CHƯƠNG VIII:
GIAO TIẾP MẠNG TRONG SILVERLIGHT

1 Giao tiếp HTTP và bảo mật trong Silverlight

Silverlight cho phép HTTP / HTTPS giao tiếp với các dịch vụ lưu trữ trên máy chủ web với cả hai trường hợp bên trong và bên ngoài tên miền của bạn được lưu trữ trên cơ sở ứng dụng Silverlight. Chủ đề này thảo luận về một số kịch bản HTTP giao tiếp và làm thế nào để bạn có thể kích hoạt các kịch bản.

1.1 Mặc định hỗ trợ giao thức HTTP

Có một số khả năng cơ bản cho tất cả những kiểu giao tiếp HTTP/HTTPS của Silverlight

- Trong cùng tên miền thì mọi triệu gọi luôn được chấp nhận
- Khi các Web service được thiếp lập đúng trên máy chủ lưu trữ thì việc truy cập từ tên miền khác tới là được hỗ trợ.
- Tất cả các giao tiếp là không đồng bộ
- Chỉ hỗ trợ phương thức GET và POST
- Hầu hết các triệu gọi tiêu chuẩn và các tùy chỉnh Header đã được hỗ trợ. (Header phải được cho phép trong file cross-domain policy cho các triệu gọi tới từ domain khác.)
- Chỉ có mã trạng thái 200-“OK” và 400-“Không tìm thấy”

1.2 Kịch bản giao tiếp HTTP

Silverlight hỗ trợ một số kịch bản có sử dụng giao thức HTTP / HTTPS. Mặc dù có nhiều cách thức và công nghệ có thể được sử dụng để thực hiện các triệu gọi giao thức HTTP, bảng sau mô tả phương pháp tiếp cận đối với một số tình huống giao tiếp có thể xảy ra. Những phương pháp tiếp cận sẽ được thảo luận chi tiết hơn sau này trong tài liệu này

Kịch bản	**Phương pháp tiếp cận nên dùng**
Download và Upload tài nguyên trong cùng một domain	Sử dụng lớp **WebClient** để thao tác
Triệu gọi Web service trên giao	Sử dụng lớp **WebClient** hoặc những lớp **HttpWebRequest/HttpWebResponse** để thao tác

thức HTTP trong cùng một domain	
Triệu gọi SOAP, WCF, hoặc ASP.NET AJAX Web services được lưu trữ trong cùng một domain.	Triệu gọi thông qua proxy cho Web service. Nếu bạn không muốn xử dụng proxy thì sử dụng các lớp
Gọi Web service trên tên miền khác.	Chắc chắn là file cross-domain policy nằm ở thư mục gốc của domain(mặc định là wwwroot). Sử dụng proxy, lớp WebClient hoặc các lớp HttpWebRequest/HttpWebResponse.
Thiết lập header cho triệu gọi cross-domain	- Đảm bảo là header cho phép thông qua tập tin cross-domain policy - Đối với các yêu cầu trên các dữ liệu upload, sử dụng lớp WebClient. Thiết lập các tập hợp các header muốn có. - Đối với các kịch bản sử dụng lớp HttpWebRequest. Thiết lập thuộc tính của nó với tập hợp những header mong muốn.

1.3 Giao tiếp trong cùng một domain

Mặc định Silverlight hỗ trợ triệu goi tới Web service trong cùng một domain. Trong cùng domain có tức là những triệu gọi phải trong cùng một sub domain, giao thức, và cổng. Điều này là vì lý do bảo mật, để ngăn chặn sự truy cập trái phép tới Web service.

Minh hoạ về triệu gọi Web service sau đây về sự cho phép và không cho phép sự truy cập của ứng dụng Silverlight khi sử dụng những thiết lập mặc định:

1.4 Giao tiếp Cross-domain

Bạn có thể cho phép ứng dụng Silverlight ở domain khác triệu gọi Web service của bạn thông qua việc thiết lập chính xác tập tin cross-domain policy đặt tại gốc của domain (mặc định là thư mục wwwroot).Silverlight hỗ trợ 2 loại của tập tin cross-domain policy;

- Silverlight Cross-Domain Policy (clientaccesspolicy.xml)
- Một nhóm của Flash Cross-Domain Policy(crossdomain.xml)

Ví dụ sau đây minh hoạ về giao tiếp cross-domain thông qua việc sử dụng tập tin Cross-Domain Policy.

Nói chung, khi một ứng dụng Silverlight phát hiện rằng triệu gọi của nó là cross-domain, trước tiên nó sẽ tìm tập tin Silverlight Cross-Domain (clientaccesspolicy.xml) tại vị trí gốc của Web service. Nếu triệu gọi này trả về mã lỗi 404-Không tìm thấy hoặc các lỗi khác, ứng dụng sau đó sẽ tìm tập tin Flash Cross-Domain (crossdomain.xml) tại vị trí gốc domain.

Bảng sau liệt kê các triệu gọi và URIs nơi Silverlight dựa trên ứng dụng sẽ tìm tập tin cross-domain

Request URI	Cross-Domain Policy File Location
http://contoso.com/services/data	http://contoso.com/clientaccesspolicy.xml
http://sales.contoso.com/services/data	http://sales.contoso.com/clientaccesspolicy.xml
http://contoso.com:8080/services/data	http://contoso.com:8080/clientaccesspolicy.com

Không thể dùng GET request bao gồm header khi sử dụng cross-domain. Chỉ sử dụng được request header với POST request khi nó được chỉ ra ở tập tin cross-domain policy.

- Lưu ý bảo mật Web service được dùng cho Silverlight với triệu gọi cross-domain

nghiên cứu kỹ lưỡng về bảo mật trước khi bạn cho phép các kết nối từ Silverlight truy cập vào Web service sử dụng tập tin cross-domain. Bất cứ khi nào bạn đặt tệp tin cross-domain policy ở nơi mà bạn cấu hình máy chủ lưu trữ web service thì nên thiết lập vô hiệu hoá bộ nhớ đệm của trình duyệt. Điều này cho phép bạn dễ dàng cập nhật các tập tin hoặc hạn chế quyền truy cập vào các Web service của bạn nếu cần thiết.

Ngoài ra, tất cả các request của Silverlight được gửi với các tập tin cookie và xác thực. Điều này có nghĩa là nếu bạn có Web service cho phép người sử dụng truy cập thông tin cá nhân, bạn nên lưu trữ nó với tên miền khác nhau hơn là trên Web service tiếp xúc với bên thứ ba. Ví dụ, bạn có một kho lưu trữ trên máy chủ web tại http://contoso.com. Trang web của bạn cho phép khách hàng lưu trữ thông tin thanh toán bao gồm số thẻ tín dụng. Bạn không nên làm một Web service để trả lại sản phẩm tồn kho của bên khách hàng thứ ba sử dụng Silverlight tại cùng một tên miền. Bởi vì các tập tin cookie và xác thực được gửi đi với mỗi request, nếu bạn lưu trữ các Web service trên cùng một tên miền, thì các bên thứ ba sử dụng dịch vụ của bạn có thể truy cập vào dữ liệu thanh toán cá nhân của khách hang của bạn. Trong ví dụ này, công khai giao tiếp của Web service của bạn có thể được lưu trữ trên máy chủ an toàn tại http://services.contoso.com, bởi vì đây là một tên miền khác. Bạn phải cẩn thận xem xét những đối tượng tiếp xúc với Web service, và những Web service khác đang nằm tại domain đó. Ngoài ra, bạn nên luôn luôn giữ tập tin cross-domain policy càng hạn chế càng tốt.

- Ví dụ về tập tin Cross-Domain Policy

Tập tin cross-domain policy của Silverlight là một tập tin XML có định dạng đơn giản. Ví dụ dưới đây chỉ ra tập tin cross-domain policy của Silverlight cho phép requests tới Web service liên kết tới đường dẫn “services” của domain. Tập tin cross-domain policy cũng chỉ ra loại nội dung header là SOAPAction.

```
<?xml version="1.0" encoding="utf-8"?>
  <access-policy>
    <cross-domain-access>
```

```
    <policy >
      <allow-from http-request-headers="SOAPAction">
        <domain uri="*"/>
      </allow-from>
      <grant-to>
        <resource path="/services/" include-subpaths="true"/>
      </grant-to>
    </policy>
  </cross-domain-access>
</access-policy>
```

- Tập tin Flash Cross-Domain Policy

Silverlight hỗ trợ Flash Cross-Domain (crossdomain.xml). Silverlight hỗ trợ thẻ <allow-access-from> với các thuộc tính sau:

+ **domain** với thuộc tính giá trị "*", có nghĩa là tất cả các domain đều có thể triệu gọi.

+ **secure** chấp nhận giá trị là **true** hoặc **false** là sử dụng tương ứng HTTPS hay HTTP.

+ **headers** nhận các giá trị header.

Sau đây là ví dụ của một tập tin Flash crossdomain.xml với Silverlight.

```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM
   "http://www.macromedia.com/xml/dtds/cross-domain-policy.dtd">
<cross-domain-policy>
 <allow-access-from domain="*"/ headers="SOAPAction"
 secure="true">
</cross-domain-policy>
```

1.5 Thiết lập triệu gọi HTTP

Tùy thuộc vào kịch bản, bạn có thể thực hiện các cuộc gọi HTTP sử dụng lớp clien-side proxy hoặc xây dựng triệu gọi của chính bạn. Dưới đây là những phần mô tả các phương pháp khác nhau để thực hiện các cuộc gọi trên mạng bằng cách sử dụng HTTP

- Sử dụng lớp Proxy

  Bạn có thể tạo ra một lớp proxy từ Web service metadata và sử dụng proxy để kết nối tới Web service từ ứng dụng Silverlight của bạn. Silverlight sử dụng Windows Communication Foundation(WCF) để tạo ra proxy và gửi message SOAP 1.1 thông qua giao thức HTTP. Nếu bạn sử dụng Visual Studio, bạn chuột phải vào dự án Silverlight và chọn **Add Service Reference** sẽ tự động tạo proxy cho bạn. Proxy sẽ tạo message và xử lý giao tiếp mạng cho bạn.

- Tạo request HTTP

  + Nếu bạn muốn tạo triệu gọi HTTP của bạn, bạn có thể sử dụng những lớp được cung cấp trong namespace **System.Net**

  + WebClient

  + **HttpWebRequest** và **HttpWebResponse**

  Những lớp này cho phép bạn tạo request GET hoặc POST và cho phép request header trong một vài trường hợp. Thêm nữa, bạn có thể thiết lập những lớp này để tăng cường download với GET request

- Lớp **WebClient**

  Lớp WebClient cung cấp sự kiện đơn giản, dựa trên mô hình cho phép bạn tải xuống và tải lên suối và strings. Các WebClient là một lựa chọn tốt nếu bạn không muốn sử dụng một proxy lớp học. Nói chung, lớp này là dễ sử dụng, nhưng cung cấp ít hơn các tùy chọn để tùy chỉnh các message được gửi qua mạng.

  Bảng sau đây tóm tắt các header hỗ trợ cho các lớp **WebClient**.

Giao thức HTTP	Gọi trong cùng domain	Gọi Cross-Domain
GET	Header luôn được chấp nhận	Request header không được chấp nhận
POST	Header luôn được chấp nhận	Request headers chấp nhận thông qua tập tin cross-domain policy.

Để tạo POST request với **WebClient** và tải lên tập tin bất kỳ hoặc những chuỗi, bạn sử dụng một trong những phương thức sau đây.

+ WebClient..::.OpenWriteAsync

+ WebClient..::.UploadStringAsync

Bạn có thể xét header cho POST request thông qua thuộc tính WebClient..::.Headers . Request header phải được cho phép trong tập tin cross-domain policy.

Đoạn mã ví dụ dưới đây tạo POST request sử dụng WebClient

```
    // Tạo một instance của WebClient.
WebClient client = new WebClient();
public Page()
{
```

```
            InitializeComponent();

             // Đăng ký hóng sự kiện UploadStringCompleted.
            client.UploadStringCompleted +=
                new UploadStringCompletedEventHandler(client_UploadStringCompleted);
        }
```

```
        private void Button_Click(object sender, RoutedEventArgs e)
        {
             // Tạo request.
            string postRequest = "<entry xmlns='http://www.w3.org/2005/Atom'>"
            + "<title type='text'>New Restaurant</title>"
            + "<content type='xhtml'>"
            + "  <div xmlns='http://www.w3.org/1999/xhtml'>"
            + "   <p>There is a new Thai restaurant in town!</p>"
            + "   <p>I ate there last night and it was <b>fabulous</b>.</p>"
            + "   <p>Make sure and check it out!</p>"
            + "  </div>"
            + " </content>"
            + " <author>"
            + "   <name>Pilar Ackerman</name>"
            + "  <email>packerman@contoso.com</email>"
            + " </author>"
            + "</entry>";

              // Gửi request tới URL cụ thể.
            client.UploadStringAsync(new                Uri("http://blogs.contoso.com/post-
create?blogID=1234",
                UriKind.Absolute), postRequest);
        }

        // Hóng sự kiện UploadStringCompleted.
        void client_UploadStringCompleted(object sender,
          UploadStringCompletedEventArgs e)
        {
             // Hiển thị ra kết quả nhận được.
            if (e.Error != null)
                tb1.Text = e.Error.Message;
            else
                tb1.Text = e.Result;
    }
```

2 Các hạn chế trong truy cập mạng với silverlight

Phiên bản Silverlight 2 hỗ trợ hai kiểu chính sau để ứng dụng kết nối tới máy chủ từ xa:

- Lớp WebClient và những lớp HTTP có trong namespace System.Net – những lớp này sử dụng giao thức HTTP or HTTPS cho giao tiếp mạng.
- Những lớp Sockets trong namespace System.Net.Sockets – những lớp này cung cấp interface ở mức độ thấp của các socket nó có thể được sử dụng rộng một cách rộng rãi trong giao tiếp mạng.

Ở cả hai trường hợp trên, cần phải cung cấp bảo mật và ngăn chặn ứng dụng Silverlight kết nối tới từ các kết nối không được phép. Những mối đe doạ tấn công từ mạng bao gồm:

- Tấn công từ trối dịch vụ (DoS) – Một số lượng lớn các máy từ xa được sử dụng nhằm mục tiêu tấn công vào một trang web để mục tiêu là không thể request dịch vụ hợp lệ.
- Tấn công thay đổi thông tin DNS – Sử dụng DNS để quản lý máy chủ và cập nhật lại tên máy chủ tới địa chỉ IP của nạn nhân, bởi vậy cho phép truy cập sẽ vào máy chủ khác hơn là vào máy chủ thực sự của website đó.
- Reverse tunnel attack – Use a remote client's outgoing connection as a back tunnel to the client's private network.

Các chính sách bảo mật hệ thống kết hợp trong Silverlight được thiết kế để ngăn chặn các mối đe dọa mạng. Ngoài ra, hệ thống chính sách cũng nhằm mục đích cung cấp cho các quản trị viên kiểm soát nhiều hơn các nguồn lực từ xa mà khách hàng được phép để kết nối đến.

Trước đây thiết kế cho các mạng có khả năng bổ sung giới hạn kết nối đến máy chủ của trang web hoặc web gốc. Điều này có nghĩa là trang web đó, một ứng dụng có thể chỉ giao tiếp lại cho triển khai các máy chủ, do đó, mạng lưới các ứng dụng được phép để kết nối đến máy chủ mà từ đó chúng đã được tải về.

Silverlight 2 bao hỗ trợ kết nối cross-domain cho phép một ứng dụng để truy cập vào các tài nguyên từ các địa điểm khác(sub domaint, …) hơn là với trang web gốc. Đây là một tính năng quan trọng để tạo điều kiện cho các ứng dụng Silverlight sử dụng các dịch vụ hiện có trên web. Các chính sách an ninh hệ thống trong Silverlight 2 runtime bây giờ, đòi hỏi một tập tin cross-domain policy được tải về trước khi sự kết nối được phép truy cập vào tài nguyên đó. Điều này ảnh hưởng đến chính sách bảo mật của hệ thống domain qua mạng truy cập cho WebClient HTTP và các lớp trong namespace System.Net.

Đối với các sockets, các chính sách an ninh hệ thống trong Silverlight 2 ảnh hưởng đến quyền truy cập của cả trang web gốc và cross-domain. Chính sách bảo mật là bắt buộc cho bất kỳ kết nối kiểu socket, thậm chí cả khi kết nối lại tới chính trang web đó. Điều

này khác nhau từ các hành vi ứng xử trong Silverlight 2 Beta 1, nơi socket được cho phép kết nối đã được cho phép từ site gốc.

Các nội dung trong phần này cung cấp thông tin chi tiết về việc làm thế nào để sử dụng trong hệ thống chính sách bảo mật Silverlight 2 và mô tả chính sách tập tin định dạng được hỗ trợ.

2.1 Khái niệm cơ bản về chính sách bảo mật hệ thống

Silverlight hỗ trợ hai loại tập tin quy định cách ứng sử bảo mật:

Tập tin chính sách của Adobe Flash - crossdomain.xml. Tập tin chính sách này chỉ có thể được sử dụng bởi WebClient và các lớp HTTP trong namespace System.Net. Qua tập tin chính sách Flash, cho phép truy cập vào tất cả các tên miền sẽ được sử dụng bởi các Silverlight 2.

Tập tin chính sách của Silverlight – clientaccesspolicy.xml. Tập tin có thể được sử dụng bởi lớp WebClient và các lớp HTTP trong namespace System.Net và cũng có thể cho các lớp sockets trong namespace System.Net.Sockets. Tập tin chính sách này định dạng khác với tập tin chính sách của Flash.

Trước khi cho phép kết nối với tài nguyên mạng, Silverlight 2 sẽ cố gắng tải về tập tin chính sách bảo mật từ tài nguyên mạng đó. Có hai phương pháp khác nhau được sử dụng để tải về các chính sách bảo mật mà phụ thuộc vào việc kết nối đã được request từ một WebClient hoặc HTTP hoặc các lớp được request kết nối từ các sockets.

Nếu đã được yêu cầu kết nối từ WebClient hoặc các lớp HTTP tới cross-domain site, Silverlight 2 cố gắng tải về các tập tin chính sách bảo mật bằng cách sử dụng giao thức HTTP. Silverlight 2 đầu tiên cố gắng tải về tập tin chính sách Silverlight với tên "clientaccesspolicy.xml" ở gốc của site mục tiêu bằng cách sử dụng giao thức HTTP. Nếu tập tin chính sách của Silverlight trả lại (thậm chí nếu có lỗi trong phân tích cú pháp tập tin), nó được sử dụng như là chính sách cho các tập tin mà request cross-domain và tất cả các request tới máy chủ cho những phiên làm việc của các ứng dụng Silverlight. Nếu tập tin chính sách Silverlight không được tìm thấy, Silverlight 2 sau đó sẽ cố gắng tải về một tập tin chính sách Flash có tên "crossdomain.xml" ở gốc của site mục tiêu được yêu cầu bằng cách sử dụng giao thức HTTP. Các chính sách tập tin Flash phải cho phép kết nối đến tất cả các tên miền cho nó sẽ được sử dụng bởi các Silverlight WebClient và các lớp HTTP.

Nếu được yêu cầu kết nối từ các sockets cho trang web (cross-domain site hoặc site gốc), Silverlight 2 cố gắng để mở một kết nối TCP vào cổng 943 trên trang web mục tiêu. Nếu kết nối TCP có thể được thiết lập, Silverlight 2 gửi chuỗi đặc biệt <policy-file-request/> đến server để yêu cầu tập tin chính sách Silverlight. Silverlight 2 đợi để sau đó nhận được trả lời từ web mục tiêu có chứa tập tin chính sách Silverlight. Nếu tập tin chính sách Silverlight này được trả lại (thậm chí nếu có một lỗi trong phân tích cú pháp tập tin), nó được sử dụng như là tập tin chính sách cho các request socket và tất cả các request cho trang web mục tiêu tất cả các session của ứng dụng Silverlight.

Nếu tập tin chính sách tải về là thành công và phân tích cú pháp cho phép, cuối cùng là mở một kết nối đến các host mục tiêu. Nếu tập tin chính sách tải về là không hợp lệ và không thể phân tích cú pháp một cách chính xác, sau đó kết nối tới tài nguyên mạng của Silverlight 2 sẽ bị từ chối và các yêu cầu kết nối sẽ không thành công. Nếu không có chính sách tập tin được tìm thấy, sau đó kết nối vào mạng nguyên là bị từ chối của Silverlight 2 và các yêu cầu kết nối sẽ không thành công.

Bổ sung một trong những hạn chế bằng cách sử dụng các lớp sockets là ứng dụng mạng cho phép kết nối chỉ với những cổng trong phạm vi 4502-4534. Ứng dụng Silverlight chỉ được phép kết nối tới những cổng như vậy khi sử dụng socket. Nếu các cổng mà bạn truy cập không phải trong phạm vi cổng này, các cố gắng kết nối sẽ thất bại.

Để triển khai tập tin chính sách bảo mật sử dụng các kết nối từ WebClient và các lớp HTTP, quản trị viên hệ thống cần phải cấu hình Web service cho mỗi địa chỉ IP mà là cung cấp cho các tập tin định nghĩa chính sách và tạo những tập tin chính sách bảo mật của Flash và Silverlight để có thể tìm thấy qua giao thức HTTP.

Để triển khai tập tin chính sách bảo mật trên máy cho socket, quản trị viên hệ thống cần phải cấu hình một dịch vụ xác thực riêng biệt trên cổng 943 cho mỗi địa chỉ IP đó là cung cấp định nghĩa các tập tin chính sách.

3 Truy cập web service trong silverlight

Ứng dụng máy khách Silverlight 2 chạy trên trình duyệt và thường cần phải kết nối tới nhiều nguồn cơ sở dữ liệu ngoài khác nhau. Một ví dụ điển hình là việc truy cập dữ liệu vào cơ sở dữ liệu từ một máy chủ và hiển thị nó lên giao diện người dùng Silverlight. Kịch bản phổ biến khác là cập nhật dữ liệu trên service thông qua ứng dụng Silverlight để truyền tải thông tin tới service đó. Những nguồn dữ liệu ngoài thường được lấy về từ Web service. Những service này có thể là SOAP service được tạo sử dụng Windows

Communication Foundation(WCF) hoặc một vài nền tảng SOAP khác hoặc chỉ đơn giản là HTTP hoặc Service tĩnh. Ứng dụng Silverlight máy khách có thể truy cập những Web service trực tiếp hoặc sử dụng proxy được sinh ra từ metadata được published bởi service.Silverlight cung cấp các tính năng cần thiết để làm việc với một loạt các định dạng dữ liệu được sử dụng bởi các Service. Bao gồm các định dạng XML, JSON, RSS, và Atom. Những định dạng dữ liệu đó được truy cập sử dụng Serialization components, Linq to XML, Linq to JSON, và Syndication components.

3.1 Bảo mật cho truy cập Service

- Bảo vệ ứng dụng máy khách Silverlight và dữ liệu người dùng

Để bảo vệ ứng dụng silverlight khỏi những service độc hại, bạn phải biết về một số vấn đề bảo mật dưới đây:

+ Service có thể gửi về những thông tin mà làm cho ứng dụng máy khách Silverlight và có thể là trình duyệt của người dụng bị treo hoặc lỗi. Điều này là có thể với bất kỳ dữ liệu nào có định dạng XML, JSON, RSS, Atom và SOAP. Để tránh việc đó cần thực hiện như sau:

+ Tránh khởi tạo giao tiếp tới service không tin tưởng khi mà bị lỗi hoặc treo sẽ gây ra mất dữ liệu của ứng dụng của người dùng. Ví dụ, lưu lại công việc của người dùng xuống máy trước khi khởi tạo giao tiếp.

+ Tránh tự động khởi tạo tới service không tin tưởng, mà không cho phép người dùng lựa chọn. Điều này có thể dẫn tới ứng dụng của bạn không thể dùng được. Hãy ghi nhớ rằng một dịch vụ độc hại có thể không luôn trả về dữ liệu độc hại.

- Bảo vệ Service

Hãy ghi nhớ rằng service mà bạn publish cho các ứng dụng máy khách Silverlight có thể được truy cập của bất kỳ người nào và được sử dụng trong với những cách mà bạn có thể không mong đợi. Xem xét việc phòng ngừa sau đây:

+ Không nên đưa ra dữ liệu thừa với mục đích sử dụng.

+ Chắc chắn sử dụng xác thực người dùng thích hợp về mặt công nghệ

3.2 Tạo một Service vượt qua phạm vi domain

Để cho phép Silverlight quản lý quyền truy cập vào service của domain khác, thì service phải tuỳ chỉnh và chỉ ra rõ ràng cross-domain nào được phép truy cập. Bằng việc tuỳ chỉnh, Silverlight control có thể truy cập tới service một cách thật an toàn, mà không gây ảnh hưởng xấu đến dữ liệu mà lưu trữ bởi service.

Silverlight 2 hỗ trợ hai cơ chế khác nhau cho service để tuỳ chỉnh quyền truy cập cross-domain:

- Đặt tập tin clientaccesspolicy.xml ở gốc của domain nơi mà service được lưu trữ để thiết lập quyền truy cập của cross-domain tới service.
- Đặt tập tin crossdomain.xml hợp lệ ở gốc của domain nơi service được lưu trữ. Tập tin phải được đánh dấu là public cho toàn domain

- Sử dụng tập tin clientaccesspolicy.xml để cho phép truy cập cross-domain

1. Tạo service cho phép máy khách Silverlight truy cập
2. Tạo tập tin clientaccesspolicy.xml, cho phép truy cập vào các service. Thiết lập dưới đây cho phép truy cập từ bất kỳ domain nào tới nguồn tài nguyên của domain.

```
<?xml version="1.0" encoding="utf-8"?>
<access-policy>
 <cross-domain-access>
  <policy>
   <allow-from http-request-headers="*">
    <domain uri="*"/>
   </allow-from>
   <grant-to>
    <resource path="/" include-subpaths="true"/>
   </grant-to>
  </policy>
 </cross-domain-access>
</access-policy>
```

3. Lưu tập tin clientaccesspolicy.xml tới gốc của domain nơi service được lưu trữ. Cho ví dụ này, nếu service được lưu trữ tại http://fabrikam.com thì tập tin phải đặt tại http://fabrikam.com/clientaccesspolicy.xml.

- Sử dụng tập tin crossdomain.xml để cho phép truy cập cross-domain

1. Tạo service cho phép Silverlight client truy cập
2. Tạo tập tin crossdomain.xml có những thiết lập dưới đây. Tập tin phải được cấu hình để cho phép truy cập vào service từ bất kỳ tên miền khác, hoặc nó không được nhận ra bởi Silverlight 2

```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM
"http://www.macromedia.com/xml/dtds/cross-domain-policy.dtd">
<cross-domain-policy>
 <allow-http-request-headers-from domain="*" headers="*"/>
```

```
</cross-domain-policy>
```

3. Lưu tập tin crossdomain.xml vào gốc của domain là nơi mà các service được lưu trữ trên máy chủ. Ví dụ, nếu service được lưu trữ trên máy chủ http://fabrikam.com thì tập tin phải được đặt tại http://fabrikam.com/crossdomain.xml

4 Làm việc với socket

Namespace System.Net.Sockets được thêm vào trong phiên bản Silverlight 2 cung cấp quản lý việc triển khai thực hiện của socket giao tiếp mạng cho những người lập trình viên mà cần phải kiểm soát chặt chẽ quyền truy cập vào mạng. Trên Windows, System.Net.Sockets cung cấp một tên quản lý việc triển khai thực hiện của Windows sockets (Winsock). Với Apple trên hệ điều hành Mac OS X, các namespace System.Net.Sockets cung cấp, quản lý việc triển khai thực hiện của các sockets dựa trên Berkeley Software Distribution (BSD) UNIX.

Namespace System.Net.Sockets cung cấp một cơ chế cho thời gian thực, hai giao tiếp với mạng lưới các tài nguyên từ xa và cho phép mức độ cao hơn các API để giao tiếp qua một bi-directional. Điều này cũng cho phép một ứng dụng để nội tác như là một client hiện tại với các service TCP.

4.1 Hỗ trợ giao thức mạng

Namespace System.Net.Sockets hỗ trợ việc sử dụng các giao thức IPv4 hay IPv6 chỉ cần mạng trên máy tính đã kích hoạt hỗ trợ cho IPv4 và IPv6. Các lớp được thêm vào trong namespace System.Net ở Silverlight 2 để làm việc với System.Net.Sockets. Những lớp mới bao gồm sau đây:

+ EndPoint - xác định một địa chỉ mạng. Đây là một lớp trừu tượng.

+ DnsEndPoint - đại diện cho một mạng lưới endpoint như là một tên máy chủ hoặc một chuỗi ký tự đại diện của một địa chỉ IP và một số cổng.

+ Ipaddress - Cung cấp địa chỉ giao thức mạng (IP).

+ IPEndPoint - đại diện cho một mạng lưới endpoint như là một địa chỉ IP và một số cổng.

+ SocketAddress – Lưu trữ thông tin được sắp xếp và kế thừa từ lớp EndPoint.

Lớp Socket cung cấp một tập hợp các phương thức cho giao tiếp mạng. Lớp Socket cho phép bạn thực thi bất đồng bộ việc truyền tải dữ liệu sử dụng bất cứ giao thức giao

tiếp nào được liệt kê ở kiểu ProtocolType. Hiện tại, trong Silverlight 2 chỉ hỗ trợ loại giao thức TCP.

Một trong những hạn chế về việc sử dụng các sockets trong Silverlight 2 là các cổng phải trong phạm vi 4502-4534. Chỉ những cổng này cho phép kết nối bằng cách sử dụng sockets trong ứng dụng Silverlight 2. Nếu một kết nối đến một cổng không phải là trong phạm vi cổng này, việc kết nối sẽ không thành công

4.2 Lập trình mạng cơ bản với Socket

Lớp Socket cho phép bạn thực thi bất đồng bộ truyền tải dữ liệu sử dụng những phương thức sau đây:

+ ConnectAsync - Bắt đầu Asynchronous request tới máy chủ.

+ SendAsync - Ghi dữ liệu từ một hay nhiều buffers tới Socket đã kết nối .

+ ReceiveAsync - Đọc dữ liệu vào một hoặc nhiều buffers từ Socket đã kết nối.

+ Shutdown - Kết thúc mọi thao tác gửi đang chờ, và các tín hiệu ở endpoint rằng các kết nối có thể đã bị đóng lại. Nếu việc gửi được xác định, dữ liệu có thể vẫn đang nhận được cho đến khi máy chủ kết thúc kết nối của nó (chỉ nhận được 0 byte).

+ Close - Là đóng kết nối máy chủ và lưu trữ tất cả các tài nguyên quản lý và không quản lý có liên quan tới Socket.

Trong lớp Socket, hoạt động asynchronous của Socket được mô tả bởi khả năng dùng lại System.Net.Sockets.SocketAsyncEventArgs phân bổ và duy trì các đối tượng của ứng dụng. Các ứng dụng có thể tạo bao nhiêu SocketAsyncEventArgs của các đối tượng mà nó cần. Ví dụ, nếu một ứng dụng Silverlight cần phải có 10 Socket cho các thao tác cùng một lúc, nó có thể phân bổ 10 SocketAsyncEventArgs cho các đối tượng trong trường hợp này.

Vòng đời của SocketAsyncEventArgs được sử dụng trong asynchronous Socket động được xác định bởi sự tham chiếu tới mã của ứng dụng và tham chiếu tới asynchronous I / O . Nó là không cần thiết cho ứng dụng để duy trì một tham chiếu cho các đối tượng SocketAsyncEventArgs sau khi nó được gửi đi như là một tham số vào một trong những phương pháp asynchronous Socket. Nó sẽ vẫn được tham chiếu cho đến khi được trả về hoàn tất. Tuy nhiên, đó là lợi thế cho ứng dụng để giữ lại tham chiếu đến các đối tượng SocketAsyncEventArgs để nó có thể được tái sử dụng cho asynchronous socket tiếp theo hoạt động.

5 Mã hóa dữ liệu của service

Namespace System.Security.Cryptography chứa các lớp mà cho phép bạn thực hiện cả hai loại mã hoá đối xứng và mã hoá bất đối xứng, tạo ra hashes và chữ ký số, và số ngẫu nhiên. Phần này mô tả cách thức để tạo ra mã hash và chữ ký số.

Các bạn có thể dọc thêm tại .NET Framework, Cryptographic Services