Tentacle protocol specifications

Tentacle protocol specifications

¿What is Tentacle?

Tentacle, a client/server file transfer tool, is:

- Secure by design.
- Easy to use and integrate with other tools.
- Versatile, flexible and cross-platform.

Tentacle has been created to replace more complex tools such as SCP/SSH and FTP for simple file transfers and to move away from insecure authentication mechanisms such as .netrc , as well as automated interactive logins with expect and the SSH key mechanism, to an authentication based on the X.509 standard, using certificates.

The client and server are designed to be run from the command line, or called from a shell script. Tentacle is since 2008 the default transfer method for Pandora FMS, replacing SCP.

Tentacle is implemented in Perl and ANSI C (both platforms included in MS Windows®).

You may download it and learn more about it at the Official website of the project at SourceForge.

GNU Linux Tentacle User's Guide

Perl version installation

Installing from Source Forge Net

To install Tentacle server you must have rights equivalent to root user, after having installed it you will be able to run it as a standard user.

Get the file `tentacle_server-762.tar.gz` from Source Forge Net:

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

For example (you must have wget installed):

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

Installation on Rocky Linux 8

- Unzip the downloaded file with `tar xzvf tentacle_server-762.tar.gz` .
- Install the Perl language with `dnf install perl` .
- Enter the directory with `cd tentacle` .
- Install with `./tentacle_server_installer –install` .

Installation on CentOS 7

- Unzip the downloaded file with `tar xzvf tentacle_server-762.tar.gz` .
- Install the Perl language with `yum install perl perl-IO-Compress zlib` .
- Enter the directory with `cd tentacle` .
- Install with `./tentacle_server_installer –install` .

Installation from SVN

The process consists of downloading the source code using Apache® Subversion® (svn) and compiling it. To do this you will need to have administrator or root rights (in this documentation it is the lines starting with the numeral character #). You are solely responsible for this key.

To install both the client and server versions, run:

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

To install only the client part run:

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

To install only the server part of the server run:

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

If you want to install to a specific directory, replace:

```
$ perl Makefile.PL
```

by:

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

Manual installation

If make is not available on your system, you may install it manually by copying the `tentacle_client` and `tentacle_server` files to the appropriate directory (e.g. `/usr/local/bin`).

In this case, if the Perl binary is not located at `/usr/bin/perl`, edit both Tentacle files and change the first line so that it points to the correct path where your Perl binary is located. So, for example, replace `location` with the location of Perl on the system to install:

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

Installing the C version

Installing from SVN

Taking into account the preamble to the installation of the previous section, to install the Tentacle client run:

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

Make sure that the output of the configure command does not generate any errors, incomplete header dependencies, etc..

To disable OpenSSL support, which is enabled by default, replace:

```
$ ./configure
```

by:

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

Examples of Tentacle usage

To display the available options run with the -h parameter, both in the client and server versions:

```
$ 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.)
```

The default values for all options will also be shown in the Help.

For all the examples shown below, the server is located at address `192.168.1.1` and the client's private key is not password protected..

- Simple transfer of a file limited to a maximum size of 1 megabyte and deposited on `/tmp`:

```

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

- Single transfer on port 65000 with overwrite mode enabled:

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

- Simple transfer with password-based authentication:

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

- Secure transfer, without client certificate:

```
$ 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
```

- Secure transfer with customer certificate:

```
$ 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
```

- Secure transfer with client certificate and additional password authentication (note the use of the connector to facilitate the typing of various parameters):

```
$ 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
```

The Tentacle server allows its configuration through a plain text file. All command line options are available through this file. If the same configuration option is specified in the file and on the command line, the value specified in the command line will take precedence. The full path to the configuration file is indicated by option `-F`.

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

Tentacle Proxy

The Tentacle server can work as a proxy communicating many Tentacle clients to an inaccessible Tentacle server..

The following diagram shows how Tentacle's proxy server works:

```
+-----------------+              +-----------------+            +-----------------+
| Tentacle client |              | Tentacle Proxy  |            | Tentacle server |
+-----------------+              +-----------------+            +-----------------+
  |                                      |                                      |
  +------'SEND <file> SIZE size\n'---->>>+------'SEND <file> SIZE size\n'---->>>+
  |                                      |                                      |
  +<<<----------'SEND OK\n'--------------+<<<----------'SEND OK\n'--------------+
  |                                      |                                      |
  +----------------data--------------->>>+----------------data--------------->>>+
  |                                      |                                      |
  +----------------data--------------->>>+----------------data--------------->>>+
  |                                      |                                      |
  +----------------data--------------->>>+----------------data--------------->>>+
  |                                      |                                      |
  +<<<----------'SEND OK\n'--------------+<<<----------'SEND OK\n'--------------+
  |                                      |                                      |
  +--------------'QUIT\n'------------->>>+--------------'QUIT\n'------------->>>+
  |                                      |                                      |
  .                                      .                                      .
```

The proxy does not have any information, but only sends the information from the clients to the Tentacle server. For example, to launch the Tentacle server in proxy mode use the following parameters:

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

These parameters are IP address (`-b`) and port (`-g`) *of the inaccessible tentacle server*. Add, in addition, the normal parameters in a single line:

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

The Tentacle protocol in proxy mode also supports authentication and encryption parameters.

Tentacle Guide on MS Windows

Configure and run the Tentacle client and server on MS Windows®..

Perl version installation

Installation of the Perl environment

Using ActiveState® download ActivePerl 5.8 using the following link and run the installer with the default options:

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

Installation of the IO-Socket-SSL module

Download and install OpenSSL from:

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

Download the following Perl modules:

- 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

Execute from command line in the directory where the files are located .ppd:

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

Running the Tentacle client and server

the execution is similar to that of Unix/Linux systems, you only need to enter the Perl command first, followed by the full syntax, e.g.:

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

Tentacle protocol definition

The Tentacle protocol itself is very simple and straightforward. Some important design features are:

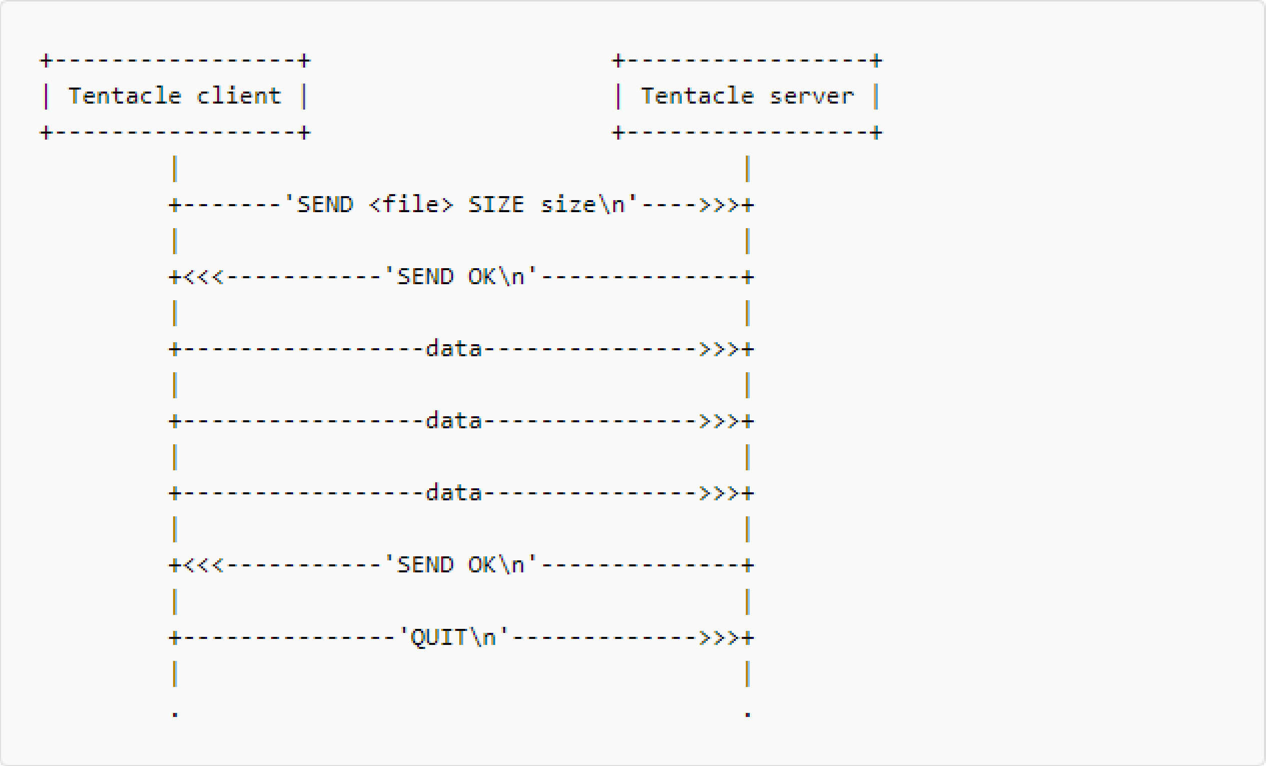
- Communication is always established by the customer.
- Commands always end with an end-of-line character.
- The following characters cannot be part of a file name:

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

ASCII sequence diagrams will be used to illustrate the possible cases. Commands are shown in single quotes.

Send file(s)

A successful file transfer is displayed first



```
+-----------------+                     +-----------------+
| Tentacle client |                     | Tentacle server |
+-----------------+                     +-----------------+
         |                                       |
         +-------'SEND <file> SIZE size\n'---->>>+
         |                                       |
         +<<<-----------'SEND OK\n'--------------+
         |                                       |
         +-----------------data--------------->>>+
         |                                       |
         +-----------------data--------------->>>+
         |                                       |
         +-----------------data--------------->>>+
         |                                       |
         +<<<-----------'SEND OK\n'--------------+
         |                                       |
         +---------------'QUIT\n'------------->>>+
         |                                       |
         .                                       .
```

To allow multiple file transfers within the same session, a new “SEND” command must be sent, after a successful transfer, and before a “QUIT” command.

If the server rejects a file, a generic error message is sent back to the client. For security reasons, no details are shown as to why the command fails. This occurs when:

- The file has an invalid file name, or a path is specified.
- It is empty or exceeds the maximum size specified by the server.
- It already exists on the server and file overwriting is not enabled.

File receipt

Tentacle also supports file requests from the client.

The client has the opportunity to reject the file after the server informs about its size.

As with the “SEND” command, a new “RECV” command can be sent after a successful transfer (even if the file was rejected by the client) and always before the “QUIT” command. A generic error will be sent if the server refuses to send the file. The latter may take place when:

- You have an invalid file name, or a path has been specified.
- Does not exist on the server.

```
+-----------------+                     +-----------------+
| Tentacle client |                     | Tentacle server |
+-----------------+                     +-----------------+
         |                                       |
         +------------'RECV <file>\n'--------->>>+
         |                                       |
         +<<<-----------'RECV ERR\n'-------------+
         |                                       |
         .                                       .
```

Password authentication

If the server requires a password, the client must authenticate before sending any other commands.

```
+-----------------+                     +-----------------+
| Tentacle client |                     | Tentacle server |
+-----------------+                     +-----------------+
         |                                       |
         +----------'PASS pwd_digest\n'------->>>+
         |                                       |
         +<<<-----------'PASS OK\n'--------------+
         |                                       |
         .                                       .
```

A double MD5 of the password will be sent to obfuscate. If you are working over an unencrypted connection, this does NOT implement or add any security. *If you need security use* SSL encrypted connections .

Error handling

In case of any error, the server will close the connection without explanation. This may be due to an incorrect command, an erroneous password, more data sent than was supposed to be sent, or any other reason that causes the server to operate outside of what is established or considered “normal”.

```
+-----------------+                     +-----------------+
| Tentacle client |                     | Tentacle server |
+-----------------+                     +-----------------+
         |                                       |
         +-------------'!@#$%&/()=?¿'--------->>>+
         |                                       |
         .                                       .
```

```
+-----------------+                     +-----------------+
| Tentacle client |                     | Tentacle server |
+-----------------+                     +-----------------+
         |                                       |
         +---------'PASS bad_pwd_digest'------>>>+
         |                                       |
         .                                       .
```

By default, the Tentacle log is set to `/dev/null`.

Quick guide to OpenSSL certificates

This is a quick start guide on OpenSSL certificates for use with Tentacle or other applications. For more information you may check the official website of the OpenSSL project:

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

Certificate creation

Environment preparation:

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

Remember to set, for security purposes, writing and
reading permissions of the different users in your system
on the newly created folders.

The next step is to make a self-signed CA certificate and move it into the created directories:

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

Fill in the requested fields for the certificate and remember them well because they will be needed again later, exactly the same. Now create a certificate request:

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

Sign the certificate request, also establishing a consecutive series of certificates as a control and auditing mechanism:

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

Please note that if you submit the random seed file If there is any inconvenience, you may delete it with root user rights: `sudo rm ~/.rnd`. That way it can be created again with its own writing and reading rights. You are solely responsible for this root key.

Create a self-signed certificate

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

Generate an RSA private key

This is very useful to avoid having to enter a password on the client side using Tentacle.

Generate key:

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

And replace `-keyout` with `-key` in the previous sections.

Export certificate to another format

Certificates may be required in DER format instead of PEM for some operating systems (such as Ubuntu® or Windows®). If that is the case, the certificate can be obtained in that format from the generated PEM.:

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

Secure communication configuration with Tentacle

It explains step by step how to configure both Software Agents and Tentacle servers for secure communication.

First of all, it is highly recommended to test them manually from the terminals to ensure that the configuration, parameters and certificates are correct.

A permanent configuration can then be made in the respective configuration files:

Tentacle Servers

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

Software Agents on Unix/Linux

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

MS Windows® Software Agents

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

Satellite Servers

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

Tentacle Proxy Servers

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

Remember to restart the corresponding services after any modification. In the case of Unix/Linux you may also use the option `TENTACLE_EXT_OPTS` located at `/etc/init.d/tentacle_serverd` (see the rest of the options for this daemon in this link).

Communication encryption

In order to encrypt the communication between the clients and the Tentacle server, it will be necessary to have SSL certificates and keys. In this guide we will see all the possible configuration options, so the certificates can be either self-signed as signed by a valid CA.

To avoid misinterpretations in this article, certificates and keys on each side are identified with the following names:

- `ca_cert`: Certificate of the CA used for signing the certificates.
- `tentacle_key`: Generated key for Tentacle server.
- `tentacle_cert`: Certificate generated for Tentacle server.
- `tentacle_client_key`: Key generated for Tentacle customer.
- `tentacle_client_cert`: Certificate generated for Tentacle client.

It is ALWAYS necessary to indicate in the parameters the absolute paths where certificates are located, e.g. `/etc/ssl/tentaclecert.pem`

To use Tentacle's safe options, please check that the `perl(IO::Socket::SSL)` package is *installed on your system*.

Certificate configuration on Tentacle server accepting any certificate on the client

For this configuration you must specify the certificate and key used for encryption in the Tentacle server configuration..

Run manually on the server with parameters -e and -k:

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

Execute manually on the client with the parameter -c.:

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

If this manual execution works correctly, you may configure it permanently in the corresponding file:

- For a Tentacle server.:

```
ssl_cert tentacle_cert
ssl_key tentacle_key
```

- For a Software Agent.:

```
server_opts -c
```

- For a SATELLITE server:

```
server_opts -c
```

Certificate configuration on the Tentacle server and on the client by verifying the certificate with a specific CA on the client.

For this configuration specify the certificate and key used for encryption in the Tentacle server configuration and the certificates used for encryption on the clients..

Run manually on the server with parameters -e and -k:

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

Execute manually on the client with parameter -e and -f:

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

If this manual execution works correctly, you may configure it permanently in the corresponding file:

- For a Tentacle server.:

```
ssl_cert tentacle_cert
ssl_key tentacle_key
```

- For a Software Agent.:

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

- For a SATELLITE server:

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

Certificate configuration on the Tentacle server and on the client by verifying the certificate with a specific CA on the server.

For this configuration specify the certificates and keys used for encryption in the configuration of the Tentacle server and clients..

Run manually on the server with the parameters `-e`, `-k` and `-f`:

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

Manually run on the client with the `-e` and `-k` parameters (note the use of the \ line connector):

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

If this manual execution works correctly, you may configure it permanently in the corresponding file:

- For a Tentacle server.:

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

- For a Software Agent:

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

- For a SATELLITE server:

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

Configuration of certificates on the Tentacle server and on the client by verifying the certificate with a specific CA in both

For this configuration specify the certificates and keys used for encryption in the configuration of the Tentacle server and clients.

Run manually on the server with the parameters `-e`, `-k` and `-f`:

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

Execute manually on the client with the parameters `-e`, `-k` and `-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
```

If this manual execution works correctly, you may configure it permanently in the corresponding file:

- For a Tentacle server:

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

- For a Software Agent:

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

- For a SATELLITE server:

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

Secure configuration of Tentacle

Both the Tentacle server and the Software Agents can use secure communication with certificates and passwords, either direct communication between the two, or via a Tentacle Proxy server.

It is ALWAYS necessary to indicate in parameters the full paths where certificates are located, for instance `/etc/ssl/tentaclecert.pem`.

To use Tentacle's secure options, please verify that the `perl(IO::Socket::SSL)` package is *installed on your system*.

In the previous sections the different combinations are explained in detail; in this section the password options, Tentacle Proxy server and the use of `TENTACLE_EXT_OPTS` to set configurations are added. Also check this previous section for certificate names and keys on each side. A simplified syntax is used for didactic purposes only:

Simple transfer with password-based authentication:

Extra parameter in the server for password:

```
-x password
```

Extra parameter in the client for password (`TENTACLE_EXT_OPTS`):

```
-x password
```

Secure transfer, without client certificate:

Extra parameters on the server:

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

Secure transfer with client certificate

Extra parameters on the server:

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

Extra parameters on the client (`TENTACLE_EXT_OPTS`):

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

Secure transfer with client certificate and additional password authentication:

Extra parameters on the server:

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

Extra parameters on the client (`TENTACLE_EXT_OPTS`):

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

Secure setup use case with Tentacle proxy

It explains step by step how to configure both the Software Agents and the Tentacle server for secure communication, also using a Tentacle Proxy server.

Manual tests:

1. Start `tentacle_server` manually:

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

2. Start proxy manually:

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

3. Start `tentacle_client` manually:

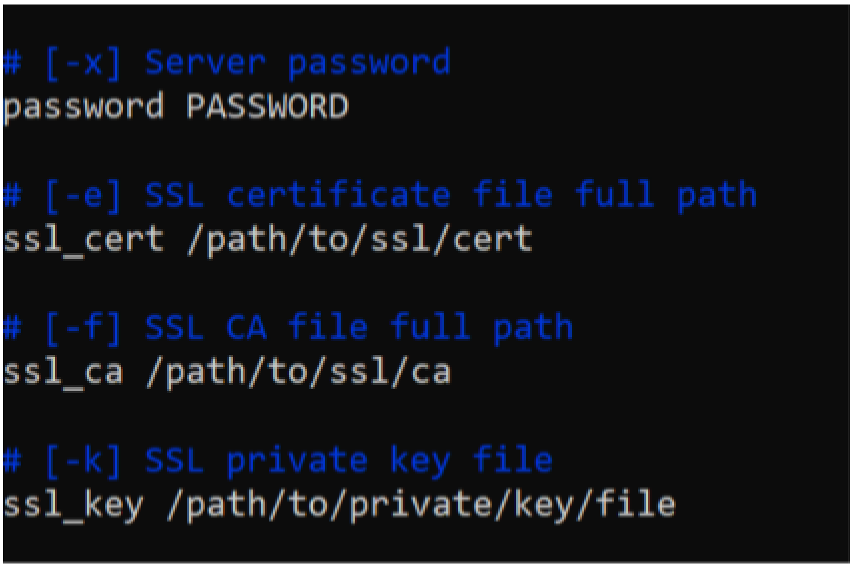
```
sudo -u //user//  tentacle_client \
           -a //ip_proxy/ip_server//  \
           -x password \
           -e tentaclecert.pem \
           -k tentaclekey.pem \
           -v //file//
```

Once you have verified that file submission was successful, you may proceed to permanently configure the `tentacle_server` and clients.

To configure the `tentacle_server` with the certificate options, edit the tentacle_serverd service configuration file, commonly located at `/etc/tentacle/tentacle_server.conf`, the same to configure an intermediate point to work as a proxy. To configure Software Agents to use Tentacle safe communication, edit the pandora_agent.conf configuration files, commonly located at `/etc/pandora/pandora_agent.conf`.

Permanent settings:

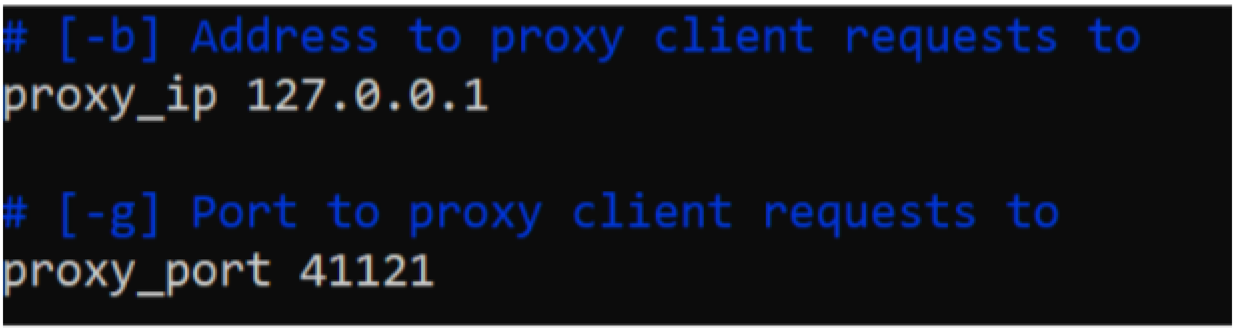
1. Start the server with SSL. Modify the configuration file `/etc/tentacle/tentacle_server.conf` and uncomment and complete the lines `password`, `ssl_cert`, `ssl_key`, `ssl_ca` with the values or the valid paths for your certificate:



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

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

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

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

Remember that every time you make changes to the Tentacle configuration file, it is necessary to restart the service for the changes to take effect: `/etc/init.d/tentacle_serverd start` .

2. Start the proxy. As in previous point number 1, modify the configuration file `/etc/tentacle/tentacle_server.conf` of the machine that will act as proxy. Likewise, uncomment and complete the `proxy_ip` and `proxy_port` lines with the valid configuration in your environment:



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

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

Remember that every time you make changes to the Tentacle configuration file, it is necessary to restart the service for the changes to take effect: `/etc/init.d/tentacle_serverd start` .

3. Start the Software Agent with the corresponding options. Modify the file `pandora_agent.conf`, look for the line `server_opts` and add:

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

Remember that the `server_ip` token should be set to point to the proxy's IP instead of the main server's. It would look like this:

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

If you do not want to use any of the options, such as the password, simply do not use the corresponding parameter.

Data compression in Tentacle

Version NG 725 or higher.

Tentacle allows you to enable data compression in transit with the `-z` command line option, reducing the size of transferred data at the expense of CPU load.

Pandora FMS Agent

Edit the file `/etc/pandora/pandora_agent.conf` and add `-z` to `server_opts`:

```
server_opts -z
```

Satellite server

Edit the file `/etc/pandora/satellite_server.conf` and add `-z` to `server_opts`:

```
server_opts -z
```

Configuration file elements

By default the Tentacle configuration file is located in `/etc/tentacle/tentacle_server.conf`.

Remember that every time you make changes to the Tentacle configuration file, you need to restart the service for the changes to take effect: `/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
```

- IPv4 address where the Tentacle server will listen. Multiple IP addresses can be separated by commas.
- Command line equivalent parameter: `-a`.

port

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

- Port number where the Tentacle server will listen.
- Command line equivalent parameter: `-p`.

max_connections

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

- Maximum number of simultaneous connections.
- Command line equivalent parameter: `-c`.

daemon

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

- Run as daemon `1`, otherwise `0`.
- Command line equivalent parameter: `-d`.

insecure

```

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

- Enable unsafe mode `1`, otherwise `0` (refers to operations such as listing files, etc.).
- Command line equivalent parameter: `-I`.

filters

```
# Filters (regexp:dir;regexp:dir...)
```

```
filters
.*\.conf:conf;.*\.md5:md5;.*\.zip:collections;.*\.lock:trans;.*\.rcmd:commands;.
*\.agent_setup\.exe:agent;.*\.agent_setup\.tar\.gz:agent
```

- It allows you to set filters for file types in specific directories. Enter a regular expression (filter as such) separated by `:` and the corresponding directory. To add another filter separate with `;`.
- Command line equivalent parameter: `-i`.

When upgrading to version 777, the Tentacle server must be restarted to take all the new filters installed.

max_size

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

- Maximum file size allowed (in bytes).
- Command line equivalent parameter: `-m`.

overwrite

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

- It allows overwriting if the received file has the same name and already exists, disabled by default (`0`), to activate it enter `1`.
- Command line equivalent parameter: `-o`.

quiet

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

- Avoid displaying error messages; activated `1`, deactivated `0`.
- Command line equivalent parameter: `-q`.

retries

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

- Number of retries for reading and writing operations.
- Command line equivalent parameter: `-r`.

directory

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

- It allows you to set the storage directory.
- Command line equivalent parameter: `-s`.

proxy_ip

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

- It allows you to set the IP address for an intermediary device (proxy client).
- Command line equivalent parameter: `-b`.

proxy_port

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

- It allows you to set the port number for an intermediary device (proxy client).
- Command line equivalent parameter: `-g`.

timeout

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

- Expiration time, in seconds, for read and write operations.
- Command line equivalent parameter: `-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
```

- It sets the amount of information to display for debugging purposes.
  - `-v 0` : No informational messages.
  - `-v 1` or `-v`: Only show important messages.
  - `-v 2` or `-V`: Show all messages.

log_file

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

- It allows you to establish a log or file to record events.
- Command line equivalent parameter: `-l`.

password

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

- Set the password for the Tentacle server.
- Command line equivalent parameter: `-x`.

ssl_cert

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

- It allows you to set the full path to the file that contains the SSL certificate.
- Command line equivalent parameter: `-e`.

ssl_ca

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

- It allows you to set the full path to the file that contains the Certification Authority (CA) of the SSL certificate.
- Command line equivalent parameter: `-f`.

ssl_key

```

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

- Fi locationle with the private key of the SSL certificate.
- Command line equivalent parameter: `-k`.

ssl_password

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

- If the SSL certificate contains a password, allow it to be requested (1) on the command line.
- Command line equivalent parameter: `-w`.

use_libwrap

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

- For Perl language, it allows using the `Authen::Libwrap` module. Enabled `1`, deactivated `0`.
- Command line equivalent parameter: `-T`.

ssl_version

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

- It sets one or more SSL versions allowed to be used separated by a colon (`:`), for example: `SSLv3:TLSv1:TLSv1.1:TLSv1.2:TLSv1.3`.
- Each version can be explicitly excluded by means of a closing exclamation point, for example to allow SSL2 and SSL3 and exclude the use of the other versions: `SSLv23:!TLSv1:!TLSv1_1:!SSLv3:!SSLv2`.
- Command line equivalent parameter: `-z`.

ssl_cipher

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

- It sets one or more encryption mechanisms allowed to be used separated by a colon (`:`), for example: `ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384`.
- Encryption mechanisms can be excluded by means of a closing exclamation point, for example `HIGH:!aNULL:!MD5:!3DES` means that high security ciphers are allowed, excluding those that are null, based on MD5 or 3DES.
- Command line equivalent parameter: `-u`.

Return to Pandora FMS documentation index