The Wowza Streaming Cloud™ service allows you to secure HLS streams using the external method of AES-128 encryption. When you use the external method of AES-128 encryption, encryption keys are delivered to devices from an external URL. This article describes how to use the Wowza Streaming Cloud web manager to configure AES-128 encryption for an HLS stream.

Configure AES-128 encryption

1. Do one of the following:
   - Click Advanced on the menu bar, click Transcoders, and then click Add Transcoder. Complete the steps to finish adding the transcoder. For more information, see Create and manage transcoders in Wowza Streaming Cloud.
   - Click Advanced on the menu bar, click Transcoders, and then select a transcoder on the Transcoders page.

   **Note:** Transcoders created through the live stream workflow appear as [Live stream name] / Transcoder.

2. Click the Properties tab and then click Edit.
3. Under HLS, select Enabled for HLS AES128 Host and enter the URL that devices will use to fetch the key to decrypt the stream.
4. Under HLS, select Enabled for HLS AES128 Secret and enter a 16-byte key that will be used to decrypt the stream. The key must be 32 characters in length and can only contain hex characters (a-z, A-Z, 0-9). The key must match the key returned by the HLS AES128 Host.
5. Click Save.

Examples
The following examples show how to use popular web application technologies such as **ASP.NET**, **JSP**, and **PHP** to send the key data. Each example includes a Boolean `isValid` value that defaults to **true**. You can modify the examples to provide your own security tests to validate that users can access the content. If users shouldn’t be allowed to access the content, you can block them from receiving the decryption key by setting `isValid` to **false**.

If the request for the key returns a status of **403**, then the device won’t be able to decrypt and play the stream. If the key is returned, then the stream will be decrypted and played. Require HTTPS access to this key to ensure that it isn’t sent over an unsecured connection on the Internet.

The key being sent in these examples is `DE51A7254739C0EDF1DCE13BBB308FF0`. You should substitute this value with a different 16-byte key. The key should match the key entered in the **HLS AES128 Secret** field on the transcoder **Properties** tab.

**Note:** These examples are provided as-is with no expressed warranty. You can modify or distribute them without restriction.

### ASP.NET example

```csharp
<%@ Page Language="C#" %>
<%

    Boolean isValid = true;
    if (!isValid)
    {
        Response.Status = "403 Forbidden";
    }
    else
    {
        Response.AddHeader("Content-Type", "binary/octet-stream");
        Response.AddHeader("Pragma", "nocache");

        String keyStr = "DE51A7254739C0EDF1DCE13BBB308FF0";

        int len = keyStr.Length/2;
        byte[] keyBuffer = new byte[len];

        for (int i=0; i
```

### JSP example

```jsp
<%@ Page Language="C#" %>
<%

    Boolean isValid = true;
    if (!isValid)
    {
        Response.Status = "403 Forbidden";
    }
    else
    {
        Response.AddHeader("Content-Type", "binary/octet-stream");
        Response.AddHeader("Pragma", "nocache");

        String keyStr = "DE51A7254739C0EDF1DCE13BBB308FF0";

        int len = keyStr.Length/2;
        byte[] keyBuffer = new byte[len];

        for (int i=0; i
PHP example

More resources

- Test AES encryption for HLS streams from Wowza Streaming Cloud