You can use the Wowza Streaming Cloud™ service REST API to restrict access to a stream by applying token authentication to a Wowza CDN on Fastly stream target. Token authentication allows only viewers who have the token, which is hashed and appended to the playback URL, to access the stream. You can use token authentication with Wowza CDN on Fastly stream targets to make the stream playback URL unavailable after a certain length of time, to limit access to approved IP addresses, or apply other restrictions. A common use is to protect pay-per-view content to only paying viewers.

**Note:** This article applies to Wowza CDN on Fastly stream targets only. To secure Wowza CDN on Akamai stream targets with token authentication, see Protect streams with token authorization using the Wowza Streaming Cloud REST API.

### About token authentication

Token-based authentication uses a multipart token that consists of a delimited list of string fields. One field is an HMAC, or keyed-hash message authentication code. HMAC is a common mechanism for message authentication that uses cryptographic hash functions. The HMAC portion of the token hashes a trusted shared secret that you create in Wowza Streaming Cloud. It is short-lived and secures initial access to the stream.

The second part of the token, a cookie, is valid for the duration of the stream and protects segments that are delivered during playback. It restricts access to the stream according to query parameters that you specify. For example, you can expire the stream after a certain length of time or only allow whitelisted IP addresses to access it.

You append the token to the stream target’s playback URL, and then Wowza Streaming Cloud only lets viewers receive the content after it verifies the presence and validity of the token.

Token authentication is managed by the browser. No configuration is required for the player. However, token authentication requires that the viewer’s browser supports cookies.

**Notes:**
- Token authentication works with third-party players and with Wowza Player Builder. It doesn’t work with a player created in the Wowza Streaming Cloud live stream workflow and embedded in a hosted or third-party webpage.
- If using Wowza Player Builder, enable the `withCredentials` configuration property. See Customize Wowza Player with configuration properties for more information.

### Create a Wowza CDN on Fastly stream target with token authentication enabled

When you create a Wowza CDN on Fastly stream target, you can configure it for token authentication.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
</table>

...
<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>string</td>
<td>A descriptive name for the stream target. Maximum 255 characters.</td>
</tr>
<tr>
<td>token_auth_enabled</td>
<td>Boolean</td>
<td>If true, token authentication is enabled. If false, it’s disabled. The default is false.</td>
</tr>
<tr>
<td>token_auth_shared_secret</td>
<td>string</td>
<td>A trusted, shared secret for token authentication. It must contain only hexadecimal characters and be an even number of characters equal to or less than 32. If you enable token authentication but don’t set a token_auth_shared_secret value, the Wowza Streaming Cloud service generates the value for you.</td>
</tr>
<tr>
<td>token_auth_playlist_only</td>
<td>Boolean</td>
<td>If true, Wowza Streaming Cloud protects the master playlist only and leaves individual media playlists and media segments unprotected. If false, the master playlist, media playlists, and media segments are all protected. This feature enables playback compatibility with media players that don’t support the withCredentials property. It may also be useful when addressing token auth compatibility issues with specific browsers. The default is false.</td>
</tr>
</tbody>
</table>

Example request and response

The following request generates a Wowza CDN on Fastly stream target with token authentication:

```
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 Target",
  "token_auth_enabled": true,
  "token_auth_shared_secret": "12345678ABCDEF",
  "token_auth_playlist_only": true
}}' ${WSC_HOST}/api/${WSC_VERSION}/stream_targets/fastly
```

The details of the configured target are listed in the response, which should look something like this:

```
{
  "stream_target_fastly": {
    "id": "1234abcd",
    "name": "My Secure Target",
    "state": "activated",
    "stream_name": "st1r2eam",
    "playback_url": "https://[subdomain].wowza.com/1/[stream_id]/[stream_name]/his/live/playlist.m3u8",
    "token_auth_enabled": true,
    "token_auth_shared_secret": "12345678ABCDEF",
    "token_auth_playlist_only": true,
    "geoblock_enabled": false,
    "geoblock_by_location": "disabled",
    "geoblock_ip_override": "disabled",
    "force_ssl_playback": false,
    "created_at": "2019-09-23T16:04:23.170Z",
    "updated_at": "2019-09-23T16:04:23.170Z"
  }
}
```
Update the token authentication applied to a stream target:

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
-d '{
  "stream_target_fastly": {
    "token_auth_enabled": false
  }
}' "${WSC_HOST}/api/${WSC_VERSION}/stream_targets/fastly/[fastly_stream_target_id]"
```

Assign the stream target to a transcoder

Assign the token-authenticated stream target to a transcoder's output rendition(s).

**Note:** For an adaptive bitrate transcoder, you must assign the target to all of the transcoder's output renditions.

Add stream target parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td>The unique alphanumeric string that identifies the output rendition that will deliver content to the stream target. You can find the ID in the details of the output's transcoder.</td>
</tr>
<tr>
<td>stream_target_id</td>
<td>string</td>
<td>The unique alphanumeric string that identifies the stream target. You can find the ID in the details of the Wowza CDN on Fastly stream target you just created.</td>
</tr>
<tr>
<td>transcoder_id</td>
<td>string</td>
<td>The unique alphanumeric string that identifies the transcoder.</td>
</tr>
</tbody>
</table>

Example request and response

The following request adds the token-authorized stream target 1234abcd to the output rendition whose ID is 5678efgh for the transcoder 9012ijkl.

```
curl -X POST \
-H "Content-Type: application/json" \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
-d '{
  "output_stream_target": {
    "stream_target_id": "1234abcd"
  }
}' "${WSC_HOST}/api/${WSC_VERSION}/transcoders/9012ijkl/outputs/5678efgh/output_stream_targets/"
```

The details of the configured target are listed in the response, which should look something like this:

```json
{
  "output_stream_target": {
    "stream_target_id": "1234abcd",
    "use_stream_target_backup_url": false
  }
}
```
Generate the hashed token

Once enabled, your playback URLs need to include the `hdnts` query parameter for playback to work. You can write a query in C, Java, PHP, Ruby, or other languages. We've provided a Wowza CDN on Fastly Token Authentication examples GitHub repository to help you get started. Various parameters can be used along with the `token_auth_shared_secret`. To learn more, see the code example README.

Attach the hashed token to the URL of your stream using this format:

```
[playback_URL]?hdnts=[parameters_and_hashed_token]
```

For example:

```
https://[subdomain].wowza.com/[stream_id]/[stream_name]/hls/live/playlist.m3u8?
hdnts=exp=1578424041~hmac=0428782df32a8a8b91823889756d8084997c45c58375d526dc9852808b35721
```