# Setting up a secure communication (SSL/TLS)

The D2000 system can be configured to ensure that communication between the server and clients takes place through a secure encrypted communication channel. Security is implemented by **Transport Layer Security** (TLS v1.2).

The following steps are required to enable secure communication:

## 1. For the server, it is necessary to obtain/generate the encryption key and certificate. The certificate has to be distributed through the client process.

The key and certificate can be generated, for example, using the **openssl** utility (https://slproweb.com/products/Win32OpenSSL.html).

### Generating an encryption key

```
openssl genrsa -out server.key 4096
```

### Generating a certificate signing request

```
openssl req -new -key server.key -out server.csr
```

### Generating a self-signed certificate

```
openssl x509 -req -days 730 -in server.csr -signkey server.key -out server.crt
```

## 2. Setting up TLS support in the kernel registers

```
 HKEY_LOCAL_MACHINE\SOFTWARE\Ipesoft\<instalacia>\cfg_<aplikacia>\TLS_Server\TLS_CertFile = c:\<cesta>\server.
crt
 HKEY_LOCAL_MACHINE\SOFTWARE\Ipesoft\<instalacia>\cfg_<aplikacia>\TLS_Server\TLS_KeyFile = c:\<cesta>\server.key
 HKEY_LOCAL_MACHINE\SOFTWARE\Ipesoft\<instalacia>\cfg_<aplikacia>\TLS_Server\TLS_RequiredLevel = <level>
```

Setting the required security level of the connecting client <level>:

- **None** - kernel allows client to connect without security and also with security
- **TLSNoPeerAuth** - kernel allows connection only from a client who communicates securely

## 3. Setting up TLS support in the registers for clients

```
 HKEY_LOCAL_MACHINE\SOFTWARE\Ipesoft\<instalacia>\cfg_<aplikacia>\TLS_Client\TLS_TrustedCerts = c:
\<cesta>\server.crt
 HKEY_LOCAL_MACHINE\SOFTWARE\Ipesoft\<instalacia>\cfg_<aplikacia>\TLS_Client\TLS_RequiredLevel = <level>
```

Setting the required security level of the connecting client <level>:

- **None** - the client will connect to the kernel if the kernel supports secure communication and even if it does not support secure communication
- **TLSNoPeerAuth** - the client will only connect to the kernel ensuring secure communication and it is verifiable by the certificate

## 4. To use TLS, the client must **also start with /C<*application_name*> parameter** in addition to the usual parameters (/S, /RD or /RF)

The reason is to already know the name of the application before connecting to the application server and load the parameters from the TLS  registers (see point 3).

ⓘ **Related pages:**

- D2000 system processes
- Start parameters of processes