Enable TLS/SSL trasnferring layer encryption in SuperSocket

English (United States) ▾ v1.6 ▾

Keywords: TLS, SSL, Certificate, X509 Certificate, Local Certificate Store

SuperSocket supports the transport layer encryption (TLS/SSL)

SuperSocket has automatically support for TLS/SSL. You needn't change any code to let your socket server support TLS/SSL.

To enable TLS/SSL for your SuperSocket server, you should prepare a certificate at first.

There are two ways to provide the certificate:

1. a X509 certificate file with private key
   - for testing purpose you can generate a certificate file by the CertificateCreator in SuperSocket(http://supersocket.codeplex.com/releases/view/59311)
   - in production environment, you should purchase a certificate from a certificate authority
2. a certificate in local certificate store

Enable TLS/SSL by a certificate file

You should change the configuration file to use the certificate file following the below steps:

1. set security attribute for the server node;
2. add the certificate node in server node as child.

The configuration should look like:

```
<server name="EchoServer"
        serverTypeName="EchoService"
        ip="Any" port="443"
        security="tls">
    <certificate filePath="localhost.pfx" password="supersocket"></certificate>
</server>
```

Note: the password of the certificate node is the private key of the certificate file.

There is one more option named "**keyStorageFlags**" for certificate loading:

```
<certificate filePath="localhost.pfx"
             password="supersocket"
             keyStorageFlags="UserKeySet"></certificate>
```

You can read the MSDN article below for more information about this option: http://msdn.microsoft.com/zh-cn/library/system.security.cryptography.x509certificates.x509keystorageflags(v=vs.110).aspx (http://msdn.microsoft.com/zh-cn/library/system.security.cryptography.x509certificates.x509keystorageflags(v=vs.110).aspx)

Enable TLS/SSL by a certificate in local certificate store

You also can use a certificate in your local certificate store without a physical file. The thumbprint of the certificate you want to use is required:

```
<server name="EchoServer"
        serverTypeName="EchoService"
        ip="Any" port="443"
        security="tls">
    <certificate storeName="My" thumbprint="f42585bceed2cb049ef4a3c6d0ad572a6699f6f3"></certificate>
</server>
```

Other optional options:

- **storeLocation** - CurrentUser, LocalMachine

  ```
  <certificate storeName="My"
               thumbprint="f42585bceed2cb049ef4a3c6d0ad572a6699f6f3">
               storeLocation="LocalMachine"
  </certificate>
  ```

You also can only apply TLS/SSL for one listener of the appserver instance

```
<server name="EchoServer" serverTypeName="EchoService" maxConnectionNumber="10000">
    <certificate storeName="My" thumbprint="f42585bceed2cb049ef4a3c6d0ad572a6699f6f3"></certificate>
    <listeners>
      <add ip="Any" port="80" />
      <add ip="Any" port="443" security="tls" />
    </listeners>
</server>
```

Client certificate validation

In TLS/SSL communications, the client side certificate is not a must, but some systems require much higher security guarantee. This feature allow you to validate the client side certificate from the server side.

At first, to enable the client certificate validation, you should add the attribute "clientCertificateRequired" in the certificate node of the configuration:

```
<certificate storeName="My"
             storeLocation="LocalMachine"
             clientCertificateRequired="true"
             thumbprint="f42585bceed2cb049ef4a3c6d0ad572a6699f6f3"/>
```

Then you can override the AppServer's method "ValidateClientCertificate(...)" the implement your validation logic:

```
protected override bool ValidateClientCertificate(YourSession session, object sender, X509Certificate certificate, X509Chain chain, SslPolicyErrors sslPolicyErrors)
{
   //Check sslPolicyErrors

   //Check certificate

   //Return checking result
   return true;
}
```

- Prev: The Built in Flash Silverlight Policy Server in SuperSocket (/v1-6/en-US/The-Built-in-Flash-Silverlight-Policy-Server-in-SuperSocket)
- Next: Run SuperSocket in Windows Azure (/v1-6/en-US/Run-SuperSocket-in-Windows-Azure)