Manage HLS playback over SSL for Wowza CDN on Fastly with the Wowza Streaming Cloud REST API

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 Streaming Cloud uses relative playlists, which allows 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 live stream over HTTPS. Requiring viewers to watch a stream over HTTPS ensures an encrypted connection during playback.

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

About SSL playback from Wowza CDN on Fastly stream targets

Live 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 called stream targets. Those Wowza CDN on Fastly targets then deliver the live 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. For this 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 secured or unsecured protocol.

When you choose to deliver a live stream via Wowza CDN on Fastly stream targets, you can enjoy the default flexibility of relative playlists, or you can control whether the viewer has to use HTTPS to watch the stream.

HLS stream target SSL playback cheat sheet

The following table summarizes how to configure the type of playback you want to support.

<table>
<thead>
<tr>
<th>Desired result:</th>
<th>Force SSL Playback:</th>
<th>Relative playlists:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Play HLS over HTTP or HTTPS</td>
<td>False (default)</td>
<td>True (default)</td>
</tr>
<tr>
<td>Play HLS over HTTPS</td>
<td>True</td>
<td>False</td>
</tr>
</tbody>
</table>

**Tip:** Click the desired result to jump to instructions on how to configure the target with these settings.
Deliver an HLS stream for playback over HTTP or HTTPS

Live stream workflow — HTTP or HTTPS

The easiest option is to create a live stream that plays HLS over HTTP or HTTPS.

1. Using the Wowza Streaming Cloud REST API, create a live stream.

Example request and response

```bash
curl -X POST \
-H "Content-Type: application/json" \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
-d '{
  "live_stream": {
    "aspect_ratio_height": 720,
    "aspect_ratio_width": 1280,
    "billing_mode": "pay_as_you_go",
    "broadcast_location": "us_west_california",
    "delivery_method": "push",
    "encoder": "wowza_gocoder",
    "hosted_page": true,
    "hosted_page_sharing_icons": true,
    "name": "MyLiveStream",
    "player_countdown": false,
    "player_responsive": true,
    "player_type": "original_html5",
    "target_delivery_protocol": "hls-https",
    "transcoder_type": "transcoded"
  }
}'

${WSC_HOST}/api/${WSC_VERSION}/live_streams
```

The request creates a live stream with an id parameter, an associated player, and a hosted page. The details of the live stream’s configuration are listed in the response, which should look something like this:
Note: Initially, the response for the player embed code and hosted page URL indicates that they're "in progress." To get the code and URL, use a GET request to retrieve the live stream's details.

```bash
curl -X GET \
-H "wsc-api-key: ${WSC_API_KEY}" \ 
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
"${WSC_HOST}/api/${WSC_VERSION}/live_streams/[live_stream_id]"
```

The POST /live_streams request automatically creates a Wowza CDN on Fastly stream target with relative playlists, meaning that viewers can watch the stream over HTTP or HTTPS, whichever protocol their browser calls.

2. Depending on how you want to make the stream available, do one of the following using details from the live stream.

- Use the HLS playback URL returned as the `player_hls_playback_url` value 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.wowza.com/1/abc1TnJwZEpwYXxy/a12BeGd1/hls/live/playlist.m3u8

- If you created a hosted page, use the `hosted_page_url` value to share the hosted page URL. Although the hosted page URL appears as HTTPS, the page can also be viewed using HTTP.
  For example:
  https://player.cloud.wowza.com/hosted/[player_id]/player.html
If you want to include the stream on an external website (HTTP or HTTPS), use the `player_embed_code` value, which uses a relative URL in the JavaScript call for the player. For example:

```html

Transcoder workflow — HTTP or HTTPS

For a more modular API workflow with detailed control over configuration, you can create a transcoder with outputs and stream targets that play HLS over HTTP or HTTPS.

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

```
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 Wowza CDN on Fastly stream targets. 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}.wowza.com/l/abc1TnJwZEpwYXxy/a12BeGd1/hls/live/playlist.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 over HTTPS.

Live stream workflow –HTTPS only

The simpler option for HTTPS-only playback is to create a live stream that plays HLS over HTTPS only.

1. Using the Wowza Streaming Cloud REST API, create a live stream.
2. In the response to the POST /live_streams request, note the stream target id for the next step.
3. Update the Wowza CDN on Fastly targets associated with your transcoder to enable `force_ssl_playback`.
4. Configure the `relativePlaylists` property for the stream target associated with your live stream. See Configure the `relativePlaylist` property to deliver an HLS stream for playback over HTTPS only.

Example request

Update a Wowza CDN on Fastly stream target
Transcoder workflow — HTTPS only

For a more modular API workflow with detailed control over configuration, you can 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 Transcoder workflow — HTTP and HTTPS.

2. When you create Wowza CDN on Fastly stream targets for your transcoder, set force_ssl_playback to true.

3. Configure the relativePlaylists property for the stream target associated with your transcoder. See Configure the relativePlaylist property to deliver an HLS stream for playback over HTTPS only.

Example request

Create a Wowza CDN on Fastly stream target

```
curl -X POST \
-H "Content-Type: application/json" \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
-d '{
  "stream_target_fastly": {
    "name": "My secure playback Fastly target",
    "force_ssl_playback": true
  }
}
"${WSC_HOST}/api/${WSC_VERSION}/stream_targets/fastly"
```

Configure the relativePlaylist property to deliver an HLS stream for playback over HTTPS only

```
curl -X POST \
-H "Content-Type: application/json" \
-H "wsc-access-key: [key]" \
-H "wsc-timestamp: [timestamp]" \
-H "wsc-signature: [signature]" \
-d '{
  "property": {
    "key": "relativePlaylists",
    "section": "playlist",
    "value": false
  }
}
"${WSC_HOST}/api/${WSC_VERSION}/stream_targets/[stream_target_id]/properties"
```

Example request and response

Configure a property for a stream target

```
curl -X POST \
-H "Content-Type: application/json" \
-H "wsc-access-key: [key]" \
-H "wsc-timestamp: [timestamp]" \
-H "wsc-signature: [signature]" \
-d '{
  "property": {
    "key": "relativePlaylists",
    "section": "playlist",
    "value": false
  }
}
"${WSC_HOST}/api/${WSC_VERSION}/stream_targets/[stream_target_id]/properties"
```

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

**Playback over HTTPS only**

### Live stream workflow

Do one of the following for playback over HTTPS only, depending on how you want to make your live stream available:

- Use the HLS playback URL returned as the `player_hls_playback_url` value 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].wowza.com/1/abc1TnJwZEpwYXxy/a12BeGd1/hls/live/playlist.m3u8
  ```

- If you created a hosted page, use the `hosted_page_url` value to share the hosted page URL. The hosted page can only be viewed using HTTPS. For example:
  
  ```
  https://player.cloud.wowza.com/hosted/[player_id]/player.html
  ```

- If you want to include the stream on an external website (HTTPS), use the `player_embed_code` value, which uses a relative URL in the JavaScript call for the player. For example:
  
  ```
  \n
  ```

### Transcoder workflow

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].wowza.com/1/abc1TnJwZEpwYXxy/a12BeGd1/hls/live/playlist.m3u8
```

**More resources**

- [Security features in Wowza Streaming Cloud](#)
- [Send streams securely to Wowza CDN using the Wowza Streaming Cloud REST API](#)