Authorizing Access to Service Operations

.NET Framework 4.5   Este tópico ainda não foi avaliado como

This sample demonstrates how to use the <serviceAuthorization> element to enable use of the PrincipalPermissionAttribute attribute to authorize access to service operations. This sample is based on the Getting Started Sample sample. The service and client are configured using the wsHttpBinding Element. The **mode** attribute of the Security element has been set to **Message** and **clientCredentialType** has been set to **Windows**. The PrincipalPermissionAttribute is applied to each service method and used to restrict access to each operation. The caller must be a Windows administrator to access each operation.

In this sample, the client is a console application (.exe) and the service is hosted by Internet Information Services (IIS).

Note:
The setup procedure and build instructions for this sample are located at the end of this topic.

The service configuration file uses the <serviceAuthorization> element to set the **principalPermissionMode** attribute:

```
<behaviors>
  <serviceBehaviors>
    <behavior>
      ...
      <!-- The serviceAuthorization behavior sets the          principalPermission
           This puts a WindowsPrincipal on the current thread when a            ser
      <serviceAuthorization principalPermissionMode="UseWindowsGroups" />
    </behavior>
  </serviceBehaviors>
</behaviors>
```

Setting the **principalPermissionMode** to **UseWindowsGroups** enables the use of PrincipalPermissionAttribute based on Windows group names.

The PrincipalPermissionAttribute is applied to each operation to require the caller to be part of the Windows administrators group, as shown in the following sample code.

```
[PrincipalPermission(SecurityAction.Demand,
                     Role = "Builtin\\Administrators")]
public double Add(double n1, double n2)
{
    double result = n1 + n2;
    return result;
```

```
    }
```

When you run the sample, the operation requests and responses are displayed in the client console window. The client successfully communicates with each operation if it is running under an account that is part of the Administrators group; otherwise, access is denied. To experiment with authorization failure, run the client under an account that is not part of the Administrators group. Press ENTER in the console window to shut down the client.

A service can be notified of authorization failures by implementing an IErrorHandler. See Extending Control Over Error Handling and Reporting for information about implementing `IErrorHandler`.

To set up, build, and run the sample

1. Ensure that you have performed the One-Time Setup Procedure for the Windows Communication Foundation Samples.

2. To build the C# or Visual Basic .NET edition of the solution, follow the instructions in Building the Windows Communication Foundation Samples.

3. To run the sample in a single- or cross-computer configuration, follow the instructions in Running the Windows Communication Foundation Samples.

Contribuições da comunidade