CDN 2.2

Overview

CDN 2.2 implements the following features in addition to CDN 2.0 and 2.1:

- CDN streams access control

CDN streams access control

CDN 2.2 allows to resctrict subscribers access to certain streams in CDN. Access is restricted by keys combined to access control list (ACL) for every stream. Subscriber can request stream playback by setting a valid key only. A key should be set in a special parameter passed to the server when establishing connection.

Any new stream is publishing on Origin server as accessible to all subscribers by default. Keys setting and futher ACL management should be done on Origin server using REST API.

Any stream ACL changes are send to all CDN server immediatly, and applied by server as received. Therefore stream playback will be immediatly stopped with Resctricted access error if the subscriber key is not valid anymore.

After stopping a stream, when stream is published again with the same name, it is created with clean ACL, i.e. it is accessible to all the subscribers.

Configuration

To set subscribers access key parameter name, the following setting should be done in flashphoner.properties file on all CDN servers

```
client_acl_property_name=aclAuth
```

In this case access key parameter will be named `aclAuth`

To make sure key values will not be intercepted, it is recommended to enable SSL usage in CDN with the following setting

```
wcs_agent_ssl=true
```

Stream ACL management using REST API

REST API is used to manage stream access keys on Origin server

REST query sho;ud be HTTP/HTTPS POST request as follows:

- HTTP: `http://test.flashphoner.com:8081/rest-api/cdn/stream/access_list/add`
- HTTPS: `https://test.flashphoner.com:8444/rest-api/cdn/stream/access_list/add`

Where:

- `test.flashphoner.com` - WCS server address
- `8081` - standard REST / HTTP port of WCS server
- `8444` - standard HTTPS port
- `rest-api` - mandatory part of URL
- `/rest-api/cdn/stream/access_list/add` - REST query used

REST queries and response states

REST query	Request body	Response body	Response status	Description

`/cdn/stream/access_list/add`	`{`
	`"name":"test",`
	`"keys":["key1","key2","key3"]`
	`}`
			200 OK 400 Bad request 404 Stream not found 500 Internal Server Error	Add keys to stream ACL

`/cdn/stream/access_list/remove`	`{`
	`"name":"test",`
	`"keys":["key1"]`
	`}`
			200 OK 400 Bad request 404 Stream not found 500 Internal Server Error	Remove keys from stream ACL

REST query	Request body	Response body	Response status	Description

`/cdn/stream/access_list/delete`	{
	"name":"test"
	}
			200 OK 400 Bad request 404 Stream not found 500 Internal Server Error	Clean stream ACL

`/cdn/stream/access_list/print`	{
	"name":"test"
	}
		[
		"key1"
		]
			200 OK 404 Stream not found 500 Internal Server Error	Show stream ACL

Parameters

Parameter	Description	Example
name	CDN stream name	`test`
keys	Stream access keys list	`["key1","key2","key3"]`

Access to stream for subscribers

Passing a key using WebSDK to play stream via WebRTC

To play stream via WebRTC stream access key should be passed as custom parameter to `createSession` function

```
Flashphoner.createSession({urlServer: "wss://test.flashphoner.com:8443",
custom: {aclAuth: "key1"}}).on(SESSION_STATUS.ESTABLISHED, function(session){
    ...
});
```

Parameter name should match to configuration setting, `aclAuth` in this case

Passing a key as connection parameter to play stream via RTMP

To play stream via RTMP stream access key should be passed in RTMP connection parameters

```
rtmp://edge1.flashphoner.com:1935/live?aclAuth=key1/test
```

Where

- `edge1.flashphoner.com` - server name
- `aclAuth` - parameter name as set in configuration
- `test` - stream name

Passing a key as stream parameter to play stream via RTMP

To pass access key in stream parameters, the following setting in flashphoner.properties file on all Edge servers should be set

```
rtmp_use_stream_params_as_connection=true
```

In this case access key can be passed as follows

```
rtmp://edge1.flashphoner.com:1935/live/test?aclAuth=key1
```

Passing a key to play stream via HLS

To play stream as HLS, access key should be passed in s

tream URL

```
https://edge1.flashphoner.com:8445/test/test.m3u8?aclAuth=key1
```

Where

- `edge1.flashphoner.com` - server name
- `aclAuth` - parameter name as set in configuration
- `test` - stream name

When HLS ABR is used, access key should be passed in master playlist URL

```
https://edge1.flashphoner.com:8445/test/index.m3u8?aclAuth=key1
```

In this case access key will be set to master playlist for every variant

```
#EXTM3U
#EXT-X-STREAM-
INF:BANDWIDTH=1000000,RESOLUTION=1280x720,CODECS="avc1.42e01f,mp4a.40.2"
../test-720p/test-720p.m3u8?aclAuth=key1
#EXT-X-STREAM-
INF:BANDWIDTH=1000000,RESOLUTION=852x480,CODECS="avc1.42e01f,mp4a.40.2"
../test-480p/test-480p.m3u8?aclAuth=key1
#EXT-X-STREAM-
```

```
INF:BANDWIDTH=400000,RESOLUTION=426x240,CODECS="avc1.42e01f,mp4a.40.2"
../test-240p/test-240p.m3u8?aclAuth=key1
```

Passing a key to play stream via RTSP

Since build 5.2.1008 access key should be passed in stream URL to play stream as RTSP

```
rtsp://edge1.flashphoner.com:5554/test?aclAuth=key1
```

Where

- `edge1.flashphoner.com` - server name
- `aclAuth` - parameter name as set in configuration
- `test` - stream name

Note that some players, VLC for example, pass a key in uppercase only. Use the following parameter to play stream via RTSP in such player

```
case_sensitive_auth_match=false
```

Passing a key using Android SDK to play stream via WebRTC

To play stream via WebRTC in Android application, access key should be passed to `Connection.setCustom` method while connection establishing

```
private Session session;
private Connection connection;
...
connection = new Connection();
connection.setCustom("aclAuth", "key1");
session.connect(connection);
```

Parameter name should match to configuration setting, `aclAuth` in this case

Backward compatibility

CDN 2.2 is compatible to CDN 2.0 and 2.1, only streams with clean ACL (as just published) can be played.

Known limits

1. It is strongly not recommended to publish streams with same name to two Origin servers in the same CDN.
2. A stream published to one of Origin servers should be played on the same Origin server or any Edge server (through Transcoder server if necessary), but should not be played

from another Origin server in the same CDN.

3. ACL can be set to a stream only on Origin server where this stream is published.