

Flex secure-amf 채널 사용하기

최근 갑자기 보안에 관심이 생겨서, 여러 가지 문제점들을 해결해보고자 이 글을 적습니다.

LCDS(LCDS 2.6 기준)에서는 몇 개의 보안 채널을 제공합니다.

그 내용은 LCDS 를 배포 후 WEB-INF\flexWservices-config.xml 에서 확인할 수 있는데요,

```
<?xml version="1.0" encoding="UTF-8"?>
<services-config>

  <channels>
    <!-- Secure Servlet-based endpoints -->
    <channel-definition id="my-secure-amf" class="mx.messaging.channels.SecureAMFChannel">
      <endpoint url="https://{server.name}:{server.port}/{context.root}/messagebroker/amfsecure"
class="flex.messaging.endpoints.SecureAMFEndpoint"/>
      <properties>
        <!--HTTPS requests on some browsers do not work when pragma "no-cache" are set-->
        <add-no-cache-headers>false</add-no-cache-headers>
      </properties>
    </channel-definition>

    <channel-definition id="my-secure-http" class="mx.messaging.channels.SecureHTTPChannel">
      <endpoint url="https://{server.name}:{server.port}/{context.root}/messagebroker/httpsecure"
class="flex.messaging.endpoints.SecureHTTPEndpoint"/>
      <properties>
        <!--HTTPS requests on some browsers do not work when pragma "no-cache" are set-->
        <add-no-cache-headers>false</add-no-cache-headers>
      </properties>
    </channel-definition>

    <!-- Secure NIO based endpoints -->
    <!--
    <channel-definition id="secure-nio-amf" class="mx.messaging.channels.SecureAMFChannel">
      <endpoint url="https://{server.name}:2443/securenioamf" class="flex.messaging.endpoints.SecureNIOAMFEndpoint"/>
      <server ref="secure-nio-server"/>
      <properties>
        <polling-enabled>false</polling-enabled>
      </properties>
    </channel-definition>
  </channels>
</services-config>
```

```

    </properties>
</channel-definition>

<channel-definition id="secure-nio-http" class="mx.messaging.channels.SecureHTTPChannel">
  <endpoint url="https://{server.name}:2443/secureniohttp" class="flex.messaging.endpoints.SecureNIOHTTPEndpoint"/>
  <server ref="secure-nio-server"/>
  <properties>
    <polling-enabled>false</polling-enabled>
  </properties>
</channel-definition>
-->
</channels>
</services-config>

```

위와 같이 *SecureAMFChannel*, *SecureHTTPChannel* 채널을 이용한 *my-secure-amf*, *my-secure-http*, *secure-nio-amf*, *secure-nio-http* 가 있습니다.

다른 건 같으리라 생각되어, 가장 많이 사용하는 RemoteObject 에서 사용하는 amf 의 보안채널인 *secure-amf* 채널을 사용해보겠습니다.

일단 HTTPS/SSL 을 사용하기 위해서는 필요한 것이 몇 가지가 있는데,.

JSSE (Java Secure Socket Extension) 을 사용해 디지털 인증서를 만들어야 합니다.
 (<http://docs.sun.com/app/docs/doc/820-4605/ablqz?!=ko&a=view>)

내용이 옵션도 많고 복잡합니다 ㅎㅎ

인증서를 만들고 그다음으로는 WAS 에서 설정을 해야합니다.

Tomcat 6.0 을 기준으로 설명하겠습니다.

우선 다음 링크를 참고하면,
<http://www.mbaworld.com/docs/ssl-howto.html>에 SSL Configuration HOW-TO라는 제목으로 SSL을 설정하는 방법이 나와있습니다.

Quick Start 에 간단히 인증서를 만드는 방법이 나와있습니다.

- Create a certificate keystore by executing the following command:

Windows:

```
%JAVA_HOME%\bin\keytool -genkey -alias tomcat -keyalg RSA
```

Unix:

```
$JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA
```

and specify a password value of "changeit".

- Uncomment the "SSL HTTP/1.1 Connector" entry in `$CATALINA_BASE/conf/server.xml` and tweak as necessary.

이제 만들어 볼까요?

다음과 같이 keytool 을 이용해 키와 인증서를 생성합니다.

```
C:\WDocuments and Settings\dannys>keytool -genkey -keyalg RSA -alias tobit -keystore .keystore
```

keystore 암호를 입력하십시오:

이름과 성을 입력하십시오.

```
[Unknown]: localhost
```

조직 단위 이름을 입력하십시오.

```
[Unknown]: Solution Dept
```

조직 이름을 입력하십시오.

```
[Unknown]: WeMB
```

구/군/시 이름을 입력하십시오?

```
[Unknown]: GaSan
```

시/도 이름을 입력하십시오.

```
[Unknown]: Seoul
```

이 조직의 두 자리 국가 코드를 입력하십시오.

```
[Unknown]: KR
```

CN=localhost, OU=Solution Dept, O=WeMB, L=GaSan, ST=Seoul, C=KR 이(가) 맞습니까?

```
[아니오]: y
```

<tobit>에 대한 키 암호를 입력하십시오.

(keystore 암호와 같은 경우 Enter 를 누르십시오):

주의!> 모두 영문으로 넣어야합니다.

이렇게 하면 인증서가 생성됩니다.

그 다음 인증서를 정보인증기관에서 신청을 받아야 한다.
매년 약 20~30 만원 정도의 금액이 사용된다.

그래서, 테스트를 위해 14 일간 사용이 가능한 Trial 인증서를 받아서 하겠습니다.

(http://www.crosscert.com/service_global/issuance/Main.jsp?_action=SHOW&_param=GLOBAL_PRODUCTS_TRIAL_INTRO_PAGE)

다음 페이지에 접속해,

keytool 로 생성한 인증서의 내용으로 넣고, 인증서 정보 등을 넣으면 된다. 이것 역시 모두 영문으로 해야 합니다.

그러면 잠시 후 다음과 같은 등록 메일이 옵니다.

YOUR ORDER NUMBER: 373912354534

Dear VeriSign Customer:

Congratulations! Your Trial SSL Certificate, issued to:

CN: LOCALHOST
O: WEMB
OU: SOLUTION DEPT

can be installed by following the instructions below.

1. Before using your Trial SSL Certificate, install the Test CA Root in each browser you plan to use as part of your test of SSL. To download the Test CA Root, go to:

<http://www.verisign.com/server/trial/faq/index.html>

and follow the instructions there.

2. Install the Secure Site Trial Intermediate CA on each Web server you are testing with. To download the the Trial intermediate CA.

go to: <http://www.verisign.com/support/verisign-intermediate-ca/trial-secure-server-intermediate/index.html>

Note: Customers using IIS 5.0 or 6.0 servers can skip this step

3. To install your Trial SSL Certificate, go to:

<http://www.verisign.com/support/install/index.html#trial>

and follow the instructions there.

4. After testing your Trial SSL Certificate, you'll need to purchase a full-service Secure Site SSL Certificate, available as part of VeriSign's trust solutions:

- Secure Site Pro Certificates that enable 128-bit SSL encryption -- the world's strongest -- with all Microsoft and Netscape browsers.

- Secure Site Certificates that, like your Trial SSL Certificate, enable industry-standard 40-bit SSL when communicating with export-version Netscape and Microsoft Internet Explorer browsers, and 128-bit SSL encryption when communicating with domestic-version browsers.

- VeriSign trust solutions also include additional services, such as up to \$250,000 of NetSure protection, the widely recognized VeriSign Secured Seal to post on your site as a symbol of trust, Keynote performance monitoring services, and more.

- Learn more about all of VeriSign's trust solutions at: <http://www.verisign.com/products/site>.

If you have any questions about installing or using your Trial SSL Certificate, please contact us at 1-650-426-3400.

Thank you for your interest in VeriSign products!

VeriSign Customer Support Department
Hours of Operation: 5AM-6PM Pacific Time, Monday-Friday
E-mail: support@verisign.com
Web: <http://www.verisign.com>
Phone: 1-877-GET-VRSN 1-877-438-8776 or 1-650-426-3400
Fax: 1-650-961-8870

-----BEGIN CERTIFICATE-----

MIIFBTCCA+2gAwIBAgIQDWB6/7He+cwj8NhHEU0x4TANBgkqhkiG9w0BAQUFADCB
yzELMAkGA1UEBhMCVVMxZmFzAVBgNVBAoTDDZlcmIITaWduLCBjbmuMTAwLgYDVQQL

...

kpk+L87mADNxjpe50vPP9mZ76UzluQYn0ByL7unRcPRW5eLxPnkO5rHLBxY04sVe
OTu5vF4ZZIXA6Mo6a6PNtQXWFInOAIaOphKqO8RqpxY7LtbAjfYmgWfHiFqYhN8z

JjzoM4VPZOII12S2jBvi9Q925YuawvFPr2E/QcYImLzzE/RPtP8Agj4=
-----END CERTIFICATE-----

메일의 내용대로 진행하시면 됩니다.

그러면 인증서 내용을 새 텍스트에 붙여넣고 저장합니다.

3 개의 인증서 파일로 저장합니다. TrialRootCA.cer / IntermediateCA.cer / id.cer

이메일 하단이 id.cer / 1 번 항목이 TrialRootCA.cer / 2 번이 IntermediateCA.cer 입니다.

그러면 이제 인증서를 서버에 import 해야 합니다.

>TrialRootCA.cer Import

```
C:\WDocuments and Settings\Wdanny>keytool -import -alias root -keystore .keystore -trustcacerts -file TrialRoot  
CA.cer
```

keystore 암호를 입력하십시오:

소유자: CN=VeriSign Trial Secure Server Test Root CA, OU="For Test Purposes Only. No assurances.", O="VeriSign, Inc.", C=US

발급자: CN=VeriSign Trial Secure Server Test Root CA, OU="For Test Purposes Only. No assurances.", O="VeriSign, Inc.", C=US

일련 번호: 20a897aedb8202dec136a04e26bd8773

유효 기간 시작: Wed Feb 09 09:00:00 KST 2005 끝: Sun Feb 09 08:59:59 KST 2025

인증 지문:

MD5: B6:9D:A4:40:52:02:50:0D:D5:9C:E1:B8:4B:66:C4:AC

SHA1: 81:A7:B1:CA:51:66:D1:2D:CB:32:CA:00:21:C3:9E:49:54:73:56:65

서명 알고리즘 이름: MD2withRSA

버전: 1

이 인증서를 신뢰하십니까? [아니오]: y

인증이 keystore 에 추가되었습니다.

> IntermediateCA.cer Import

```
C:\WDocuments and Settings\Wdannys>keytool -import -alias ca -keystore .keystore -trustcacerts -file IntermediateCA.cer
keystore 암호를 입력하십시오:
인증이 keystore 에 추가되었습니다.
```

>Id.cer Import

```
C:\WDocuments and Settings\Wdannys>keytool -import -alias tobit -keystore .keystore -trustcacerts -file id.cer
keystore 암호를 입력하십시오:
인증서 회신이 keystore 에 설치되었습니다.
```

그 다음 WAS 에 TOMCAT 에 셋팅을 합니다.

%TOMCAT_HOME%\conf\server.xml 중간에 보면, 주석으로 되어있습니다.

```
<!-- Define a SSL HTTP/1.1 Connector on port 8443
      This connector uses the JSSE configuration, when using APR, the
      connector should be using the OpenSSL style configuration
      described in the APR documentation -->

    <Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
               maxThreads="150" scheme="https" secure="true"
               clientAuth="false" sslProtocol="TLS"
               keystoreFile="${user.home}/.keystore" keystorePass="password"
               />
```

다음과 같이 넣어줍니다.

Flex RemoteObject 를 사용하기 위해 WEB-INF\flex\Wremoting-config.xml 에 secure-amf 채널을 설정해줍니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<service id="remoting-service"
  class="flex.messaging.services.RemotingService">

  <adapters>
    <adapter-definition id="java-object" class="flex.messaging.services.remoting.adapters.JavaAdapter" default="true"/>
  </adapters>

  <default-channels>
    <channel ref="my-secure-amf"/>
  </default-channels>

  <destination id="secureService">
    <properties>
      <source>secure.SecureService</source>
    </properties>
  </destination>

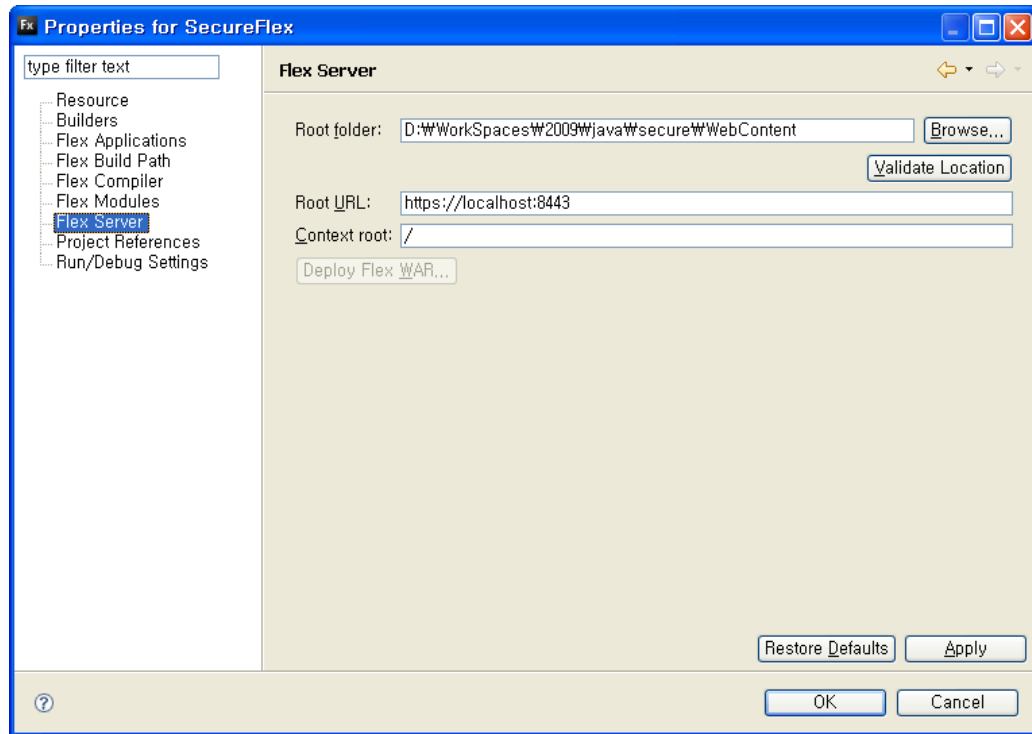
</service>
```

이제 SSL 을 사용하기 위한 서버 설정은 모두 끝납니다.

그러면 Flex 에서의 사용법을 알아보겠습니다.

LCDS 를 사용하는 Flex Project 를 생성합니다.

생성할 때, 다음과 같이



Root URL 을 https 로 주는 방법과, 일반적인 http 로 주는 방법이 있습니다.

전자의 방법으로 하겠습니다.

Application 파일에 다음과 같이 내용을 넣고, 실행하면 됩니다.

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
  layout="vertical"
  creationComplete="initApp()">

  <mx:Script>
    <![CDATA[
      import mx.messaging.Channel;
      import mx.messaging.ChannelSet;
      import mx.messaging.channels.SecureAMFChannel;
      import mx.rpc.events.FaultEvent;
      import mx.rpc.events.ResultEvent;

      private function initApp():void
      {
        var cs:ChannelSet = new ChannelSet();
        var customChannel:Channel
          = new SecureAMFChannel("my-secure-amf", "https://localhost:8443/messagebroker/amfsecure");

        cs.addChannel(customChannel);

        remoteObject.channelSet = cs;
      }

      private function resultHandler(event:ResultEvent):void
      {
        ta.text = event.message.toString();
      }

      private function faultHandler(event:FaultEvent):void
      {
        ta.text = event.message.toString();
      }
    ]]>
  </mx:Script>

  <mx:RemoteObject id="remoteObject"
    destination="secureService"
    result="resultHandler(event)"
    fault="faultHandler(event)">
  </mx:RemoteObject>

  <mx:Button label="Send" click="remoteObject.secureTest('aaa')"/>
</mx:Application>
```

```
<mx:TextArea id="ta" width="100%" height="100%"/>
```

```
</mx:Application>
```

혹, "crossdomain.xml 파일을 가져올 수 없습니다." 라고 에러가 날 경우, 브라우저에서 crossdomain.xml 을 열어보시기 바랍니다.
https 로 요청시 거부되어서 그럴수도 있습니다.

이렇게 하면 테스트가 끝납니다. ㅎㅎ

하기 전에는 꽤 복잡했는데, 하고 나니 별거 아닙니다. ㅎㅎ