When you broadcast an HLS stream using the Wowza Streaming Cloud™ service, you have complete control over whether or not Secure Socket Layer (SSL) is used to establish a handshake for encrypting the HTTP connections that deliver the stream to viewers. By default, Wowza CDN on Akamai and custom stream targets with a provider of akamai_cupertino use relative playlists, which allow streams to be played over HTTP or HTTPS—or both. This provides the greatest flexibility for your viewers. You can, however, require that viewers watch a stream from a Wowza CDN on Akamai target over HTTP or HTTPS. Requiring viewers to watch a stream over HTTPS ensures an encrypted connection during playback. You can also configure Wowza Streaming Cloud to send the stream from the transcoder to the target using SSL, allowing encryption during this data transfer.

**Note:** This article applies to Wowza CDN on Akamai and custom stream targets only. To manage HLS playback over SSL for Wowza CDN on Fastly stream targets, see Manage HLS playback over SSL for Wowza CDN on Fastly with the Wowza Streaming Cloud REST API.

### About SSL playback from Wowza Streaming Cloud stream targets

Streams delivered from Wowza Streaming Cloud travel across the internet in two stages to reach their audiences. After Wowza Streaming Cloud transcodes (or passes through) the encoded live source video, it sends the stream to geographically distributed servers. Those servers then deliver the stream to viewers, such as through a hosted webpage or a direct playback URL.

Wowza Streaming Cloud uses the HTTP protocol to make these two outbound network transfers, delivering streams for playback over HLS or over HLS and HDS. For both types of HTTP delivery, Wowza Streaming Cloud generates a relative...
playlist, which means that the stream can be viewed over HTTP or HTTPS. The viewer is not restricted to the secure or unsecured protocol.

When you choose to deliver a stream over Wowza CDN on Akamai stream targets or custom stream targets with the provider `akamai_cupertino`, you can enjoy the default flexibility of relative playlists, or you can control whether the viewer has to use HTTP or HTTPS to watch the stream by editing the stream target’s properties.

### Stream target SSL playback cheat sheet

Three stream target properties are used to manage SSL playback for HLS streams. Here’s a table that summarizes how to configure those properties for the type of playback you want to support.

**Tip:** Click the desired result to jump to instructions on how to configure the target with these settings.

<table>
<thead>
<tr>
<th>Desired result</th>
<th>Send stream to target over SSL</th>
<th>Play over SSL</th>
<th>Relative Playlists</th>
</tr>
</thead>
<tbody>
<tr>
<td>Play HLS over HTTP or HTTPS</td>
<td>True*</td>
<td>False (default)</td>
<td>True (default)</td>
</tr>
<tr>
<td>Play HLS over HTTPS</td>
<td>True*</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>Play HLS over HTTP</td>
<td>False (default)*</td>
<td>False (default)</td>
<td>False</td>
</tr>
</tbody>
</table>

* Sending the stream from the transcoder to the target over SSL isn’t required for secure playback, but it’s good practice to use SSL for the connection in and out of the target to keep communications as secure as possible.

### Deliver an HLS stream for playback over HTTP or HTTPS

You can create a transcoder with outputs and stream targets that play HLS over
1. Using the Wowza Streaming Cloud REST API, create a transcoder. Depending on your streaming setup needs, you can specify a delivery_method of **push** or **pull** and a transcoder_type of **passthrough** or **transcoded** (ABR).

Example request and response

Create a transcoder

```bash
curl -X POST \
-H "Content-Type: application/json" \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
-d '{
    "transcoder": {
        "billing_mode": "pay_as_you_go",
        "broadcast_location": "us_west_california",
        "buffer_size": 4000,
        "delivery_method": "push",
        "name": "MyABRtranscoder",
        "protocol": "rtmp",
        "transcoder_type": "transcoded"
    }
}' "${WSC_HOST}/api/${WSC_VERSION}/transcoders"
```

The request creates a transcoder and returns a response that includes details of the transcoder, including its unique transcoder ID. There are no outputs associated with the transcoder yet. Make note of the ID, which is the transcoder ID that Wowza Streaming Cloud creates for your stream. You'll need it to associate your transcoder with output renditions and stream targets. The response should look something like this:
2. Complete the transcoder by adding outputs and stream targets, but keep the following in mind:
   - Create either a custom stream target or a Wowza CDN on Akamai stream target.
   - Ensure that your stream target has a provider of `akamai_cupertino`. This will enable the `relativePlaylists` stream target property by default, allowing viewers to play your stream over HTTP or HTTPS.
For instructions on adding output renditions and stream targets, see one of the following articles, depending on whether you’re creating an adaptive bitrate or passthrough transcoder:
   - Create an ABR stream and send it to a target with the Wowza Streaming Cloud REST API
   - Pass a stream through the transcoder to a target with the Wowza Streaming Cloud REST API

3. Refer to the `hls_playback_url` value returned in the stream target details. Use this URL to play the stream in a browser or player that supports HLS. Although
the playback URL appears as HTTPS, the playback URL can also be accessed using HTTP.
For example:
https://[wowzasubdomain]-f.akamaihd.net/i/32a5814b_1@7217/master.m3u8

Deliver an HLS stream for playback over HTTPS only

You can require SSL for HLS playback, if desired. This ensures that viewer clients connect securely to view the stream. You can also optionally send the stream from the transcoder to the target using SSL.

Create a transcoder with outputs and stream targets that play HLS over HTTPS only.

1. Using the Wowza Streaming Cloud REST API, create a transcoder. Depending on your streaming setup needs, you can specify a `delivery_method` of `push` or `pull` and a `transcoder_type` of `passthrough` or `transcoded` (ABR). For instructions, see steps 1 and 2 of Deliver an HLS stream that can be played over HTTP or HTTPS.

2. Configure properties for the stream target or stream targets that you associated with your transcoder.

Configure the following properties for the stream target one at a time. The `sendSSL` property is optional but recommended.

<table>
<thead>
<tr>
<th>Key (property name)</th>
<th>Section</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sendSSL</td>
<td>hls</td>
<td>true</td>
<td>When true, Wowza Streaming Cloud sends the stream from the transcoder to the target using SSL (HTTPS).</td>
</tr>
<tr>
<td>playSSL</td>
<td>playlist</td>
<td>true</td>
<td>When true, the target sends the stream to the player using SSL (HTTPS).</td>
</tr>
</tbody>
</table>

When false, the viewer’s playback protocol (HTTP or HTTPS) must
Example request and response

Configure a property for a stream target

```bash
curl -X POST \
-H "Content-Type: application/json" \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
-d '{
  "property": {
    "key": "playSSL",
    "section": "playlist",
    "value": true
  }
}' "${WSC_HOST}/api/${WSC_VERSION}/stream_targets/[stream_target_id]/properties"
```

This request assigns the value **true** to the **playSSL** stream target property. The details of the target’s property are listed in the response, which should look something like this:

```json
{
  "property": {
    "key": "playSSL",
    "section": "playlist",
    "value": true
  }
}
```

For more information about configuring stream target properties and related requests, see [How to set advanced properties using the Wowza Streaming Cloud REST API](#).

3. For playback over HTTPS only, refer to the **hls_playback_url** value returned in the stream target details. Use this URL to play the stream in a browser or player that supports HLS. The playback URL can only be accessed using HTTPS.

For example:

https://[wowzasubdomain]-f.akamaihd.net/i/32a5814b_1@7217/master.m3u8
Deliver an HLS stream for playback over HTTP only

You can create a transcoder with outputs and stream targets that play HLS over HTTP only.

1. Using the Wowza Streaming Cloud REST API, create a transcoder. Depending on your streaming setup needs, you can specify a delivery_method of **push** or **pull** and a transcoder_type of **passthrough** or **transcoded** (ABR).
   For instructions, see steps 1 and 2 of Deliver an HLS stream that can be played over HTTP or HTTPS.
2. Configure properties for the stream target or stream targets that you associated with your transcoder.

Configure the following properties for the stream target one at a time. The sendSSL property is optional, and the sendSSL and playSSL properties are false by default. If you have already set them to true, you can configure them to false using the example request below to reset the value.

<table>
<thead>
<tr>
<th>Key (property name)</th>
<th>Section</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sendSSL</td>
<td>hls</td>
<td>false</td>
<td>When false, Wowza Streaming Cloud sends the stream from the transcoder to the target using HTTP.</td>
</tr>
<tr>
<td>playSSL</td>
<td>playlist</td>
<td>false</td>
<td>When false, the target sends the stream to the player using HTTP.</td>
</tr>
<tr>
<td>relativePlaylists</td>
<td>playlist</td>
<td>false</td>
<td>When false, the viewer’s playback protocol (HTTP or HTTPS) must correspond to the target’s specified playback protocol.</td>
</tr>
</tbody>
</table>

**Example request and response**
Configure a property for a stream target

This request assigns the value `false` to the `relativePlaylists` stream target property. The details of the target's property are listed in the response, which should look something like this:

```json
{
    "property": {
        "key": "relativePlaylists",
        "section": "playlist",
        "value": false
    }
}
```

For more information about configuring stream target properties and related requests, see How to set advanced properties using the Wowza Streaming Cloud REST API.

More resources

- Security features in Wowza Streaming Cloud
- Send streams securely to Wowza CDN using the Wowza Streaming Cloud REST API