



# Referencia técnica del protocolo Tentacle



om:

<https://pandorafms.com/manual/!current/>

permanent link:

[https://pandorafms.com/manual/!current/es/documentation/pandorafms/technical\\_reference/09\\_tentacle](https://pandorafms.com/manual/!current/es/documentation/pandorafms/technical_reference/09_tentacle)

2024/03/18 21:07



# Referencia técnica del protocolo Tentacle

## ¿Qué es Tentacle?

Tentacle, una herramienta cliente/servidor de transferencia de ficheros, es:

- Segura por diseño.
- Fácil de usar e integrar con otras herramientas.
- Versátil, flexible y multiplataforma.

Tentacle ha sido creada para reemplazar herramientas más complejas como SCP/SSH y FTP para realizar transferencias simples de ficheros y dejar de usar mecanismos de autenticación poco seguros como .netrc , así como inicios de sesión interactivos automatizados con expect y el mecanismo de claves SSH, para pasar a usar una autenticación basada en el estándar X.509, utilizando certificados.

El cliente y el servidor han sido diseñados para ejecutarse desde línea de comandos, o llamados desde un *shellscript*. Tentacle es desde el año 2008 el método de transferencia por defecto para Pandora FMS, sustituyendo a SCP.

Tentacle está implementado en Perl y en ANSI C (ambas plataformas incluidas en MS Windows®).

Puede descargarlo y obtener más información en el sitio [Web oficial del proyecto en SourceForge](#).

## Guía del usuario de Tentacle en GNU Linux

### Instalación de la versión de Perl

#### Instalando desde Source Forge Net

Para instalar Tentacle server deberá contar con derechos equivalentes a usuario root, luego de haberlo instalado podrá ejecutarlo como un usuario estándar.

Obtenga el fichero `tentacle_server-762.tar.gz` desde Source Forge Net:

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

Por ejemplo (debe tener instalado wget):

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

## Instalación en Rocky Linux 8

- Descomprima el fichero descargado con `tar xzvf tentacle_server-762.tar.gz`.
- Instale el lenguaje Perl con `dnf install perl`.
- Entre al directorio con `cd tentacle`.
- Instale con `./tentacle_server_installer --install`.

## Instalación en CentOS 7

- Descomprima el fichero descargado con `tar xzvf tentacle_server-762.tar.gz`.
- Instale el lenguaje Perl con `yum install perl perl-IO-Compress zlib`.
- Entre al directorio con `cd tentacle`.
- Instale con `./tentacle_server_installer --install`.

## Instalación desde el SVN

El proceso consiste en descargar el código fuente mediante [Apache® Subversion®](#) (svn) y compilarlo. Para ello necesitará tener derechos de administrador o *root* (en esta documentación son las líneas que comiencen con el carácter numeral `#`). Usted es el único responsable de dicha clave.

Para instalar tanto la versión cliente como la del servidor, ejecutar:

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

Para instalar sólo la parte cliente ejecute:

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

Para instalar solo la parte del servidor ejecute:

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

```
# make install
```

Si quiere instalar en un directorio concreto, sustituya:

```
$ perl Makefile.PL
```

por:

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

## Instalación manual

Si make no está disponible en su sistema, puede hacer la instalación manualmente, copiando los ficheros `tentacle_client` y `tentacle_server` al directorio apropiado (por ejemplo, `/usr/local/bin`).

En este caso, si el binario de Perl no está situado en `/usr/bin/perl`, edite ambos ficheros de Tentacle y cambie la primera línea de forma que apunte a la ruta correcta donde se encuentre su binario de Perl. Así, por ejemplo, sustituya `location` por la ubicación de Perl en el sistema a instalar:

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

## Instalando la versión de C

### Instalando desde SVN

Teniendo en cuenta el preámbulo de instalación de la [sección anterior](#), para instalar el cliente de Tentacle ejecute:

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

Asegúrese de que la salida del comando `configure` no genera ningún error, dependencias incompletas de cabeceras, etc.

Para desactivar el soporte de OpenSSL, activado por defecto, sustituya:

```
$ ./configure
```

por:

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

## Ejemplos de uso de Tentacle

Para visualizar las opciones disponibles ejecute con el parámetro -h, tanto en la versión cliente como la versión servidor:

```
$ 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 not 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 not 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.)
```

Los valores predeterminados para todas opciones también se mostrarán en la ayuda.

Para todos los ejemplos mostrados a continuación, el servidor se encuentra en la dirección 192.168.1.1 y la clave privada del cliente no está protegida con contraseña.

- Transferencia simple de un fichero limitado a un tamaño máximo de 1 megabyte y depositado sobre /tmp:

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

- Transferencia simple en el puerto 65000 con el modo de sobreescritura activado:

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

- Transferencia simple con autenticación basada en contraseña:

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

- Transferencia segura, sin certificado cliente:

```
$ 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
```

- Transferencia segura con certificado de cliente:

```
$ 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
```

- Transferencia segura con certificado de cliente y autenticación adicional con contraseña (note el uso del conector para facilitar la escritura de varios parámetros):

```
$ 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
```

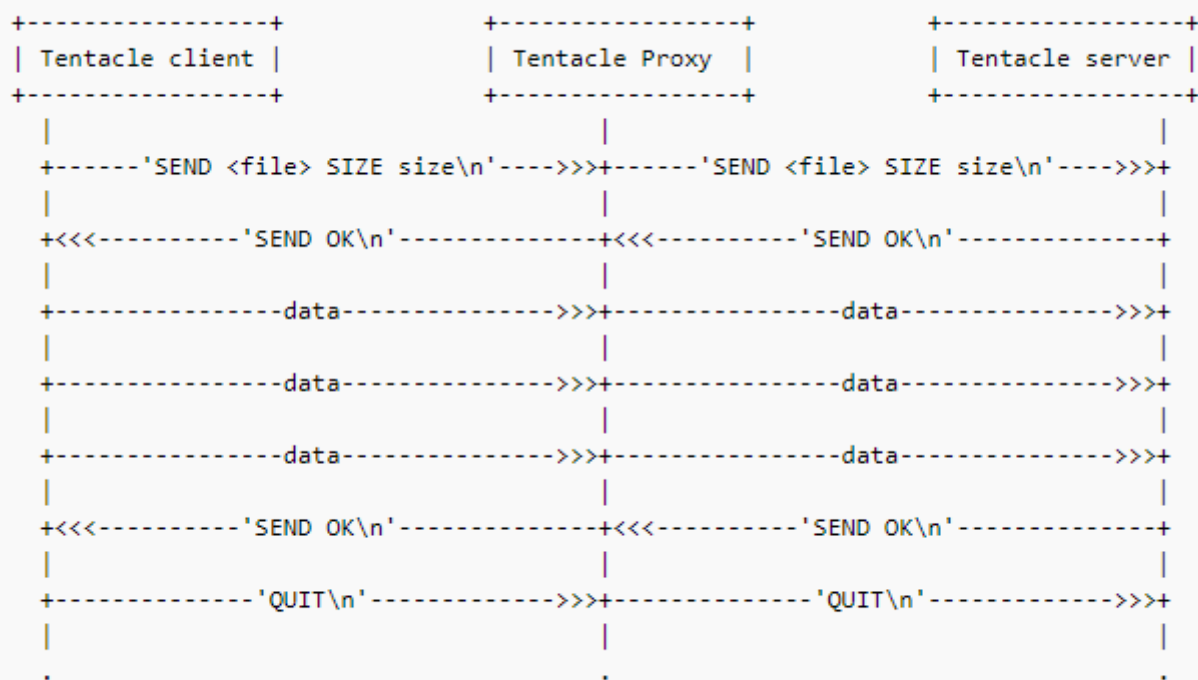
El servidor de Tentacle permite su configuración mediante un fichero de texto plano. Todas las opciones de línea de comandos están disponibles a través de este archivo. Si se especifica una misma opción de configuración en el fichero y en línea de comandos, tendrá preferencia el valor indicado en esta última. La ruta completa al fichero de configuración se indica con la opción -F.

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

## Tentacle Proxy

El servidor de Tentacle puede funcionar como *proxy* comunicando muchos clientes Tentacle hacia un servidor Tentacle inaccesible.

El siguiente diagrama muestra cómo funciona el servidor *proxy* de Tentacle:



El proxy no posee ninguna información, sino que únicamente envía la información desde los clientes hasta el servidor Tentacle. Por ejemplo, para lanzar el servidor Tentacle en modo proxy utilice los siguientes parámetros:



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

Estos parámetros son dirección IP (-b) y puerto (-g) *del servidor tentacle inaccesible*. Agregue, además, los parámetros normales en una sola línea:

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

El protocolo Tentacle en modo *proxy* también soporta [parámetros de autenticación y encriptación](#).

## Guía de Tentacle sobre MS Windows

Configure y ejecute el cliente y el servidor de Tentacle sobre MS Windows®.

### Instalación de la versión de Perl

#### Instalación del entorno de Perl

Mediante ActiveState® descargue usted ActivePerl 5.8 mediante el siguiente enlace y ejecute el instalador con las opciones por defecto:

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

#### Instalación del módulo IO-Socket-SSL

Descargue e instale OpenSSL desde:

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

Descargue los siguientes módulos de Perl :

- [http://archive.apache.org/dist/perl/win32-bin/ppms/Net\\_SSLeay.pm.ppd](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>

Ejecute desde línea de comandos en el directorio donde se encuentren los ficheros .ppd:

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

## Ejecución del cliente y el servidor de Tentacle

La ejecución es **similar a la de sistemas Unix/Linux**, únicamente necesita pasar delante el comando Perl, seguido de la sintaxis completa, por ejemplo:

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

## Definición del protocolo de Tentacle

El protocolo Tentacle en sí mismo es muy simple y directo. Algunas características de diseño importantes son:

- La comunicación la establece siempre el cliente.
- Los comandos siempre terminan con un carácter de fin de línea.
- Los siguientes caracteres no pueden formar parte de un nombre de archivo:

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

Los diagramas de secuencia ASCII se usarán para ilustrar los posibles casos. Los comandos se muestran en comillas simples.

### Enviar fichero(s)

Se muestra primero una transferencia de ficheros correcta

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

Para permitir multiples transferencias de archivos dentro de la misma sesión, un nuevo comando “SEND” debe ser enviado, después de una transferencia exitosa, y antes de un comando “QUIT”.

Si el servidor rechaza un fichero, un mensaje genérico de error es enviado de vuelta al cliente. Por razones de seguridad, no se muestran detalles de por qué falla el comando. Esto ocurre cuando:

- El fichero tiene un nombre de archivo no válido, o se especifica una vía.
- Está vacío o excede el tamaño máximo especificado por el servidor.
- Ya existe en el servidor y la sobre-escritura de archivos no está activada.

## Recepción de ficheros

Tentacle también soporta la solicitud de ficheros por parte del cliente.



El cliente tiene oportunidad de rechazar el fichero después de que el servidor informe acerca de su tamaño.

Igual que con el comando “SEND”, un nuevo comando “RECV” puede ser enviado después de una transferencia exitosa (incluso si el fichero ha sido rechazado por el cliente) y siempre antes del comando “QUIT”. Un error genérico será enviado si el servidor rechaza enviar el fichero. Eso último puede ocurrir cuando:

- Tiene un nombre de fichero inválido, o se ha especificado una vía.
- No existe en el servidor.

```
+-----+           +-----+
| Tentacle client |   | Tentacle server |
+-----+           +-----+

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

## Autenticación mediante contraseña

Si el servidor requiere una contraseña (*password*), el cliente debe autenticarse antes de enviar cualquier otro comando.

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

Un doble MD5 de la contraseña será enviado para ofuscar. Si usted está trabajando sobre una conexión sin cifrar esto NO implementa o agrega seguridad alguna. *Si usted necesita seguridad utilice conexiones cifradas con SSL* .

## Gestión de errores

Ante cualquier error, el servidor cerrará la conexión sin dar explicaciones. Esto puede ser a causa de un comando incorrecto, una contraseña errónea, más datos enviados de los que se supone que se iban a enviar, o cualquier otra razón que hace que la operativa del servidor se salga de lo establecido o considerado “normal”.

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

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

Por defecto, el *log* de Tentacle está configurado en */dev/null*.

## Guía rápida de certificados OpenSSL

Esta es una guía rápida de iniciación en certificados de OpenSSL para su uso con Tentacle u otras aplicaciones. Para más información puede consultar la web oficial del proyecto OpenSSL:

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

### Creación de un certificado

Preparación del entorno:

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

Recuerde establecer, por seguridad, permisos de escritura y lectura de los diferentes usuarios en su sistema en las carpetas recién creadas.

El siguiente paso es realizar un certificado de CA autofirmado y moverlo a los directorios creados:

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

Rellene los campos solicitados para el certificado y recuérdelos bien porque se necesitarán de nuevo más adelante, exactamente iguales. Ahora debe crear una petición de certificado:

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

Firmar la petición de certificado, estableciendo además un seriado consecutivo de los mismos como mecanismo de control y auditoría:

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

Tenga en cuenta que de presentar el [archivo de semilla aleatorio](#) algún inconveniente, puede borrarlo con derechos de usuario *root*: `sudo rm ~/.rnd`. De esta manera puede ser creado de

nuevo con sus propios derechos de escritura y lectura. Usted es el único responsable de dicha clave *root*.

## Crear un certificado autofirmado

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

## Generar una llave privada RSA

Esto es muy útil para evitar tener que meter una contraseña en el lado cliente usando Tentacle.

Generar la llave:

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

Y sustituir `-keyout` con `-key` en las secciones anteriores.

## Exportar certificado a otro formato

Los certificados pueden necesitarse en formato DER en lugar de PEM para algunos sistemas operativos (como Ubuntu® o Windows®). Si es el caso, se puede obtener el certificado en ese formato a partir del PEM generado:

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

## Configuración de comunicación segura con Tentacle

Se explica paso a paso cómo configurar tanto los Agentes Software como los servidores Tentacle para una comunicación segura.

En primer lugar, es muy recomendable realizar pruebas a mano desde los terminales para asegurar que la configuración, parámetros y certificados son correctos.

Luego se podrá realizar una configuración permanente en los respectivos ficheros de configuración:

Servidores Tentacle

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

Agentes Software en Unix/Linux

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

Agentes Software en MS Windows®

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

Servidores Satellite

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

Servidores Tentacle Proxy

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

Recuerde reiniciar los servicios correspondientes después de cualquier modificación. En el caso de Unix/Linux también puede utilizar la opción `TENTACLE_EXT_OPTS` ubicada en `/etc/init.d/tentacle_serverd` (puede consultar el resto de las opciones para dicho demonio [en este enlace](#) ).

## Cifrado de la comunicación

Para cifrar la comunicación entre los clientes y el servidor de Tentacle, será necesario contar previamente con certificados y claves SSL. En esta guía veremos todas las opciones de configuración posibles, por lo que los certificados pueden ser tanto **autofirmados** como firmados por una CA válida.

Para evitar confusiones en este artículo los certificados y claves de cada lado están identificados con los siguientes nombres:

- `ca_cert`: Certificado de la CA usada para la firma de los certificados.
- `tentacle_key`: Clave generada para el servidor de Tentacle.
- `tentacle_cert`: Certificado generado para el servidor de Tentacle.
- `tentacle_client_key`: Clave generada para el cliente de Tentacle.
- `tentacle_client_cert`: Certificado generado para el cliente de Tentacle.

Es necesario SIEMPRE indicar en los parámetros las rutas absolutas donde se encuentren los certificados, por ejemplo `/etc/ssl/tentaclecert.pem`

Para utilizar las opciones seguras de Tentacle, por favor, verifique que el paquete `perl (IO::Socket::SSL)` esté *instalado en su sistema*.

## Configuración de certificados en el servidor de Tentacle aceptando cualquier certificado en el cliente

Para esta configuración debe indicar el certificado y la clave usados para el cifrado en la configuración del servidor de Tentacle.

Ejecute manualmente en el servidor con los parámetros -e y -k

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

Ejecute manualmente en el cliente con el parámetro -c:

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

Si esta ejecución manual funciona correctamente, puede hacer la configuración permanente en el respectivo fichero:

- Para un servidor Tentacle:

```
ssl_cert tentacle_cert
ssl_key tentacle_key
```

- Para un Agente Software:

```
server_opts -c
```

- Para un servidor Satellite:

```
server_opts -c
```

## Configuración de certificados en el servidor de Tentacle y en el cliente verificando el certificado con una CA específica en el cliente

Para esta configuración debe indicar el certificado y la clave usados para el cifrado en la configuración del servidor de Tentacle y los certificados usados para el cifrado en los clientes.

Ejecute manualmente en el servidor con los parámetros -e y -k

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

Ejecute manualmente en el cliente con el parámetro -e y -f:

```
# echo test> file.txt
```



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

Si esta ejecución manual funciona correctamente, puede hacer la configuración permanente en el respectivo fichero:

- Para un servidor Tentacle:

```
ssl_cert tentacle_cert
ssl_key tentacle_key
```

- Para un Agente Software:

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

- Para un servidor Satellite:

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

## Configuración de certificados en el servidor de Tentacle y en el cliente verificando el certificado con una CA específica en el servidor

Para esta configuración debe indicar los certificados y las claves usados para el cifrado en la configuración del servidor de Tentacle y de los clientes.

Ejecute manualmente en el servidor con los parámetros -e, -k y -f

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

Ejecute manualmente en el cliente con los parámetros -e y -k (note el uso del conector de líneas \):

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

Si esta ejecución manual funciona correctamente, puede hacer la configuración permanente en el respectivo fichero:

- Para un servidor Tentacle:

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

- Para un Agente Software:

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

- Para un servidor Satellite:

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

## Configuración de certificados en el servidor de Tentacle y en el cliente verificando el certificado con una CA específica en ambos

Para esta configuración debe indicar los certificados y las claves usados para el cifrado en la configuración del servidor de Tentacle y de los clientes.

Ejecute manualmente en el servidor con los parámetros -e, -k y -f:

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

Ejecute manualmente en el cliente con los parámetros -e, -k y -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 esta ejecución manual funciona correctamente, puede hacer la configuración permanente en el respectivo fichero:

- Para un servidor Tentacle:

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

- Para un Agente Software:

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

- Para un servidor Satellite:

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

## Configuración segura de Tentacle

Tanto el servidor Tentacle como los Agentes Software pueden utilizar una comunicación segura con certificados y contraseña, bien sea comunicación directa entre ambos, o bien sea mediante un

servidor Tentacle Proxy.

Es necesario SIEMPRE indicar en los parámetros las rutas absolutas donde se encuentren los certificados, por ejemplo `/etc/ssl/tentaclecert.pem`

Para utilizar las opciones seguras de Tentacle, por favor, verifique que el paquete `perl (IO::Socket::SSL)` esté *instalado en su sistema*.

En las secciones anteriores se explica de manera detallada las diversas combinaciones; en esta sección se agregan las opciones de contraseña, servidor Tentacle Proxy y el uso de `TENTACLE_EXT_OPTS` para fijar configuraciones. También revise en esta sección anterior los nombres de los certificados y las claves de cada lado. Se emplea una sintaxis simplificada solo con propósitos didácticos:

Transferencia simple con autenticación basada en contraseña:

Parámetro extra en el servidor para contraseña:

```
-x password
```

Parámetro extra en el cliente para contraseña ( `TENTACLE_EXT_OPTS` ):

```
-x password
```

Transferencia segura, sin certificado cliente:

Parámetros extra en el servidor:

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

Transferencia segura con certificado de cliente

Parámetros extra en el servidor:

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

Parámetros extra en el cliente ( `TENTACLE_EXT_OPTS` ):

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

Transferencia segura con certificado de cliente y autenticación adicional con contraseña:

Parámetros extra en el servidor:

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

Parámetros extra en el cliente ( TENTACLE\_EXT\_OPTS ):

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

### Caso práctico de configuración segura con Tentacle proxy

Se explica paso a paso cómo configurar tanto los Agentes Software como el servidor Tentacle para una comunicación segura, utilizando también un servidor Tentacle Proxy.

Pruebas manuales:

1. Iniciar tentacle\_server manualmente:

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

2. Iniciar proxy manualmente:

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

3. Iniciar tentacle\_client manualmente:

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

Cuando haya comprobado que el envío del archivo ha tenido éxito, puede proceder a configurar permanentemente el tentacle\_server y los clientes.

Para configurar el tentacle\_server con las opciones de certificado, hay que editar el archivo de configuración del servicio tentacle\_serverd, comúnmente ubicado en /etc/tentacle/tentacle\_server.conf, lo mismo para configurar un punto intermedio para que actúe como proxy. Para configurar los Agentes Software para que utilicen la comunicación

segura de Tentacle, debe editar los ficheros de configuración `pandora_agent.conf`, comúnmente ubicados en `/etc/pandora/pandora_agent.conf`.

Configuración permanente:

1. Arrancar el server con SSL. Modificar el archivo de configuración `/etc/tentacle/tentacle_server.conf` y descomentar y completar las líneas `password`, `ssl_cert`, `ssl_key`, `ssl_ca` con los valores o las rutas válidas para su certificado:

```
# [-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
```

Recuerde que cada vez que realice cambios en el archivo de configuración de Tentacle, es necesario reiniciar el servicio para que apliquen los cambios:  
`/etc/init.d/tentacle_serverd start`.

2. Iniciar el *proxy*. Al igual que en el punto anterior número 1, modificar el archivo de configuración `/etc/tentacle/tentacle_server.conf` de la máquina que va a actuar como *proxy*. Igualmente, *descomentar* y completar las líneas `proxy_ip` y `proxy_port` con la configuración válida en su entorno:

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

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

Recuerde que cada vez que realice cambios en el archivo

de configuración de Tentacle, es necesario reiniciar el servicio para que se apliquen los cambios:

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

3. Arrancar el Agente Software con las opciones correspondientes. Modificar el archivo `pandora_agent.conf`, buscar la línea `server_opts` y añadir:

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

Recuerde que el *token* `server_ip` debe configurar apuntando a la IP del *proxy* en lugar de la del servidor principal. Quedaría así:

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

Si no se quiere utilizar alguna de las opciones, como por ejemplo la contraseña, basta con no utilizar el parámetro correspondiente.

## Compresión de datos en Tentacle

Versión NG 725 o superior.

Tentacle permite habilitar la compresión de datos en tránsito con la opción de línea de comandos `-z`, reduciendo el tamaño de los datos transferidos a expensas de la carga de CPU.

## Pandora FMS Agent

Edite el fichero `/etc/pandora/pandora_agent.conf` y añada `-z` a `server_opts`:

```
server_opts -z
```

## Satellite server

Edite el fichero `/etc/pandora/satellite_server.conf` y añada `-z` a `server_opts`:

```
server_opts -z
```

## Elementos del fichero de configuración

Por defecto el fichero de configuración de Tentacle está ubicado en `/etc/tentacle/tentacle_server.conf`.

Recuerde que cada vez que realice cambios en el archivo de configuración de Tentacle, es necesario reiniciar el servicio para que apliquen los cambios:  
`/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
```

- Dirección IPv4 donde escuchará el servidor Tentacle. Varias direcciones IP puede ser separadas mediante comas.
- Parámetro equivalente por línea de comando: `-a`.

### port

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

- Número de puerto donde escuchará el servidor Tentacle.
- Parámetro equivalente por línea de comando: `-p`.

### max\_connections

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

- Número máximo de conexiones simultáneas.
- Parámetro equivalente por línea de comando: `-c`.

### daemon

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

- Ejecutar como *daemon*1, caso contrario 0.
- Parámetro equivalente por línea de comando: `-d`.

## insecure

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

- Habilitar modo inseguro 1, caso contrario 0 (se refiere a operaciones como listar ficheros, etcétera).
- Parámetro equivalente por línea de comando: -I.

## filters

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

- Permite establecer filtros por tipos de archivos en directorios específicos. Coloque una expresión regular (filtro como tal) separado por : y el directorio correspondiente. Para agregar otro filtro separe con ;.
- Parámetro equivalente por línea de comando: -i.

## max\_size

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

- Tamaño máximo de fichero permitido (en bytes).
- Parámetro equivalente por línea de comando: -m.

## overwrite

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

- Permite sobrescribir si el fichero recibido tiene el mismo nombre y ya existe, desactivado por defecto (0), para activarlo coloque 1.
- Parámetro equivalente por línea de comando: -o.

## quiet

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

- Evita mostrar mensajes de error; activado 1, desactivado 0.
- Parámetro equivalente por línea de comando: -q.

## retries



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

- Número de reintentos para operaciones de lectura y escritura.
- Parámetro equivalente por línea de comando: -r.

## directory

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

- Permite establecer el directorio de almacenamiento.
- Parámetro equivalente por línea de comando: -s.

## proxy\_ip

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

- Permite establecer la dirección IP para un dispositivo intermediario (*proxy client*).
- Parámetro equivalente por línea de comando: -b.

## proxy\_port

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

- Permite establecer el número de puerto para un dispositivo intermediario (*proxy client*).
- Parámetro equivalente por línea de comando: -g.

## timeout

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

- Tiempo de expiraciones, en segundos, para operaciones de lectura y escritura.
- Parámetro equivalente por línea de comando: -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
```

- Establece la cantidad de información a mostrar con propósitos de depuración.

- -v 0 : Sin mensajes informativos.
- -v 1 o -v: Solamente muestra mensajes importantes.
- -v 2 o -V: Muestra todos los mensajes.

## log\_file

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

- Permite establecer un *log* o fichero para el registro de eventos.
- Parámetro equivalente por línea de comando: -l.

## password

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

- Establece la contraseña para el servidor Tentacle.
- Parámetro equivalente por línea de comando: -x.

## ssl\_cert

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

- Permite establecer la ruta completa al fichero que contiene el certificado SSL.
- Parámetro equivalente por línea de comando: -e.

## ssl\_ca

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

- Permite establecer la ruta completa al fichero que contiene la Autoridad Certificadora (CA) del **certificado SSL**.
- Parámetro equivalente por línea de comando: -f.

## ssl\_key

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

- Ubicación del fichero con la clave privada del certificado SSL.
- Parámetro equivalente por línea de comando: -k.

## ssl\_password

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

- Si el certificado SSL contiene contraseña permite solicitarla (1) en la línea de comando.
- Parámetro equivalente por línea de comando: -w.

## use\_libwrap

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

- Para lenguaje Perl permite utilizar el módulo Authen::Libwrap. Activado 1, desactivado 0.
- Parámetro equivalente por línea de comando: -T.

## ssl\_version

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

- Establece una o varias versiones SSL permitidas a utilizar separadas por dos puntos (:), por ejemplo: SSLv3:TLSv1:TLSv1.1:TLSv1.2:TLSv1.3.
- Se pueden excluir de manera explícita cada versión por medio de un signo de admiración de cierre, por ejemplo para permitir SSL2 y SSL3 y excluir el uso de las demás versiones: SSLv23:!TLSv1:!TLSv1\_1:!SSLv3:!SSLv2.
- Parámetro equivalente por línea de comando: -z.

## ssl\_cipher

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

- Establece una o varios mecanismos de cifrado permitidos a utilizar separados por dos puntos (:), por ejemplo: ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384.
- Se pueden excluir mecanismos de cifrado por medio de un signo de admiración de cierre, por ejemplo HIGH:!aNULL:!MD5:!3DES significa que se permiten ciphers de alta seguridad, excluyendo aquellos que son nulos, basados en MD5 o 3DES.
- Parámetro equivalente por línea de comando: -u.

[Volver al índice de documentación de Pandora FMS](#)