Bài 7

SECURITY TRONG WINDOWS COMMUNICATION FOUNDATION

Mục lục

1 Tổng quan về bảo mật 2
1.1 Ích lợi từ bảo mật trong WCF 2
1.1.1 Tích hợp với các kiến trúc bảo mật có sẵn 3
1.1.2 Tích hợp với mô hình xác thực có sẵn 3
1.1.3 Các chuẩn và tính interoperability (làm việc liên môi trường) 3
1.2 Các lĩnh vực bảo mật của WCF 3
1.2.1 Transfer security 4
1.2.2 Các chế độ bảo mật vận chuyển và bản tin 4
1.2.3 Điều khiển truy nhập 4
1.2.4 Auditing 4
2 Lập trình với bảo mật trong WCF 4
2.1 Đặt chế độ bảo mật 4
2.2 Chọn kiểu xác thực client 6
2.3 Thiết lập các giá trị credentials cho dịch vụ 6
2.4 Thiết lập các giá trị credentials cho client 7
3 Xây dựng ví dụ với bảo mật trong WCF 8
3.1 Tạo dịch vụ WCF 8
3.2 Thêm chức năng cho dịch vụ 9
3.3 Thiết lập dịch vụ WCF sử dụng wsHttpBinding với Windows Authentication và Message Security 12
3.4 Tạo ứng dụng client 13
3.5 Thêm tham chiếu tới dịch vụ WCF cho client 14
3.6 Thêm chức năng cho client 14
4 Câu hỏi ôn tập 15
5 Tài liệu tham khảo 15

Các bài trước chúng ta đã biết qua về các thành phần tạo nên một dịch vụ trên Windows Communication Foundation, đồng thời cũng biết cách xây dựng một client để sử dụng các dịch vụ đó. Tuy vậy chúng ta chưa hề đề cập tới một vấn đề rất quan trọng trong truyền dữ liệu đó là việc đảm bảo an toàn dữ liệu cũng như tính nhất quán của dữ liệu. Trong một môi trường phân tán, việc truyền và gửi dữ liệu giữa client với dịch vụ cần phải được bảo vệ khỏi sự dòm ngó của những đối tượng không mong muốn. Bạn cần phải đảm bảo chắc chắn rằng, dữ liệu chỉ trao đổi giữa các thành phần có đủ thẩm quyền mà thôi.

Với yêu cầu trên, Windows Communication Foundation cung cấp cho bạn một nền tảng cho phép bạn thực hiện trao đổi dữ liệu (bản tin) thông qua rất nhiều giao thức khác nhau, trong đó có cả các giao thức hỗ trợ bảo mật để phù hợp với từng yêu cầu của bạn.

Bài này sẽ tập trung thảo luận về các khía cạnh khác nhau về bảo mật với Windows Communication Foundation và cách khai thác chúng. Mặc dù lĩnh vực bảo mật thông tin là một lĩnh vực rất lớn, trong bài này chúng ta chỉ tập trung vào khía cạnh bảo mật bản tin trong WCF mà thôi. Chúng ta sẽ thảo luận những vấn đề sau:

- Tổng quan về bảo mật với Windows Communication Foundation
- Lập trình với bảo mật trong Windows Communication Foundation

1 Tổng quan về bảo mật

Như các bạn đã biết Windows Communication Foundation là một nền tảng lập trình phân tán dựa trên cơ sở các bản tin SOAP. Do vậy cần phải đảm bảo tính bảo mật của các bản tin trao đổi giữa client với dịch vụ để có thể bảo vệ được dữ liệu khỏi những đối tượng không mong muốn. Việc này có thể thực hiện nhờ vào việc WCF hỗ trợ trao đổi các bản tin bảo mật (secure message) dựa trên kiến trúc bảo mật nền tảng cũng như các chuẩn bảo mật cho các bản tin SOAP.

WCF sử dụng các khái niệm mà bạn đã quen khi bạn xây dựng các ứng dụng bảo mật phân tán với các công nghệ có sẵn như HTTPS, Windows integrated security (bảo mật tích hợp với hệ điều hành Windows), hoặc sử dụng username với password để xác thực người dùng. WCF không chỉ tích hợp với nền tảng bảo mật có sẵn mà còn mở rộng mức độ hỗ trợ ra khỏi giới hạn của hệ điều hành Windows bằng cách sử dụng các bản tin SOAP bảo mật.

1.1 Ích lợi từ bảo mật trong WCF

Do WCF là một nền tảng lập trình phân tán dựa trên các bản tin SOAP, với việc sử dụng WCF, các ứng dụng bạn tạo ra có thể tạo và xử lý các bản tin từ một số không hạn chế các dịch vụ và client khác. Trong các ứng dụng này, các bản tin có thể được truyền từ điểm này sang điểm khác, thông qua các thiết bị như tường lửa, router, switch, hay qua Internet, và qua một loạt các điểm trung chuyển SOAP. Điều này tạo ra không ít mối đe doạ tới an ninh của bản tin. Những ví dụ sau đây cho ta thấy một số các mối đe doạ thường thấy khi trao đổi bản tin giữa các ứng dụng, những mối đe doạ này hoàn toàn có thể loại bỏ được nhờ vào sử dụng tính năng bảo mật trong WCF:

- Quan sát các bản tin trên mạng để lấy ra các thông tin nhạy cảm. Ví dụ ở máy client thực hiện login vào một hệ thống sử dụng chế độ gửi tên tài khoản và mật khẩu dạng text không mã hoá. Hacker hoàn toàn có thể bắt được bản tin đó và trích ra thông tin về tài khoản cùng với mật khẩu.
- Đóng giả một dịch vụ mà client không hề biết. Việc này cũng tương tự như web phishing, nghĩa là làm giả một trang web giống như trang web mà người dùng quen thuộc (như trang web yahoo hay trang web ngân hàng). Người dùng sẽ nhập thông tin về tài khoản cùng với mật khẩu để đăng nhập vào trang giả đó. Khi đó hacker sẽ có được các thông tin này.
- Thay đổi nội dung bản tin. Hacker hoàn toàn có thể thay đổi nội dung của một bản tin mà client lẫn dịch vụ không biết.
- Etc.

Việc đảm bảo bảo mật cho các bản tin trao đổi giữa client với dịch vụ WCF cần phải chú trọng ở những điểm sau:

- Xác thực điểm cuối dịch vụ
- Xác thực client
- Tính nhất quán của bản tin
- Tính bảo mật của bản tin
- Phát hiện replay (hiện tượng lặp lại yêu cầu của client hoặc dịch vụ mà thực chất client/dịch vụ không đưa ra)

1.1.1 Tích hợp với các kiến trúc bảo mật có sẵn

WCF hoàn toàn có thể làm việc với các giải pháp bảo mật có sẵn như Secure Sockets Layer (SSL) hoặc giao thức Kerbeos. Ngoài ra nó cũng có thể làm việc với kiến trúc bảo mật đang sử dụng như domain trên Windows sử dụng Active Directory. Ngoài việc hỗ trợ các giải pháp bảo mật thường thấy, WCF còn tích hợp với các mô hình bảo mật sẵn có ở tầng vận chuyển và có thể chuyển hạ tầng sẵn có sang các mô hình mới hơn dựa trên bảo mật các bản tin SOAP.

1.1.2 Tích hợp với mô hình xác thực có sẵn

Một phần quan trọng trong bất kỳ mô hình bảo mật truyền tin nào là khả năng xác định và xác thực các thực thể trong quá trình trao đổi dữ liệu. Các thực thể trong quá trình này sử dụng các "định danh điện tử", hay còn gọi là credentials, để xác thực chúng với đối tượng đang trao đổi. Với sự phát triển của các nền tảng phân tán, có rất nhiều cách để xác thực các credentials được tạo ra. Lấy ví dụ, trên Internet cách thông dụng nhất là sử dụng tên tài khoản cùng với mật khẩu để xác thực người dùng. Trong mạng intranet, mô hình thông dụng là sử dụng Kerberos domain controller để xác thực người dùng cùng với dịch vụ. Do vậy với cùng một dịch vụ, ta có thể có các cách xác thực khác nhau cho dịch vụ đó tuỳ thuộc vào việc dịch vụ đó được sử dụng ở môi trường nào. WCF hỗ trợ rất nhiều các mô hình xác thực khác nhau:

- Anonymous caller (Người gọi vô danh)
- Username client credential
- Certificate client credential
- Windows (Kerberos và NT LanMan – NTML)

1.1.3 Các chuẩn và tính interoperability (làm việc liên môi trường)

Trong thế giới với nhiều nền tảng như hiện nay, các nền tảng tính toán/truyền thông phân tán cần phải làm việc liên thông với các công nghệ khác nhau mà các nhà cung cấp hỗ trợ. Cũng vì vậy mà bảo mật cũng cần phải làm việc liên môi trường (interoperable).

Để thực hiện các hệ thống bảo mật interoperable, các công ty sử dụng dịch vụ Web với một loạt các chuẩn khác nhau. Về bảo mật thì có thể kể ra một số chuẩn như sau: WS-Security, SOAP Message Security, WS-Trust, WS-SecureConversation, và WS-SecurityPolicy.

Với các dịch vụ WCF ta có thể sử dụng WSHttpBinding để hỗ trợ WS-Security 1.1 và WS-SecureConversation.

1.2 Các lĩnh vực bảo mật của WCF

Bảo mật trong WCF chia ra thành ba vùng chức năng: transfer security (bảo mật truyền thông), (access control) điều khiển truy nhập, và auditing (ghi vết).

1.2.1 Transfer security

Bảo mật truyền thông bao gồm ba chức năng chính: sự nhất quán, sự bảo mật, và sự xác thực. Sự nhất quán là khả năng phát hiện liệu bản tin có bị thay đổi hay không. Sự bảo mật là khả năng giữ cho bản tin không đọc được bởi những người không đủ thẩm quyền; điều này có được nhờ vào cryptography (mã mật – mật mã). Sự xác thực là khả năng xác minh được một định danh có thực đúng hay không. Kết hợp ba chức năng này cho ta đảm bảo rằng các bản tin được gửi đi một cách an toàn, đến đúng nơi cần đến.

1.2.2 Các chế độ bảo mật vận chuyển và bản tin

Có hai phương pháp chính dùng để thực hiện bảo mật truyền thông trong WCF là chế độ bảo mật ở tầng vận chuyển (transport security mode) và chế độ bảo mật ở bản tin (message security mode)

- *Transport security mode* sử dụng các giao thức ở tầng vận chuyển như HTTPS để đảm bảo bảo mật. Chế độ này có ưu điểm là được sử dụng rất nhiều ở các nền tảng khác nhau, và độ phức tạp tính toán ít hơn. Tuy vậy nhược điểm là chỉ đảm bảo bảo mật các bản tin từ điểm-tới-điểm.

- *Message security mode* sử dụng chuẩn WS-Security để đảm bảo bảo mật. Do bảo mật bản tin được áp dụng trực tiếp lên các bản tin SOAP và được chứa trong các vỏ (envelope) SOAP, cùng với dữ liệu của chương trình, nó có ưu điểm là không phụ thuộc vào giao thức vận chuyển, dễ mở rộng, đảm bảo bảo mật từ đầu cuối-tới-đầu cuối (thay vì điểm-tới-điểm). Nhược điểm của nó là chậm hơn so với chế độ transport security do nó phải làm việc với XML trong bản tin SOAP.

1.2.3 Điều khiển truy nhập

Điều khiển truy nhập còn được biết tới như là authorization (nhận thực). Authorization cho phép những người dùng khác nhau có các quyền khác nhau để xem dữ liệu. Trong WCF, các tính năng điều khiển truy nhập được cung cấp dựa vào sự tích hợp với CLR (common language runtime) thông qua lớp thuộc tính `PrincipalPermissionAttribute` và qua một loạt các hàm API.

1.2.4 Auditing

Auditing là quá trình ghi lại các sự kiện bảo mật vào hệ thống log của hệ điều hành Windows (Windows event log). Bạn có thể ghi lại các sự kiện có liên quan tới bảo mật như là xác thực lỗi hay thành công. Bạn có thể xem thêm trong bài How to: Audit Windows Communication Foundation Security Events

2 Lập trình với bảo mật trong WCF

Khi lập trình với tính năng bảo mật trong WCF bạn cần thực hiện 3 bước cơ bản sau: đặt chế độ bảo mật, kiểu xác thực client (client credential type), và các giá trị credentials cho dịch vụ. Cũng như các phần khác trong WCF, bạn hoàn toàn có thể thực hiện các bước này thông qua mã lập trình hoặc là sử dụng tập tin cấu hình.

2.1 Đặt chế độ bảo mật

Các bước để đặt chế độ bảo mật như sau:

1. Chọn một trong các binding phù hợp với các yêu cầu của ứng dụng của bạn. Để biết về các binding nào có thể chọn, bạn nên xem lại bài 3 về address với bindings. Mặc định thì hầu hết các binding đều có tính năng bảo mật. Tuỳ thuộc vào bạn chọn binding nào, bạn sẽ làm việc với giao thức vận chuyển tương ứng với binding đó. Ví dụ khi chọn binding là WSHttpBinding thì sẽ sử dụng giao thức vận chuyển là HTTP; còn nếu sử dụng binding là NetTcpBinding thì giao thức là TCP.

2. Chọn một trong các chế độ bảo mật cho binding đã chọn. Tuỳ theo bạn chọn binding nào thì sẽ có các chế độ bảo mật tương ứng. Ví dụ, binding WSDualHttpBinding không hỗ trợ bảo mật vận chuyển (transport security), tương tự như vậy các binding MsmqIntegrationBinding và NetNamedPipeBinding không hỗ trợ bảo mật bản tin (message security). Bảng dưới đây liệt kê các binding có thể sử dụng và các chế độ bảo mật binding đó hỗ trợ. Chú ý: trong ngoặc kép là chế độ bảo mật mặc định.

Binding	Bảo mật
BasicHttpBinding	(none), Transport, Message
WSHttpBinding	Transport, (Message), Mixed
WSDualHttpBinding	(Message)
WSFederationHttpBinding	(Message)
NetTcpBinding	(Transport), Message
NetNamedPipeBinding	(Transport)
NetMsmqBinding	Message, (Transport), Both
NetPeerTcpBinding	(Transport)
MsmqIntegrationBinding	(Transport)

3. Nếu bạn quyết định sử dụng chế độ bảo mật vận chuyển (transport security) cho HTTP (tức là sử dụng giao thức HTTPS), bạn cần phải cấu hình cho host một chứng nhận SSL (SSL certificate) và cho phép SSL ở một cổng nào đó. Để có thêm thông tin bạn có thể xem thêm ở trang MSDN của Microsoft, HTTP Transport Security.

Ví dụ sử dụng tập tin cấu hình để cấu hình cho chế độ bảo mật

```
<bindings>
  <wsHttpBinding>
    <binding name="wsHttpTransportSecurity">
      <security mode="Transport">
        <transport clientCredentialType="Windows" />
      </security>
    </binding>
  </wsHttpBinding>
  <wsHttpBinding>
    <binding name="wsHttpMessageSecurity">
      <security mode="Message">
        <message clientCredentialType="UserName" />
      </security>
    </binding>
```

```
        </wsHttpBinding>
        <basicHttpBinding>
          <binding name="basicHttp">
            <security mode="TransportWithMessageCredential">
              <transport />
              <message clientCredentialType="UserName" />
            </security>
          </binding>
        </basicHttpBinding>
      </bindings>
```

Ví dụ sử dụng mã lập trình để thiết lập chế độ bảo mật:

```
      WSHttpBinding messageSecureBinding = new WSHttpBinding();
      messageSecureBinding.Name = "messageSecureBinding";
      messageSecureBinding.Security.Mode = SecurityMode.Message;
      messageSecureBinding.Security.Message.ClientCredentialType =
MessageCredentialType.Windows;

      NetTcpBinding transportSecureBinding = new NetTcpBinding();
      transportSecureBinding.Name = "transportSecureBinding";
      transportSecureBinding.Security.Mode = SecurityMode.Transport;
      transportSecureBinding.Security.Transport.ClientCredentialType =
TcpClientCredentialType.Windows;

      BasicHttpBinding mixedSecureBinding = new BasicHttpBinding();
      mixedSecureBinding.Name = "mixedSecureBinding";
      mixedSecureBinding.Security.Mode =
BasicHttpSecurityMode.TransportWithMessageCredential;
      mixedSecureBinding.Security.Message.ClientCredentialType =
                          BasicHttpMessageCredentialType.UserName;
```

2.2 Chọn kiểu xác thực client

Các kiểu xác thực client được hỗ trợ bởi WCF như sau:

- Windows
- Certificate
- Digest
- Basic
- UserName
- NTLM
- IssuedToken

Về cách thiết lập kiểu xác thực client, các bạn có thể xem ở ví dụ trên.

2.3 Thiết lập các giá trị credentials cho dịch vụ

Sau khi đã chọn kiểu xác thực client, bạn cần phải thiết lập giá trị credential thực cho dịch vụ và client sử dụng. Ở phía dịch vụ, các credential được thiết lập sử dụng lớp `ServiceCredentials` và được trả về từ thuộc tính `Credentials` của lớp `ServiceHostBase`.

Ví dụ sau đây thiết lập một chứng nhận cho credential phía dịch vụ:

```
      // Create the binding for an endpoint.
      NetTcpBinding b = new NetTcpBinding();
```

```
        b.Security.Mode = SecurityMode.Message;

        // Create the ServiceHost for a calculator.
        Uri baseUri = new Uri("net.tcp://MachineName/tcpBase");
        Uri[] baseAddresses = new Uri[] { baseUri };
        ServiceHost sh = new ServiceHost(typeof(Calculator), baseAddresses);

        // Add an endpoint using the binding and a new address.
        Type c = typeof(ICalculator);
        sh.AddServiceEndpoint(c, b, "MyEndpoint");

        // Set a certificate as the credential for the service.
        sh.Credentials.ServiceCertificate.SetCertificate(
            StoreLocation.LocalMachine,
            StoreName.My,
            X509FindType.FindBySubjectName,
            "client.com");
        try
        {
            sh.Open();
            Console.WriteLine("Listening....");
            Console.ReadLine();
            sh.Close();
        }
        catch (CommunicationException ce)
        {
            Console.WriteLine("A commmunication error occurred: {0}", ce.Message);
            Console.WriteLine();
        }
        catch (System.Exception exc)
        {
            Console.WriteLine("An unforseen error occurred: {0}", exc.Message);
            Console.ReadLine();
        }
```

2.4 Thiết lập các giá trị credentials cho client

Ở phía client, các giá trị credentials được thiết lập thông quá lớp `ClientCredentials` và được trả về qua thuộc tính `ClientCredentials` của lớp `ClientBase`.

Ví dụ sau thiết lập một chứng nhận như là một credential phía client sử dụng giao thức TCP.

```
        // Create a NetTcpBinding and set its security properties. The
        // security mode is Message, and the client must be authenticated with
        // Windows. Therefore the client must be on the same Windows domain.
        NetTcpBinding b = new NetTcpBinding();
        b.Security.Mode = SecurityMode.Message;
        b.Security.Message.ClientCredentialType = MessageCredentialType.Windows;

        // Set a Type variable for use when constructing the endpoint.
        Type c = typeof(ICalculator);

        // Create a base address for the service.
        Uri tcpBaseAddress =
            new Uri("net.tcp://machineName.Domain.Contoso.com:8036/serviceName");
        // The base address is in an array of URI objects.
        Uri[] baseAddresses = new Uri[] { tcpBaseAddress };
        // Create the ServiceHost with type and base addresses.
        ServiceHost sh = new ServiceHost(typeof(CalculatorClient), baseAddresses);
```

```
// Add an endpoint to the service using the service type and binding.
sh.AddServiceEndpoint(c, b, "");
sh.Open();
string address = sh.Description.Endpoints[0].ListenUri.AbsoluteUri;
Console.WriteLine("Listening @ {0}", address);
Console.WriteLine("Press enter to close the service");

Console.ReadLine();
```

3 Xây dựng ví dụ với bảo mật trong WCF

Trong ví dụ này chúng ta sẽ xây dựng một ví dụ sử dụng Windows Authentication qua binding wsHttpBinding có sử dụng chế độ bảo mật vận chuyển (transport security). Mô hình này phù hợp với môi trường intranet trong doanh nghiệp khi ta có một domain controller.

Các bước để xây dựng như sau:

- Bước 1 – Tạo dịch vụ WCF
- Bước 2 – Thiết lập cấu hình cho dịch vụ WCF sử dụng wsHttpBinding với Windows Authentication và Message Security (bảo mật bản tin)
- Bước 3 – Tạo client
- Bước 4 – Thêm tham chiếu dịch vụ cho client
- Bước 5 – Chạy thử client và dịch vụ

3.1 Tạo dịch vụ WCF

1. Mở Visual Studio, chọn menu **File** -> **New Web Site**

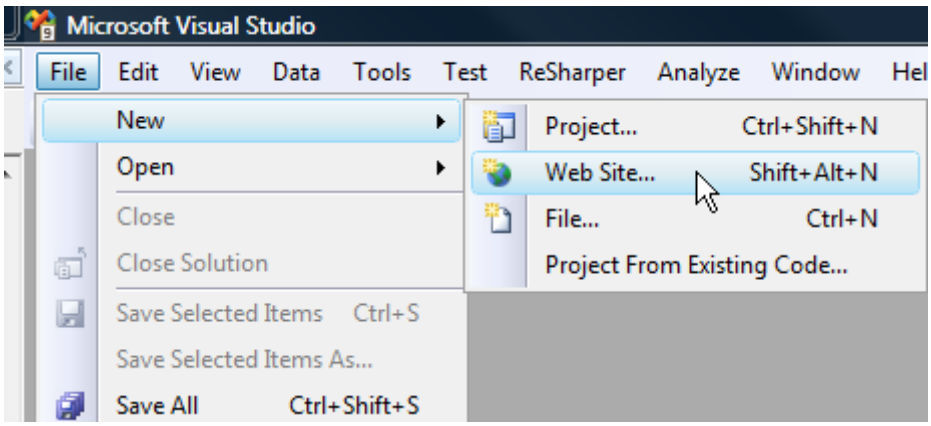

Figure 1 Tạo một Web Site mới

2. Trong hộp thoại **New Web Site**, ở phần **Templates**, chọn **WCF Service**, chọn **Location** là Http

3. Đặt địa chỉ trang web là https://localhost/SecureContact, chọn OK

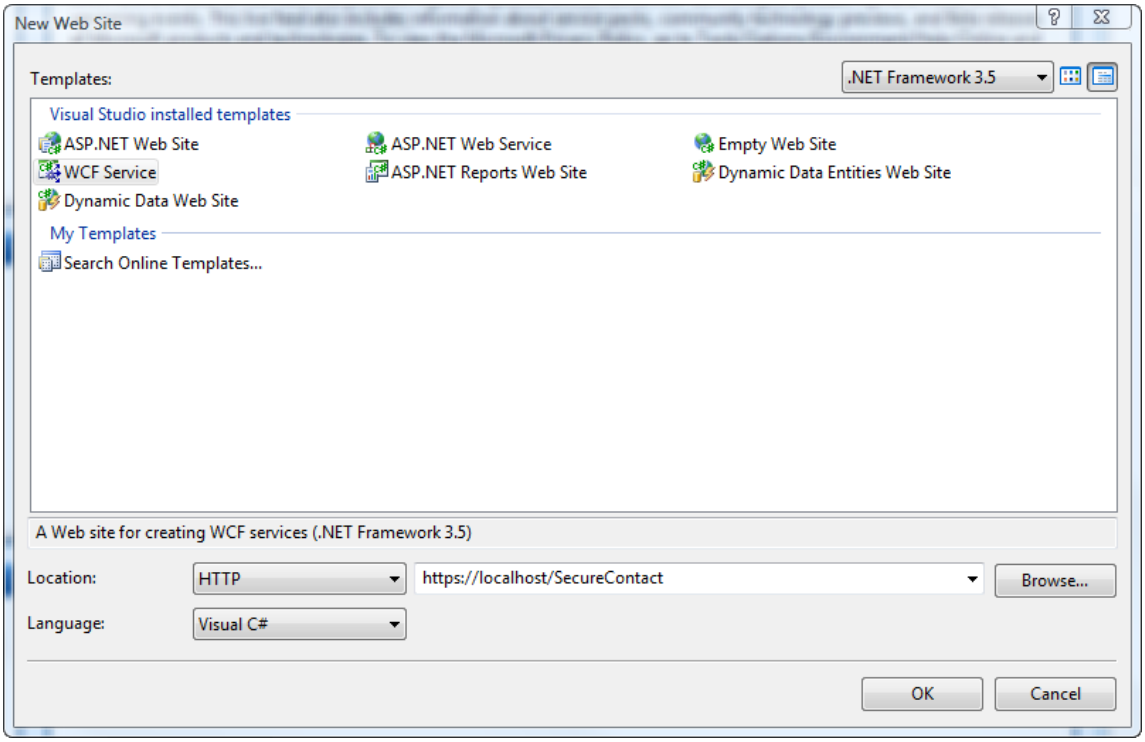

Figure 2 Tạo Web site với template là WCF Service, giao thức HTTPS

3.2 Thêm chức năng cho dịch vụ

1. Click chuột phải vào project **SecureContact**, chọn Menu **Add New Item**, sau đó chọn thêm vào **WCF Service**, đặt tên là **ContactService**.svc

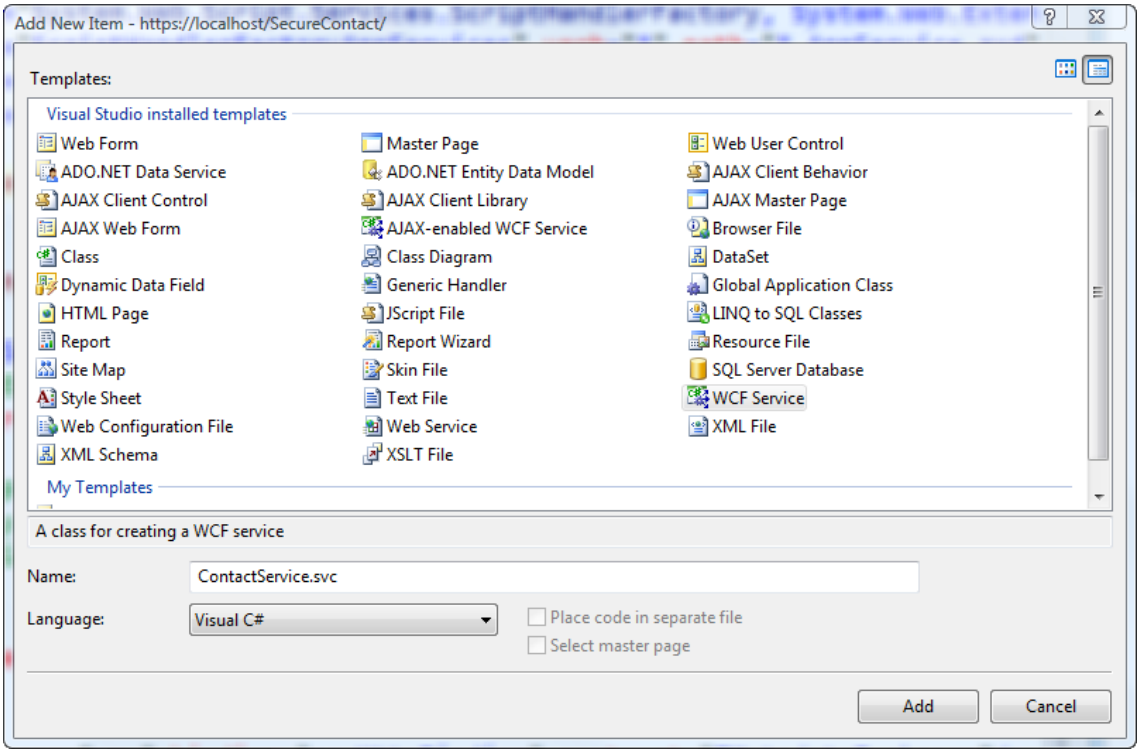

Figure 3 Thêm mới một dịch vụ WCF

2. Thêm vào đoạn mã sau để định nghĩa interface

```
        using System.Runtime.Serialization;
        using System.ServiceModel;

        /// <summary>
        /// Định nghĩa thông tin về một nhân viên
        /// </summary>
        [DataContract]
        public class Person
        {
            [DataMember]
            public int PersonId;
            [DataMember]
            public string FullName;
            [DataMember]
            public int Age;
        }

        [ServiceContract]
        public interface IContactService
        {
            [OperationContract]
            bool HasPerson(int personId);

            /// <summary>
            /// Lấy về thông tin của một nhân viên.
            /// Nếu có nhân viên với ID trùng với personId
            /// thì trả về thông tin nhân viên đó.
            /// Trong trường hợp không có nhân viên phù hợp thì báo lỗi bằng
bản tin lỗi
            /// </summary>
            /// <param name="personId">ID của nhân viên</param>
            /// <returns>Thông tin của nhân viên</returns>
            [OperationContract]
            Person GetPerson(int personId);

            [OperationContract]
            Person[] GetAll();

            [OperationContract]
            void AddPerson(Person person);

            /// <summary>
            /// Sửa thông tin của một nhân viên.
            /// Nếu có nhân viên với ID trùng với personId
            /// thì cập nhật thông tin nhân viên đó.
            /// Trong trường hợp không có nhân viên phù hợp thì báo lỗi bằng
bản tin lỗi
            /// </summary>
            /// <param name="personId">ID của nhân viên</param>
            /// <param name="person">Thông tin mới về nhân viên</param>
            [OperationContract]
            void EditPerson(int personId, Person person);

            /// <summary>
            /// Xoá thông tin một nhân viên
            /// Nếu có nhân viên với ID trùng với personId
            /// thì xoá thông tin nhân viên đó.
            /// Trong trường hợp không có nhân viên phù hợp thì báo lỗi bằng
bản tin lỗi
            /// </summary>
            /// <param name="personId">ID của nhân viên</param>
```

```
        [OperationContract]
        void DeletePerson(int personId);
    }
```

3. Thêm vào đoạn mã sau để định nghĩa phần cài đặt dịch vụ

```
    using System.Collections.Generic;
    using System.Linq;

    public class ContactService : IContactService
    {
        private readonly List<Person> personCollection;
        public ContactService()
        {
            personCollection = new List<Person>
            {
             new Person { PersonId = 0, FullName = "Trần Văn Anh", Age =
30 },
             new Person { PersonId = 1, FullName = "Lý Đào", Age = 35 },
             new Person { PersonId = 2, FullName = "Hoàng Văn Giáp", Age
= 28 },
             new Person { PersonId = 3, FullName = "Nguyễn Quang Cường",
Age = 25 },
             new Person { PersonId = 4, FullName = "Lê Thị Bảo", Age =
31 },
            };
        }

        public bool HasPerson(int personId)
        {
            return this.personCollection.Any(x => x.PersonId ==
personId);
        }

        public Person GetPerson(int personId)
        {
            if (!this.HasPerson(personId))
            {
                return null;
            }

            return
                this.personCollection.FirstOrDefault(x => x.PersonId ==
personId);
        }

        public void AddPerson(Person person)
        {
            this.personCollection.Add(person);
        }

        public Person[] GetAll()
        {
            return this.personCollection.ToArray();
        }

        public void EditPerson(int personId, Person person)
        {
            Person p = this.GetPerson(personId);
            p.FullName = person.FullName;
            p.Age = person.Age;
        }
```

```
        public void DeletePerson(int personId)
        {
            Person p = this.GetPerson(personId);
            this.personCollection.Remove(p);
        }
    }
```

3.3 Thiết lập dịch vụ WCF sử dụng wsHttpBinding với Windows Authentication và Message Security

Để cấu hình cho dịch vụ WCF chấp nhận giao thức HTTPS, ta cần thiết lập tập tin cấu hình (web.config) như sau:

```
    <system.serviceModel>
      <services>
        <service behaviorConfiguration="ContactServiceBehavior"
name="ContactService">
          <host>
            <baseAddresses>
              <add baseAddress="https://localhost/SecureContact"/>
            </baseAddresses>
          </host>
            <endpoint address="" binding="wsHttpBinding"
contract="IContactService"
                      bindingConfiguration="wsHttpSecureBinding">
              <identity>
                <dns value="localhost"/>
              </identity>
            </endpoint>
            <endpoint address="mex" binding="mexHttpsBinding"
contract="IMetadataExchange"/>
        </service>
      </services>
      <behaviors>
        <serviceBehaviors>
          <behavior name="ContactServiceBehavior">
            <serviceMetadata httpsGetEnabled="true"/>
            <serviceDebug includeExceptionDetailInFaults="false"/>
          </behavior>
        </serviceBehaviors>
      </behaviors>
      <bindings>
        <wsHttpBinding>
          <binding name="wsHttpSecureBinding">
            <security mode="Transport">
              <transport clientCredentialType="Windows"/>
            </security>
          </binding>
        </wsHttpBinding>
      </bindings>
    </system.serviceModel>
```

Đến thời điểm này bạn đã có một dịch vụ hoàn chỉnh. Mở trình duyệt Web và truy nhập tới địa chỉ https://localhost/SecureContact/ContactService.svc bạn sẽ thấy màn hình sau:

ContactService Service

You have created a service.

To test this service, you will need to create a client and use it to call the service. You can do this using the svcutil.exe tool from the command line with the following syntax:

```
svcutil.exe https://huuhoa-laptop/SecureContact/ContactService.svc?wsdl
```

This will generate a configuration file and a code file that contains the client class. Add the two files to your client application and use the generated client class to call the Service. For example:

C#

```
class Test
{
    static void Main()
    {
        ContactServiceClient client = new ContactServiceClient();

        // Use the 'client' variable to call operations on the service.

        // Always close the client.
        client.Close();
    }
}
```

Figure 4 Truy nhập bằng trình duyệt Web tới địa chỉ của dịch vụ

3.4 Tạo ứng dụng client

1. Click chuột phải vào solution hiện tại, chọn **Add** -> **New Project**
2. Trong hộp thoại **Add New Project**, chọn **Windows Forms Application**
3. Đặt tên project là **SecureContactClient**.

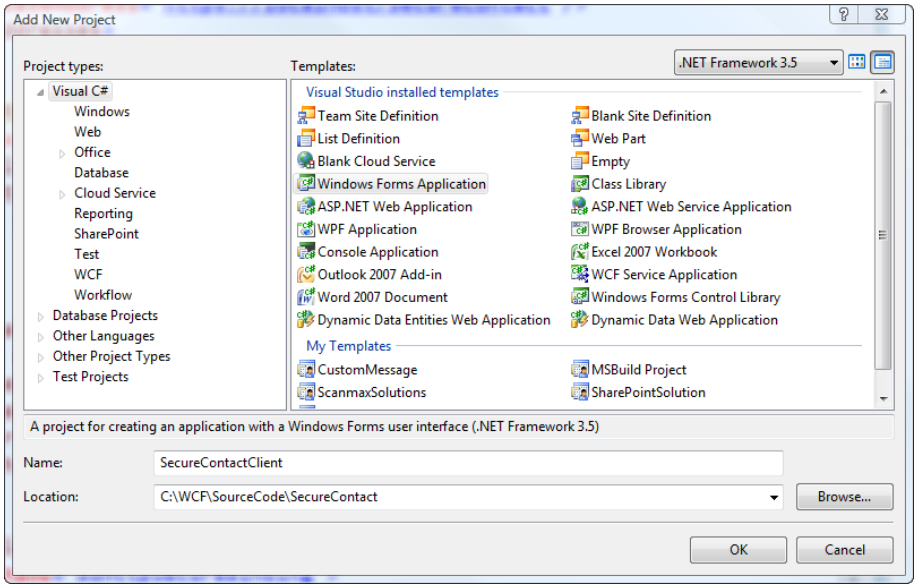

Figure 5 Tạo client

3.5 Thêm tham chiếu tới dịch vụ WCF cho client

1. Click chuột phải vào client project, chọn **Add Service Reference**
2. Trong hộp thoại **Add Service Reference**, đưa vào URL tới dịch vụ của bạn, https://localhost/SecureContact/ContactService.svc
3. Trong trường **Namespace**, nhập vào **SecureContactService**, click OK

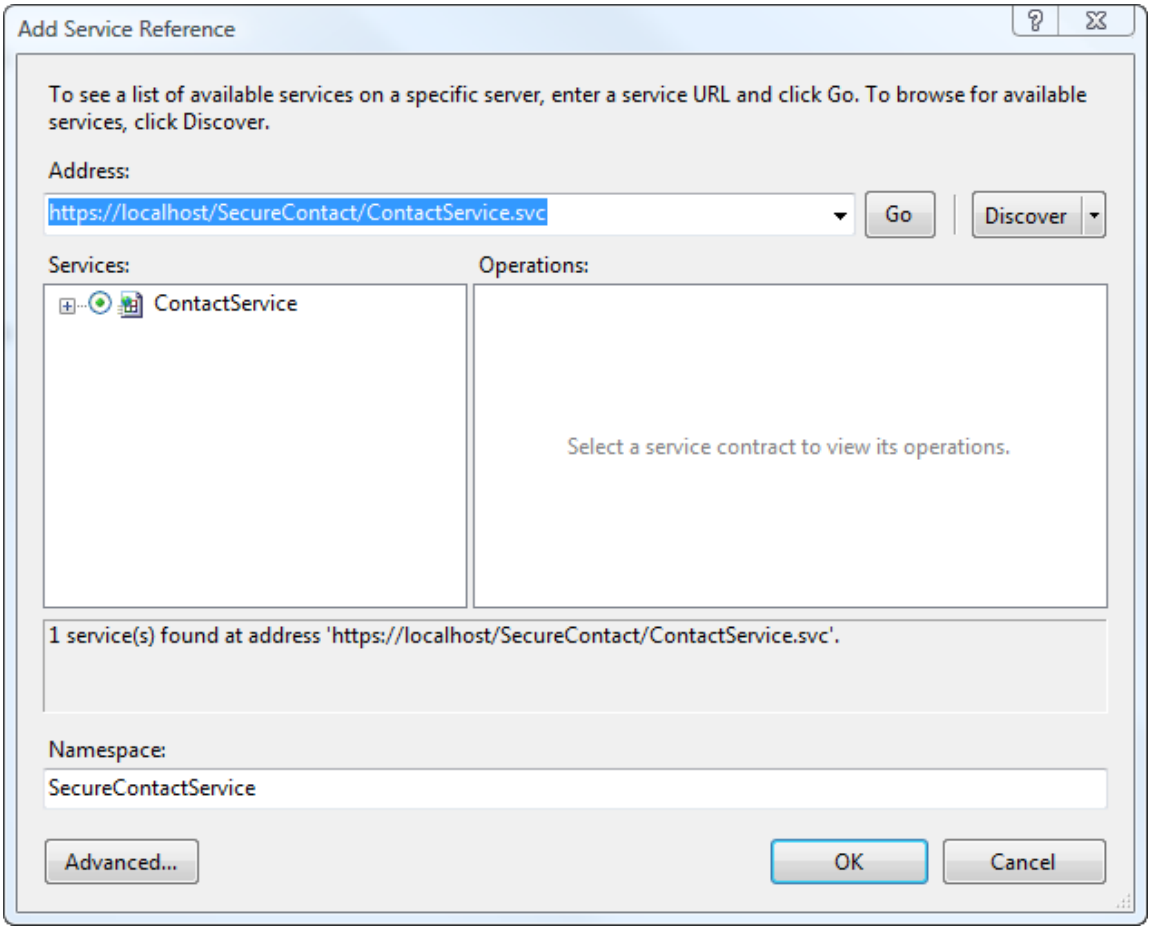

Figure 6 Thêm tham chiếu dịch vụ cho Client

3.6 Thêm chức năng cho client

Thêm vào trong form mặc định 2 Button và 1 DataGridView (xem chi tiết trong source code kèm theo). Viết mã xử lý các sự kiện bấm nút:

```
private void buttonGetInformation_Click(object sender, EventArgs e)
{
    var client = new ContactServiceClient();
    try
    {
        Person p = client.GetPerson(Convert.ToInt32(this.textBoxID.Text));
        this.dataGrid.DataSource = new[] { p };
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error");
    }
}
```

```
private void buttonGetAll_Click(object sender, EventArgs e)
{
    var client = new ContactServiceClient();
    this.dataGrid.DataSource = client.GetAll();
}
```

Chạy ứng dụng Client và nhất nút Get All ta sẽ có kết quả sau:

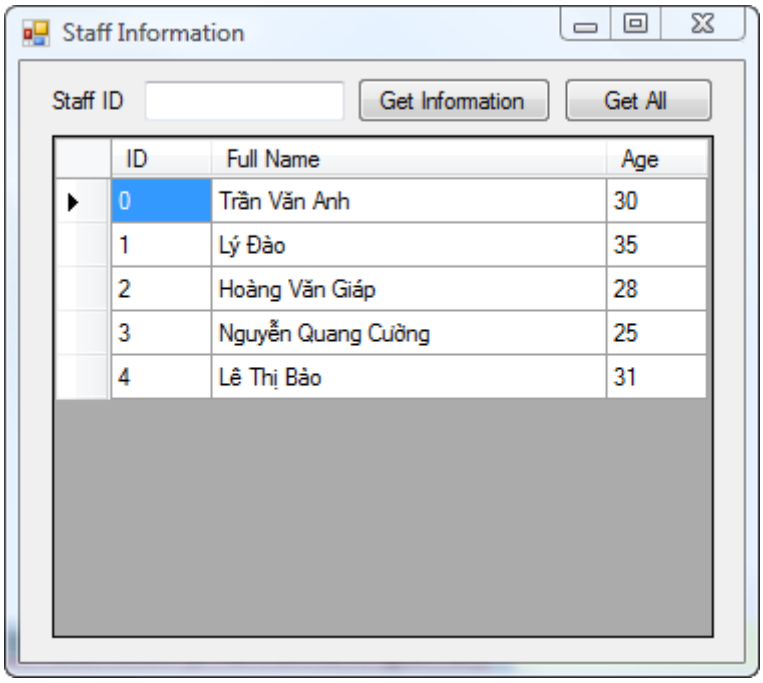

Figure 7 Kết quả chạy client

4 Câu hỏi ôn tập

1. Binding MsmqIntegrationBinding có hỗ trợ chế độ bảo mật bản tin không?

Trả lời: Không, Binding MsmqIntegrationBinding chỉ hỗ trợ chế độ bảo mật vận chuyển.

5 Tài liệu tham khảo

1. Security Overview (URL: http://msdn.microsoft.com/en-us/library/ms735093.aspx)
2. Programming Security (URL: http://msdn.microsoft.com/en-us/library/ms731925.aspx)
3. Fundametals of WCF Security (URL: http://www.devx.com/codemag/Article/33342/1954)
4. WCF Security Resources (URL: http://blogs.msdn.com/jmeier/archive/2008/05/23/wcf-security-resources.aspx)
5. WCF Security Learning Guide (URL: http://www.theserverside.net/tt/articles/showarticle.tss?id=WCFSecurityLearningGuide)