する際に、-e および -k パラメータを指定します。

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

クライアント を手動で起動する際には、-c パラメータを指定します。

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

手動実行で正しく動作することが確認できたら、それで動作するように設定を行います。

- Tentacle サーバ の場合:

```
ssl_cert tentacle_cert
ssl_key tentacle_key
```

- ソフトウエアエージェント の場合:

```
server_opts -c
```

- サテライトサーバ の場合:

```
server_opts -c
```

Tentacle サーバが特定の CA の署名がされたクライアントの検証をして接続を受け付ける設定

この設定では□Tentacleサーバの暗号化設定とクライアントに使用される証明書とキーを指定します。

サーバ を手動で起動する際に、-e および -k パラメータを指定します。

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

クライアント を手動で起動する際は、-e および -f パラメータを指定します。

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

手動実行で正しく動作することが確認できたら、それで動作するように設定を行います。

- Tentacle サーバ の場合:

```
 ssl_cert tentacle_cert
 ssl_key tentacle_key
```

- Pandora FMS ソフトウエアエージェント の場合:

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

- Pandora FMS サテライトサーバ の場合:

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

特定の CA の署名をされた Tentacle サーバへクライアントが検証し接続する設定

この設定では、Tentacle サーバとクライアントの設定で暗号化に使用される証明書とキーを設定します。

サーバ を手動で起動する際に、-e、-k、-f パラメータを指定します。

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

ライアント を手動で起動する際は、-e および -k パラメータを指定します。(改行のため \ を使っています)

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

手動実行で正しく動作することが確認できたら、それで動作するように設定を行います。

- Tentacle サーバ の場合:

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

- Pandora FMS ソフトウエアエージェント の場合:

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

- Pandora FMS サテライトサーバ の場合:

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

Tentacle サーバおよびクライアントの双方が特定の CA の署名を検証する接続設定

この設定では、Tentacleサーバとクライアントの双方で暗号化に使用される証明書と鍵を設定します。

サーバを手動で起動する際に、-e、-k、-f パラメータを指定します。

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

クライアントを手動で起動する際は、-e -k -f パラメータを指定します。

```
 # 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
```

手動実行で正しく動作することが確認できたら、それで動作するように設定を行います。

- Tentacle サーバ の場合:

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

- Pandora FMS ソフトウエアエージェント の場合:

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

- Pandora FMS サテライトサーバ の場合:

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

Tentacle 暗号化設定

Tentacle サーバとソフトウェアエージェントの両方が、証明書とパスワードを介して、直接または Tentacle プロキシを介して安全な通信を使用できます。

常に、証明書の絶対パスをパラメーターに指定します。例えば、 `/etc/ssl/tentaclecert.pem` です。

</WRAP>

Tentacle 暗号化オプションを使用するには、システムにパッケージ `perl(IO::Socket::SSL)` がインストールされていることを確認してください。

前の章では、さまざまな組み合わせについて詳しく説明しました。この章では、パスワードオプション Tentacle プロキシサーバ、および設定を追加するための TENTACLE_EXT_OPTS パラメータについて説明します。また、前述の 、証明書名とキーを確認してください。説明の目的で簡略化した表現をしています。

パスワード認証での単純な転送:

パスワード認証のためのサーバのパラメータ:

```
-x password
```

パスワード認証のためのクライアントのパラメータ(TENTACLE_EXT_OPTS):

```
-x password
```

クライアント証明無**しでの**暗号化転送:

サーバのパラメータ:

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

クライアント証明有**りでの**暗号化転送:

サーバのパラメータ:

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

クライアントのパラメータ(TENTACLE_EXT_OPTS):

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

クライアント証明書**およびパスワード**認証**での**暗号化転送:

サーバのパラメータ:

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

クライアントのパラメータ(TENTACLE_EXT_OPTS):

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

Tentacle プロキシを使った暗号化設定例

Tentacle **プロキシサーバを**利用**した**暗号化通信**のためのソフトウェアエージェントと** Tentacle **サーバ**両方**の**設定方法**を**段階的**に**説明**します。** 前**の**章**で**述**べた**証明書**の**名前**と**鍵**を**確認**してください。** パラメータ**に**関**しても**確認**してください。**

手動**テスト**:

1. tentacle_server **を**手動起動**します**:

```
 sudo -u user tentacle_server \
           -x password \
           -e tentacle_cert \
           -k tentacle_key \
```

```
            -f ca_cert -s /tmp -v
```

2. プロキシを手動起動します:

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

3. tentacle_client を手動で起動します:

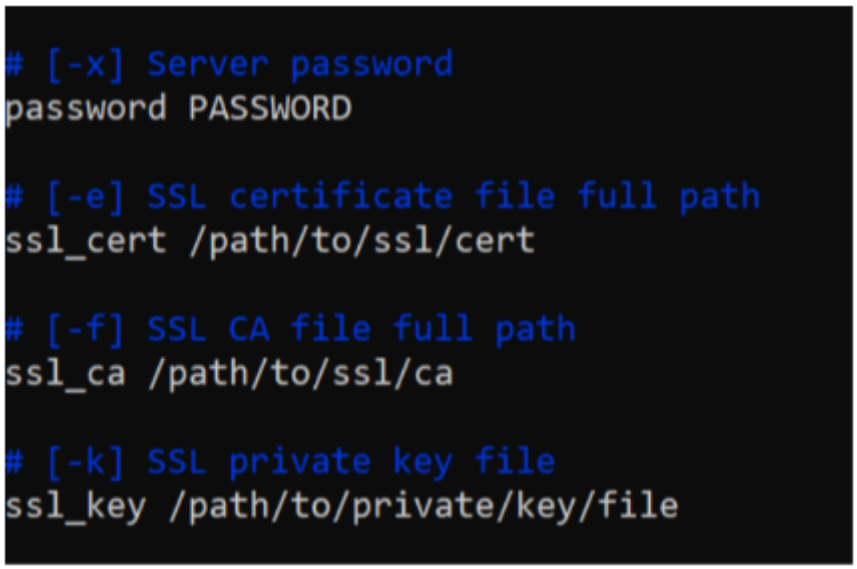
```
  sudo -u user tentacle_client \
             -a ip_proxy/ip_server \
             -x password \
             -e tentaclecert.pem \
             -k tentaclekey.pem \
             -v file
```

ファイルが正常に送信されたことを確認したら、tentacle_server とクライアントの永続的な設定に進みます。

暗号化オプション付きで tentacle_server を設定するには、通常 `/etc/init.d/tentacle_serverd` にあるサービス tentacle_serverd の起動スクリプトを編集します。 中間点は、プロキシとして機能するように設定する必要があります。Tentacle 暗号化通信を使用するようにソフトウェアエージェントを設定するには、通常は `/etc/pandora/pandora_agent.conf` にある設定ファイル pandora_agent.conf を編集します。

永続的な設定

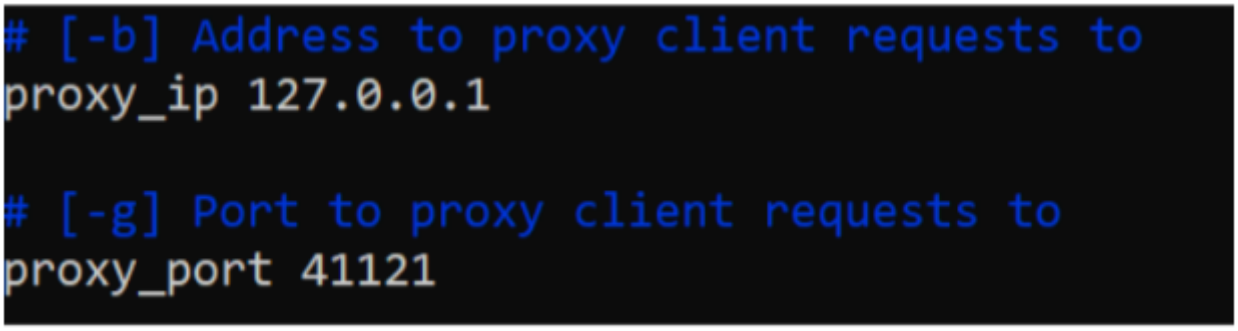
1. SSL つきでサーバを開始します。起動スクリプト `/etc/init.d/tentacle_serverd` を編集します。TENTACLE_EXT_OPTS の行を探し、以下を追加します。



Tentacle 設定ファイルに変更を加えた際は、それを有効にするためにサービスを再起動する必要があることに注意してください：

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

2. プロキシを開始します。1. と同様に、プロキシとして動作させるマシンで起動スクリプト /etc/init.d/tentacle_serverd を編集します。同様に TENTACLE_EXT_OPTS の行を探し、以下を追加します。



Tentacle 設定ファイルに変更を加えた際は、それを有効にするためにサービスを再起動する必要があることに注意してください：
/etc/init.d/tentacle_serverd start

3. 対応するオプションをつけてソフトウエアエージェントを開始します。pandora_agent.conf を編集し、server_opts という行を探し、以下を追加します。

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

server_ip で指定するアドレスは、監視サーバではなくプロキシサーバの IP にすることに注意してください。server_ops の行全体は次のようになります。

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

パスワードなどの一部のオプションを使用したくない場合は、対応するパラメーターを使用しないでください。

Tentacle データ圧縮

バージョン NG 725 以上

Tentacle では、コマンドラインのオプション -z を使用してデータ圧縮を有効にすると、CPU 負荷は上がりますが、転送されるデータのサイズを削減できます。

Pandora FMS エージェント

/etc/pandora/pandora_agent.conf を編集し、server_opts に -z を追加します。

```
server_opts -z
```

サテライトサーバ

/etc/pandora/satellite_server.conf を編集し、server_opts に -z を追加します。

```
server_opts -z
```

設定ファイル要素

デフォルトでは tentacle の設定ファイルは、/etc/tentacle/tentacle_server.conf にあります。

Tentacle の設定ファイルに変更を加えるたびに、変更を有効にするためにはサービスを再起動する必要があることに注意してください。
/etc/init.d/tentacle_serverd start

addresses

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

- tentacle サーバが待ち受ける IPv4 アドレスです。複数の IP アドレスをカンマ区切りで設定できます。
- 同等のコマンドラインパラメータ: -a

port

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

- tentacle サーバが待ち受けるポート番号です。
- 同等のコマンドラインパラメータ: -p

max_connections

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

```
max_connections 10
```

- 同時接続の最大数です。
- 同等のコマンドラインパラメータ: `-c`

daemon

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

- デーモンとして実行する場合は、1、そうでなければ 0 です。
- 同等のコマンドラインパラメータ: `-d`

insecure

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

- 安全でないモードを有効化 1、無効化 0 (ファイルの一覧表示などの操作を指します)
- 同等のコマンドラインパラメータ: `-I`

filters

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

- 特定のディレクトリ内のファイルタイプごとにフィルタを設定できます。 : で区切られた正規表現(フィルター自体)と対応するディレクトリを追加します。 別のフィルターを追加するには、; で区切ります。
- 同等のコマンドラインパラメータ: `-i`

max_size

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

- 最大のファイルサイズ(バイト単位)です。
- 同等のコマンドラインパラメータ: `-m`

overwrite

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

- 受信したファイルが同じ名前ですでに存在する場合に上書きするかどうかです。デフォルトで無効(0)、有効化するには、1 を設定します。
- 同等のコマンドラインパラメータ: `-o`

quiet

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

- **エラーメッセージの**表示を抑制します。有効化 1、無効化 0
- 同等のコマンドラインオプション: -q

retries

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

- 読み書き処理のリトライ回数です。
- 同等のコマンドラインパラメータ: -r

directory

```
# [-s] Storage directory
directory /var/spool/pandora/data_in
```

- 保存先ディレクトリの設定です。
- 同等のコマンドラインパラメータ: -s

proxy_ip

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

- 中間のデバイス(*プロキシクライアント*)の IP アドレスを設定できます。
- 同等のコマンドラインパラメータ: -b

proxy_port

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

- 中間デバイス(*プロキシクライアント*)のポート番号を設定できます。
- 同等のコマンドラインパラメータ: -g

timeout

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

- 読み書き処理のタイムアウトを秒単位で指定します。
- 同等のコマンドラインパラメータ: -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
```

- デバッグ目的で表示する情報量を設定します。
  - -v 0 : 情報を表示しません。
  - -v 1 または -v: 重要なメッセージのみ表示します。
  - -v 2 または -V: すべてのメッセージを表示します。

log_file

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

- ログファイルを設定します。
- 同等のコマンドラインパラメータ: -l

password

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

- Tentacle サーバのパスワードを設定します。
- 同等のコマンドラインパラメータ: -x

ssl_cert

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

- SSL 証明書ファイルのフルパスを設定できます。
- 同等のコマンドラインパラメータ: -e

ssl_ca

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

- SSL 証明書 の認証局(CA)ファイルのフルパスを設定できます。
- 同等のコマンドラインパラメータ: -f

ssl_key

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

- SSL 証明書の秘密鍵ファイルの場所です。
- 同等のコマンドラインパラメータ: `-k`

ssl_password

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

- SSL 証明書にパスワードが含まれる場合、コマンドラインからそれを求める(1)ことができます。
- 同等のコマンドラインパラメータ: `-w`

use_libwrap

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

- perl の `Authen::Libwrap` モジュールを使うことができます。有効化 `1`、無効化 `0`
- 同等のコマンドラインパラメータ: `-T`

Pandora FMS ドキュメント一覧に戻る