Wowza Media Server® 2

Wowza Media Server 2
Media Security

Wowza Media Server 2: Media Security

Third-Party Information

This document contains links to third-party websites that are not under the control of Wowza Media Systems, Inc. ("Wowza") and Wowza is not responsible for the content on any linked site. If you access a third-party website mentioned in this document, then you do so at your own risk. Wowza provides these links only as a convenience, and the inclusion of any link does not imply that Wowza endorses or accepts any responsibility for the content on third-party sites.

Trademarks

Wowza, Wowza Media Systems, Wowza Media Server and related logos are trademarks of Wowza Media Systems, Inc., and may be registered in the United States or in other jurisdictions including internationally.

Adobe and Flash are registered trademarks of Adobe Systems Incorporated, and may be registered in the United States or in other jurisdictions including internationally.

Silverlight is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries.

QuickTime, iPhone and iPod are either registered trademarks or trademarks of Apple, Inc. in the United States and/or other countries.

Other product names, logos, designs, titles, words, or phrases mentioned may be trademarks, service marks or trade names of other entities and may be registered in certain jurisdictions including internationally.

Third Party Copyright Notices

Log4j and Mina: Copyright © 2006 The Apache Software Foundation

Java ID3 Tag Library and JLayer 1.0 (classic): Copyright © 1991, 1999 Free Software Foundation, Inc.

Java Service Wrapper: Copyright © 1999, 2006 Tanuki Software, Inc.

Bouncy Castle Crypto API: Copyright © 2000 – 2008, The Legion Of The Bouncy Castle

Table of Contents

Supported Wowza Media Server Versions

This document describes how to use a security package that contains features that work with Wowza Media Server 2 Preview 7 to Wowza Media Server 3.1.2. The security features that are described in this document do not work with later versions of Wowza Media Server.

Introduction

This document covers the available options and procedures for securing Wowza Media Server and the media you plan to stream through Wowza Media Server. There are several Wowza Media Server features that are needed to properly secure your content. Some of these are SecureToken, RTMP authentication, RTSP authentication, StreamNameAlias and secure streaming (RTMPE, RTMPTE and RTMPS).

The methods required to secure Wowza Media Server differ based on the method used to stream media. For example to protected video on demand streaming it is best to use SecureToken along with RTMPE while protecting live streaming may require a combination of SecureToken, RTMP authentication, RTSP authentication, StreamNameAlias and RTMPE.

The SecureToken, RTMP authentication, and "ModuleRequireSecureConnection" (a module that requires secure streaming over RTMPE, RTMPTE or RTMPS) security features are included in this package. RTSP authentication is built into Wowza Media Server. The StreamNameAlias AddOn package is a separate package that can downloaded from the following article:

http://www.wowza.com/forums/content.php?116

These security features can be used in tandem and are frequently installed and configured together. The Example Configurations section of this document shows some useful examples.

Overview

Below is a description of each of the security features and packages covered in this document.

RTMPE, RTMPTE and RTMPS

RTMPE, RTMPTE and RTMPS are three methods for encrypting the data sent between the Flash player and the Wowza Media server. RTMPE is an encrypted version of RTMP. RTMPTE is an encrypted version of RTMPT (which is the HTTP tunneling version of RTMP). RTMPS

uses SSL to protected RTMPT streaming. There is no special configuration needed to do RTMPE and RTMPTE streaming. You simply just specify rtmpe:// or rtmpte:// as the protocol portion of the server URL when connecting to Wowza Media Server from the Flash player. RTMPS does require that you designate and configure an SSL enabled HostPort in [install-dir]/conf/VHost.xml for secure streaming. This process is covered in the Wowza Media Server User's Guide.

This security package includes the module "ModuleRequireSecureConnection" which will require one of these protocols is used to connect to Wowza Media Server. There is also an option in the modules "ModuleSecureToken" and "ModuleRTMPAuthentication" to turn on this same requirement.

SecureToken

SecureToken is a challenge and response based security system that when used in conjunction with RTMPE/RTMPTE provides a high level of content protection against spoofing threats like those posed by the "Replay Media Catcher" and "Grab Pro from Orbit Downloader". Each connection is protected by a random single use key and a password (shared secret). The basic security methodology is described below.

The way SecureToken works is that upon client connection the provided custom module generates a unique key for the pending connection. The generated key is encrypted using a shared secret and is returned as part of the NetConnection.onStatus info object. The Flash client movie then decrypts the unique key using the same shared secret and sends the result back to the custom module. The server then compares this key to the originally generated key. If the values do not match then the connection is aborted.

SecureToken security is included in the following modules: "ModuleSecureToken", and "ModuleRTMPAuthenticate".

RTMP Authentication

RTMP authentication provides a means for username and password protecting the RTMP publishing interface into Wowza Media Server. Authentication is configured on a per-application basis by adding and configuring the "ModuleRTMPAuthenticate" in Application.xml. Usernames and passwords are stored in the "[install-dir]/conf/publish.password" file.

RTMP authentication is included in the "ModuleRTMPAuthenticate" module.

RTSP Authentication

RTSP authentication provides a means for username and password protecting the RTSP/RTP publishing and playback interface into Wowza Media Server. Authentication is configured on a per-application basis using the RTP/Authentication/[PublishMethod and PlayMethod] properties in Application.xml. Usernames and passwords are stored in the "[install-dir]/conf/publish.password" file. This feature is covered in more detail in the Wowza Media Server 2 User's Guide.

RTSP authentication is built into Wowza Media Server.

StreamNameAlias

StreamNameAlias is system for creating simple aliases that can expand into more complicated stream names. This package can also be used to limit the stream names that can be accepted by Wowza Media Server. This is the feature of this package that we will be leveraging to help protect live re-streaming. The complete details of how this system works is covered in detail in the StreamNameAlias documentation that accompanies the package.

The StreamNameAlias package can be downloaded from here:

http://www.wowza.com/forums/content.php?116

Additional Security Methods

Below are a few additional security methods that are not covered in detail in this document. They are mainly used to protect the publishing of live streaming when using an encoder that does not support RTMP authentication:

Secure URL Params

http://www.wowzamedia.com/forums/content.php?233

onConnect Authenticate

http://www.wowzamedia.com/forums/content.php?160

onConnect Authenticate2

http://www.wowzamedia.com/forums/content.php?234

Installation

To install "ModuleSecureToken", "ModuleRTMPAuthentication" and "ModuleRequireSecureConnection", copy the files "lib/wms-plugin-security.jar" and "lib/wms-plugin-security-encryption.jar" from this package to the "[install-dir]/lib" folder of Wowza Media Server.

Server-side AddOn Modules

ModuleRequireSecureConnection

The "ModuleRequireSecureConnection" module requires that RTMPE, RTMPTE or RTMPS is used to connect to a given application in Wowza Media Server. You configure this module in

"Application.xml" by adding a <Module> entry for it as the last entry in the <Modules> list. Below are the steps to configure an application named "securestreaming" that uses this module:

1. Create folder **[install-dir]/applications/securestreaming**.

2. Create folder **[install-dir]/conf/securestreaming** and copy **[install-dir]/conf/Application.xml** into this new folder.

3. Edit the newly copied **Application.xml** file and add the following **<Module>** definition as the last entry in the **<Modules>** list:

```
<Module>
    <Name>ModuleRequireSecureConnection</Name>
    <Description>ModuleRequireSecureConnection</Description>
    <Class>com.wowza.wms.plugin.security.ModuleRequireSecureConnection</Class>
</Module>
```

To test, open the SimpleVideoStreaming player from the Wowza examples, then connect to the application and attempt to play a video using the RTMP protocol, then the RTMPE protocol. RTMP should not work because it is not a secure protocol so Wowza will reject the connection. RTMPE and RTMPTE should work. And RTMPS will work if you have a certificate setup.

ModuleRTMPAuthentication

The "ModuleRTMPAuthentication" module enables username and password RTMP authentication for encoders such as Flash Media Live Encoder (consult the EULA for streaming restrictions), Telestream Wirecast, Orban Opticodec and On2 Flix Live. Only connections that are properly authenticated with a valid username and password will be allowed to publish live content to application configured to use this module. This module can also be configured to protect playback using SecureToken. Below are the steps to configure an application named "securestreaming" that uses this module and protects publishing use RTMP authentication and playback using SecureToken:

1. Create folder **[install-dir]/applications/securestreaming**.

2. Create folder **[install-dir]/conf/securestreaming** and copy **[install-dir]/conf/Application.xml** into this new folder.

3. Edit the newly copied **Application.xml** file, change the **Streams/StreamType** to **live**, verify the **RTP/Authentication/PublishMethod** is set to **digest** and add the following **<Module>** definitions as the last entries in the **<Modules>** list:

```
<Module>
    <Name>ModuleRTMPAuthenticate</Name>
    <Description>ModuleRTMPAuthenticate</Description>
    <Class>com.wowza.wms.plugin.security.ModuleRTMPAuthenticate</Class>
</Module>
```

4. To turn on SecureToken security, add the following property to the application level **<Properties>** container at the bottom of the **Application.xml** file (be sure to get the correct **<Properties>** container – there are several in the file). If you omit this property then SecureToken will not be used to protect playback:

```
<Property>
  <Name>secureTokenSharedSecret</Name>
  <Value>#ed%h0#w@1</Value>
</Property>
```

5. To require a secure connection be used for playback of content, add the following property to the application level **<Properties>** container at the bottom of the **Application.xml** file (be sure to get the correct **<Properties>** container – there are several in the file):

```
<Property>
  <Name>requireSecureConnection</Name>
  <Value>true</Value>
  <Type>Boolean</Type>
</Property>
```

6. Edit **[install-dir]/conf/publish.password** and a username and password line for each user to which you want to grant publishing access. For example to add the user "myuser" with the password "mypassword", edit the contents of the publish.password file as follows:

```
# Publish password file (format [username][space][password])
#username password
myuser mypassword
```

ModuleSecureToken

The "ModuleSecureToken" module adds SecureToken protection to publishing and playback of content through Wowza Media Server. Below are the steps to configure an application named "securestreaming" that uses this module:

1. Create folder **[install-dir]/applications/securestreaming**.

2. Create folder **[install-dir]/conf/securestreaming** and copy **[install-dir]/conf/Application.xml** into this new folder.

3. Edit the newly copied **Application.xml** file and add the following **<Module>** definition as the last entry in the **<Modules>** list:

```
<Module>
    <Name>ModuleSecureToken</Name>
    <Description>ModuleSecureToken</Description>
    <Class>com.wowza.wms.plugin.security.ModuleSecureToken</Class>
</Module>
```

4. Add the following property to the application level **<Properties>** container at the bottom of the **Application.xml** file (be sure to get the correct **<Properties>** container – there are several in the file):

```
<Property>
  <Name>secureTokenSharedSecret</Name>
  <Value>#ed%h0#w@1</Value>
</Property>
```

5. To require a secure connection be used for playback of content, add the following property to the application level **<Properties>** container at the bottom of the

Application.xml file (be sure to get the correct **<Properties>** container – there are several in the file):

```
<Property>
  <Name>requireSecureConnection</Name>
  <Value>true</Value>
  <Type>Boolean</Type>
</Property>
```

Additional Security Modules

Below are a few additional security modules that are not covered in detail in this document. They are mainly used to protect the publishing of live streaming when using an encoder that does not support RTMP authentication:

ModuleSecureURLParams

http://www.wowzamedia.com/forums/content.php?233

ModuleOnConnectAuthenticate

http://www.wowzamedia.com/forums/content.php?160

ModuleOnConnectAuthenticate2

http://www.wowzamedia.com/forums/content.php?234

Client-side Requirements

SecureToken on the Client-side

The SecureToken security feature requires changes to your client-side ActionScript player code so that is properly responds to the SecureToken challenge. Below we document how this is done in custom Flash player code. In additional, the SecureToken feature has been integrated into several open source Flash based players such as "JW Player" and "FlowPlayer". For the most up to date information regarding SecureToken and open source players, consult our forums (SecureToken information is in the "Useful Code, Tuning and Troubleshooting" section of the forums):

http://www.wowzamedia.com/forums

To illustrate how to integrate SecureToken into your client-side ActionScript code, let's assume we have configured a server-side application named "securestreaming" that uses the "ModuleSecureToken" to protect publishing and playback of content and that the "secureTokenSharedSecret" it set to "mytestpassword".

The Flash player code to make a secure connection to the server looks like this:

```
import com.meychi.ascryptAS3.TEA;

var nc:NetConnection = new NetConnection();
function ncOnStatus(infoObject:NetStatusEvent)
{
   if (infoObject.info.code == "NetConnection.Connect.Success")
   {
      if (infoObject.info.secureToken != null)
         nc.call("secureTokenResponse", null,
            TEA.decrypt(infoObject.info.secureToken, "mytestpassword"));
   }
}
nc.addEventListener(NetStatusEvent.NET_STATUS, ncOnStatus);
nc.connect("rtmp://localhost/securestreaming");
```

The first line of this example imports the TEA library that is used to decrypt the SecureToken token. ActionScript 2 and 3 versions of this code are included in the "client/com" folder of this package. If you plan on integrating this code into your player, you will need to copy these classes into your Flash code. Next, we define and create a NetConnection object that will be used to communicate with Wowza Media Server. The next function is the NetConnection onStatus handler that will be invoked during the lifecycle of the NetConnection object. Next, we add onStatus handler as a listener to the NetConnection object and finally all NetConnection.connect(url) to connect to Wowza Media Server.

When the NetConnection object establishes a connection with Wowza Media Server, the onStatus handler will be called with an infoObject.info.code value of "NetConnection.Connection.Success". If the server is protected with SecureToken then the infoObject.info object will also contain a SecureToken challenge in the "secureToken" field. To respond to this challenge, the Flash player code calls the remote function "secureTokenResponse" with the first parameter set to the decrypted token. You can see we are decrypting the token with the call:

```
TEA.decrypt(infoObject.info.secureToken, "mytestpassword")
```

This is all that needs to be done to complete the challenge and response cycle. Once the call is made to "secureTokenResponse" the connection is validated and the rest of your Flash code will execute normally. If the "secureTokenResponse" function is not called before you Flash player code calls "play" or "publish" then Wowza Media Server will terminate the connection.

Example Configurations

Below are some example configurations that cover how to combine SecureToken, RTMP authentication, RTSP authentication, StreamNameAlias and RTMPE to protect different forms of streaming.

Video On Demand

This example illustrates how to protect video on demand streaming. Basic configuration:

```
Stream type: default
Add On Packages: ModuleSecureToken, ModuleRequireSecureConnection
```

SecureToken and RTMPE protection are needed to protect video on demand streaming. The steps are to create an application named "vod" are:

1. Create the folder **[install-dir]/applications/vod**

2. Create the folder **[install-dir]/conf/vod** and copy **[install-dir]/conf/Application.xml** into this new folder.

3. Edit the newly copied **Application.xml** file and add the following **<Module>** definitions as the last entries in the **<Modules>** list:

```
<Module>
     <Name>ModuleSecureToken</Name>
     <Description>ModuleSecureToken</Description>
     <Class>com.wowza.wms.plugin.security.ModuleSecureToken</Class>
</Module>
```

4. In the same **Application.xml** add the following properties to the application level properties at the bottom of the file (be sure to get the correct **<Properties>** container – there are several in the file):

```
<Property>
  <Name>secureTokenSharedSecret</Name>
  <Value>#ed%h0#w@1</Value>
</Property>
<Property>
  <Name>requireSecureConnection</Name>
  <Value>true</Value>
  <Type>Boolean</Type>
</Property>
```

You will need to integrate the client side SecureToken response into your Flash player code. Use the rtmpe:// protocol to playback all content.

Live Streaming using RTMP or RTSP/RTP Encoder

This example illustrates how to protect live streaming when using an RTMP or RTSP/RTP based encoder such as Flash Media Live Encoder (consult the EULA for streaming restrictions), Telestream Wirecast, QuickTime Broadcaster, Orban Opticodec and On2 Flix Live. Basic configuration:

```
Stream type: live
Add On Packages: ModuleRTMPAuthenticate, ModuleRequireSecureConnection
RTP/Authentication/Method: digest
publish.password file: add username/password entries for publishing
```

SecureToken, RTMP authentication, RTSP authentication and RTMPE are needed to protect live streaming. The steps to create an application named "live" are:

1. Create the folder **[install-dir]/applications/live**

2. Create the folder **[install-dir]/conf/live** and copy **[install-dir]/conf/Application.xml** into this new folder.

3. Edit the newly copied **Application.xml** file, change the **Streams/StreamType** to **live**, verify the **RTP/Authentication/PublishMethod** is set to **digest** and add the following **<Module>** definitions as the last entries in the **<Modules>** list:

```
<Module>
     <Name>ModuleRTMPAuthenticate</Name>
     <Description>ModuleRTMPAuthenticate</Description>
     <Class>com.wowza.wms.plugin.security.ModuleRTMPAuthenticate</Class>
</Module>
```

4. To enable SecureToken protection, add the following properties to the application level properties at the bottom of the file (be sure to get the correct **<Properties>** container – there are several in the file). If you omit this property SecureToken will not be used to protect playback:

```
<Property>
  <Name>secureTokenSharedSecret</Name>
  <Value>#ed%h0#w@1</Value>
</Property>
<Property>
  <Name>requireSecureConnection</Name>
  <Value>true</Value>
  <Type>Boolean</Type>
</Property>
```

You will need to integrate the client side SecureToken response into your Flash player code. Add a username and password entry into the "[install-dir]/conf/publish.password" file for each user to which you want to grant publishing access. Use the rtmpe:// protocol to playback all content.

Below are a few additional security methods that can be used to secure live streaming. They are mainly used to protect the publishing of live streaming when using an encoder that does not support RTMP authentication:

Secure URL Params

http://www.wowzamedia.com/forums/content.php?233

onConnect Authenticate

http://www.wowzamedia.com/forums/content.php?160

onConnect Authenticate2

http://www.wowzamedia.com/forums/content.php?234

SHOUTcast, RTSP/RTP re-streaming and native RTP

This example illustrates how to protect live streaming when re-streaming SHOUTcast/IceCast, re-streaming RTSP/RTP (such as an IP camera or from Darwin Streaming Server) or a native RTP stream (encoder that uses SDP files). Basic configuration:

```
Stream type: rtp-live
Add On Packages: ModuleSecureToken, ModuleStreamNameAlias,
                        ModuleRequireSecureConnection
RTP/Authentication/Method: digest
rtp.password file: empty
aliasmap.play.txt: default rule: *=${Stream.Name}
aliasmap.stream.txt: entries for valid stream names/urls
```

SecureToken, StreamNameAlias and RTMPE are needed to protect this form of live streaming. You will need to download and install the StreamNameAliasPackage from here:

http://www.wowza.com/forums/content.php?116

The steps are to create an application named "rtplive" are:

1. Create the folder **[install-dir]/applications/rtplive**

2. Create the folder **[install-dir]/conf/rtplive** and copy **[install-dir]/conf/Application.xml** into this new folder.

3. Edit the newly copied **Application.xml** file, change the **Streams/StreamType** to **rtp-live** , verify the **RTP/Authentication/PublishMethod** is set to **digest** and add the following **<Module>** definitions as the last entries in the **<Modules>** list (the order of these two modules is important):

```
<Module>
  <Name>ModuleStreamNameAlias</Name>
  <Description>ModuleStreamNameAlias</Description>
  <Class>com.wowza.wms.plugin.streamnamealias.ModuleStreamNameAlias</Class>
</Module>
<Module>
    <Name>ModuleSecureToken</Name>
    <Description>ModuleSecureToken</Description>
    <Class>com.wowza.wms.plugin.security.ModuleSecureToken</Class>
</Module>
```

4. In the same **Application.xml** add the following properties to the application level properties at the bottom of the file (be sure to get the correct **<Properties>** container – there are several in the file):

```
<!-- ModuleStreamNameAlias Properties -->
<Property>
  <Name>aliasMapFileStream</Name>
  <Value>${com.wowza.wms.context.VHostConfigHome}/conf/aliasmap.stream.txt</Value>
</Property>
<Property>
  <Name>aliasMapFilePlay</Name>
  <Value>${com.wowza.wms.context.VHostConfigHome}/conf/aliasmap.play.txt</Value>
</Property>
<Property>
  <Name>aliasMapPathDelimiter</Name>
  <Value>/</Value>
</Property>
<Property>
  <Name>aliasMapNameDelimiter</Name>
  <Value>=</Value>
</Property>
<Property>
  <Name>aliasMapDebug</Name>
  <Value>true</Value>
  <Type>Boolean</Type>
</Property>

<!-- ModuleSecureToken Properties -->
<Property>
  <Name>secureTokenSharedSecret</Name>
  <Value>#ed%h0#w@1</Value>
</Property>
<Property>
  <Name>requireSecureConnection</Name>
  <Value>true</Value>
  <Type>Boolean</Type>
</Property>
```

5. Edit **[install-dir]/conf/aliasmap.stream.txt** and remove the default rule (***=${Stream.Name}**) and add an entry for each stream name that wish to allow to be streamed through this application. For example to only allow the SDP file myStream.sdp, the IP camera rtsp://mycompany.com/myCamera and the SHOUTcast URL http://mycompany.com/myRadio then set the contents of **aliasmap.stream.txt** to (we are also setting up stream alias for these streams at the same time):

```
myStream=myStream.sdp
myCamera=rtsp://mycompany.com/myCamera
myRadio=http://mycompany.com/myRadio
```

You will need to integrate the client side SecureToken response into your Flash player code. Use the rtmpe:// protocol to playback all content.

Live Stream Repeater (origin/edge)

This section will cover how to setup security with the live stream repeater. The origin is protected using RTMP authentication, RTSP authentication and SecureToken and the edge servers is protected using SecureToken and RTMPE.

Origin Server

Origin server configuration:

```
Stream type: liverepeater-origin
Add On Packages: ModuleRTMPAuthenticate
RTP/Authentication/Method: digest
publish.password file: add username/password entries for publishing
```

SecureToken, RTMP authentication and RTSP authentication are needed to protect live streaming to the origin. The steps to create an application named "liveorigin" are:

1. Create the folder **[install-dir]/applications/liveorigin**

2. Create the folder **[install-dir]/conf/liveorigin** and copy **[install-dir]/conf/Application.xml** into this new folder.

3. Edit the newly copied **Application.xml** file, change the **Streams/StreamType** to **liverepeater-origin**, verify the **RTP/Authentication/PublishMethod** is set to **digest** and add the following **<Module>** definitions as the last entries in the **<Modules>** list:

```
<Module>
     <Name>ModuleRTMPAuthenticate</Name>
     <Description>ModuleRTMPAuthenticate</Description>
     <Class>com.wowza.wms.plugin.security.ModuleRTMPAuthenticate</Class>
</Module>
```

4. In the same Application.xml add the following property to the application level properties at the bottom of the file (be sure to get the correct **<Properties>** container – there are several in the file):

```
<Property>
  <Name>secureTokenSharedSecret</Name>
  <Value>#ed%h0#w@1</Value>
</Property>
```

Add a username and password entry into the "[install-dir]/conf/publish.password" file for each user to which you want to grant publishing access. Use the rtmpe:// protocol to playback all content.

Edge Server

Edge server configuration:

```
Stream type: liverepeater-edge
Add On Packages: ModuleRTMPAuthenticate, ModuleRequireSecureConnection
RTP/Authentication/Method: digest
publish.password file: add username/password entries for publishing
```

SecureToken and RTMPE are needed to protect live streaming to and from the edge server. The steps to create an application named "liveedge" are:

1. Create the folder **[install-dir]/applications/liveedge**

2. Create the folder **[install-dir]/conf/liveedge** and copy **[install-dir]/conf/Application.xml** into this new folder.

3. Edit the newly copied **Application.xml** file, change the **Streams/StreamType** to **liverepeater-edge**, verify the **RTP/Authentication/PublishMethod** is set to **digest** and add the following **<Module>** definitions as the last entries in the **<Modules>** list:

```
<Module>
     <Name>ModuleRTMPAuthenticate</Name>
     <Description>ModuleRTMPAuthenticate</Description>
     <Class>com.wowza.wms.plugin.security.ModuleRTMPAuthenticate</Class>
</Module>
```

4. In the same Application.xml add the following properties to the application level properties at the bottom of the file (be sure to get the correct **<Properties>** container – there are several in the file). The **secureTokenSharedSecret** is the shared secret that will be used when the Flash player connects to play content on this edge. The **secureTokenOriginSharedSecret** is the shared secret used when this edge connects to the origin:

```
<Property>
  <Name>secureTokenSharedSecret</Name>
  <Value>#ed%h0#w@1</Value>
</Property>
<Property>
  <Name>secureTokenOriginSharedSecret</Name>
  <Value>#ed%h0#w@1</Value>
</Property>
<Property>
  <Name>requireSecureConnection</Name>
  <Value>true</Value>
  <Type>Boolean</Type>
</Property>
```

5. Set the **Repeater/OriginURL** to the address of the origin. For example if the origin server is at the address **mycompany.com**, set the **OrginUrl** to:

```
<Repeater>
  <OriginURL>rtmp://mycompany.com/liveorigin</OriginURL>
  <QueryString></QueryString>
</Repeater>
```

You will need to integrate the client side SecureToken response into your Flash player code. Be sure the "[install-dir]/conf/publish.password" is empty so that no users can publish directly to the edge server. Use the rtmpe:// protocol to playback all content.

Example Players

The "Media Security" package includes several different examples of how to integrate SecureToken support into your Flash player ActionScript code. The "client" folder contains ActionScript 3.0 examples and the "clientAS2" folder contains ActionScript 2.0 examples. These examples use the application name "securestreaming". To setup an application to test these examples follow the steps:

1. Create the folder **[install-dir]/applications/securestreaming**

2. Create the folder **[install-dir]/conf/securestreaming** and copy **[install-dir]/conf/Application.xml** into this new folder.

3. Edit the newly copied **Application.xml** file and add the following **<Module>** definition as the last entry in the **<Modules>** list:

```
<Module>
     <Name>ModuleSecureToken</Name>
     <Description>ModuleSecureToken</Description>
     <Class>com.wowza.wms.plugin.security.ModuleSecureToken</Class>
</Module>
```

4. Add the following property to the application level properties at the bottom of the **Application.xml** file (be sure to get the correct **<Properties>** container – there are several in the file):

```
<Property>
  <Name>secureTokenSharedSecret</Name>
  <Value>#ed%h0#w@1</Value>
</Property>
```

HHH