

EZDRM Integration NGINX Kaltura VOD Module

Table of Contents

Introduction	3
Requirements	3
Architecture	3
Widevine/PlayReady Universal DRM	4
<i>Obtaining a Content ID</i>	4
<i>Setup Instructions</i>	7
Apple FairPlay DRM for HLS	9
<i>Obtaining an Asset ID</i>	9
<i>Setup Instructions</i>	10
References	11
Additional Information	12

Version 1.1 / Updated May 14, 2020

Introduction

The following document outlines the steps necessary in order to successfully integrate EZDRM on top of an existing Kaltura NGINX installation.

This guide includes information on how to configure Kaltura's open-source [nginx-vod-module](#), in particular with the [vod drm request uri](#) directive, for on-the-fly repackaging, and encryption.

Requirements

1. A backend server or cluster hosting the video content in h264 format
2. NGINX Server compiled with kaltura vod module
3. Some previous knowledge of Linux, Nginx, and Kaltura
4. Basic understanding of HLS, MPEG/DASH, and other streaming protocols and formats

Architecture

The main three components in this solution are:

1. Backend server with mp4 media files (In our example Big Buck Bunny server)
2. Nginx server with kaltura-vod-module
3. EZDRM backend server for encryption-keys and license retrieval

The Kaltura nginx-vod-module includes support for on-the-fly DRM encryption. When the client (end-user) requests an HLS playlist or an MPEG/DASH manifest, the nginx kaltura vod module can repackage remotely stored files on-the-fly into HLS or DASH segments. The nginx vod module can also encrypt the chunks and sign them using DRM (in our case EZDRM).

For example: if we upload an encoded MP4 file to Amazon S3 or Google Cloud Storage, we will then be able to repackage this file on-the-fly to HLS or MPEG/DASH using the Kaltura nginx module. While repackaging, we will be able to use EZDRM services to encrypt this file with DRM.

This guide will help you configure the NGINX kaltura vod plugin to encrypt the MP4 content on-the-fly using keys provided by EZDRM.

Widevine/PlayReady Universal DRM

Obtaining a Content ID

To request the DRM keys from EZDRM to package the media, there are two options, you can call the EZDRM web service in a browser, or you can script this process with curl or other web service calls.

Option 1: Request DRM keys using EZDRM Web Service

1. Call the EZDRM web service in a browser:
<https://wvm.ezdrm.com/ws/LicenseInfo.asmx?op=GenerateKeys>
2. Generate Key values by entering the parameters values and clicking "Invoke".

The parameters are as follows:

Parameter	Description
u	EZDRM username
p	EZDRM password
c	Content_ID **optional

Note: The Content_ID is optional. The first time you use this web service it will be blank. For additional calls it can be blank for new keys or use an existing Content_ID. Sending a Content_ID will allow you to encrypt content with the same DRM values as other content and have that content share one license. If you don't send this value, the web service will automatically generate a unique Content_ID. If you call a Content_ID, you will get the back all the DRM key information for that Content_ID.

3. The response from EZDRM will look like this:

```

<EZDRM xmlns="">
  <WideVine diffgr:id="WideVine1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <ContentID>6IxXXx0Z8xXXXXXXXXXXLbg==</ContentID>
    <Key>W5XXXXXXXXXZhxTjhXXXXXvw==</Key>
    <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXX56bf </KeyHEX>
    <KeyID>WVXXXXXXXXliBEXXw+XXXXX==</KeyID>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXb</KeyIDGUID>
    <KeyIDHEX>5XXXXXXXX36d85XXXXXXXXXXXXb24XXXX</KeyIDHEX>
    <PSSH>
      EhXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3JXXXXX==
    </PSSH>
    <ServerURL>https://widevine-dash.ezdrm.com/proxy?pX=XXXXXX</ServerURL>
    <ServerGet>
      request={"policy": "", "tracks": [ {"type": "SD"}], "content_id": "6IxXXx0Z8xXXXXXXXXXXLbg==" }
    </ServerGet>
    <ResponseRaw>
      { "status": "OK", "drm": [ { "type": "WIDEVINE", "system_id": "edef8ba979d64acea3c827dcd51d21ed" }, { "type": "SD", "key_id": "WVXXXXXXXXliBEXXw+XXXXX==", "key": "WVXXXXXXXXliBEXXw+XXXXX==", "pssh": [ { "drm_type": "WIDEVINE", "data": "EhXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3JXXXXX==" } ] } ] }
    </ResponseRaw>
  </WideVine>
  <PlayReady diffgr:id="PlayReady1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <Key>W5XXXXXXXXXZhxTjhXXXXXvw==</Key>
    <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXX56bf </KeyHEX>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXb</KeyIDGUID>
    <LAURL>
      https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXXX
    </LAURL>
    <Checksum>1Xq+XXXXXX0==</Checksum>
  </PlayReady>
</EZDRM>

```

Option 2: Request DRM keys with curl

The second option to request DRM keys from EZDRM is to script the process with curl or another web service call.

Using EZDRM's web service, the curl script below retrieves the DRM values from the web service.

```
curl -v 'http://wvm.ezdrm.com/ws/LicenseInfo.aspx/GenerateKeys?U=EZDRM_USERNAME&P=EZDRM_PASSWORD&C=''''
```

The parameters are as follows:

Parameter	Description
U	EZDRM username
P	EZDRM password
C	Content_ID <i>**optional, for blank pass ""</i>

Note: Although Content_ID is optional you must pass a "" for blank if you do not specify a Content_ID.

The following is returned from the web service:

```
<EZDRM xmlns="">
  <WideVine diffgr:id="WideVine1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <ContentID>6lxXXx0Z8xXXXXXXXXXlbg==</ContentID>
    <Key>W5XXXXXXXXZHxTjhXXXXvw==</Key>
    <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXX56bf </KeyHEX>
    <KeyID>WVXXXXXXXXliBEXXw+XXXXX==</KeyID>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXb</KeyIDGUID>
    <KeyIDHEX>5XXXXXX36d85XXXXXXXXXXXXb24XXb</KeyIDHEX>
    <PSSH>EhXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6lebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3jXXX==</PSSH>
    <ServerURL>https://widevine-dash.ezdrm.com/proxy?pX=XXXXX</ServerURL>
    <ServerGet>request={"policy": "", "tracks": [{"type": "SD"}], "content_id": "6lxXXx0Z8xXXXXXXXXXlbg=="}</ServerGet>
  et>
  <ResponseRaw>
{"status":"OK","drm":[{"type":"WIDEVINE","system_id":"edef8ba979d64acea3c827dcd51d21ed"},"tracks":[{"type":"SD","key_id":" WVXXXXXXXXliBEXXw+XXXXX==","key":" W5XXXXXXXXZHxTjhXXXXvw==","pssh":{"drm_type":"WIDEVINE","data":{"EhXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6lebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3jXXX=="}}]}]}</ResponseRaw>
  </WideVine>
  <PlayReady diffgr:id="PlayReady1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <Key>W5XXXXXXXXZHxTjhXXXXvw==</Key>
    <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXX56bf</KeyHEX>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXb</KeyIDGUID>
    <LAURL>https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXX</LAURL>
    <Checksum>1Xq+XXXXX0=</Checksum>
  </PlayReady>
</EZDRM>
```

Setup Instructions

Please follow the steps below, on the machine where the kaltura-nginx package is installed.

Note: *If you've chosen the all-in-one server installation then the nginx should be installed on the same machine as the rest of the Kaltura system. If you've chosen to install a Kaltura cluster, then please go to the machine where the kaltura-nginx package is installed.*

1. Make sure you have root access (sudo)
2. Make sure you have obtained a ContentID
3. Open your nginx configuration file and add the following configuration, replacing the `u=username`, `p=password` and `c=contentID` highlighted below with your **EZDRM username, password and content ID** and the **hostname** with your files storage backend hostname
4. The following values are **required** for optimization of the DRM licensing process:
 - a. `vod_drm_info_cache drm_cache 128m;`
 - b. `vod_drm_single_key on;`

```
vod_drm_info_cache drm_cache 128m;
location ~ ^/drmdash/ {
    vod dash;

    vod_mode remote;
    vod_upstream_location /vod_upstream;

    vod_segment_duration 4000;
    vod_bootstrap_segment_durations 3500;
    vod_align_segments_to_key_frames on;
    vod_dash_manifest_format segmenttemplate;

    vod_drm_upstream_location /drmdashinfo;
    vod_drm_enabled on;
    vod_drm_single_key on;
    vod_drm_request_uri "u=username&p=password&c=6lxXx0Z8xXXXXXXXXXXLbg==";
```

```
add_header Last-Modified "Sun, 19 Nov 2000 08:52:00 GMT";
add_header Access-Control-Allow-Headers "origin,range,accept-encoding,referrer";
add_header Access-Control-Expose-Headers "Server,range,Content-Length,Content-Range";
add_header Access-Control-Allow-Methods "GET, HEAD, OPTIONS";
add_header Access-Control-Allow-Origin "*";
expires 100d;
}

location /vod_upstream {
    internal;
    rewrite /drmdash/(.*)/$1 break;
    proxy_pass http://hostname;
}

location /drmdashinfo {
    internal;
    proxy_pass "http://cpix.ezdrm.com/kaltura.aspx?";
}
```

5. Restart NGINX

```
sudo service nginx restart
```

6. To test your configuration, you will need to load a mpeg/dash manifest in a DRM compatible player such as Bitmovin or Shaka player. An example manifest URL in mpeg/dash format is:

https://<your_nginx_server_hostname_url>/drmdash/path/to/video/file.mp4/manifest.mpd

Note: make sure to replace the highlighted parts with the relevant values.

Apple Fairplay DRM for HLS

Obtaining an Asset ID

Option 1: Request DRM keys using EZDRM URL for Kaltura

To obtain an AssetID, use the following URL replacing **u=** with your EZDRM **username** and **p=** with your EZDRM **password**.

<http://cpix.ezdrm.com/kalturagetasset.aspx?u=username&p=password>



Option 2: Request DRM keys with curl

Using EZDRM's web service, the curl script below retrieves the DRM values from the web service.

```
curl -X POST 'http://cpix.ezdrm.com/kalturagetasset.aspx?U=Username&P=Password' -d ''
```

Note: there is a space between the single quotes at the end of the line.

The parameters are as follows:

Parameter	Description
U	EZDRM username
P	EZDRM password

The following is returned:

```
[{"asset": "9c1XXX4-6d6e-4773-bXX2-c86XXXXX5c1"}]
```

Setup Instructions

Please follow the steps below, on the machine where the kaltura-nginx package is installed.

Note: *If you've chosen the all-in-one server installation then the nginx should be installed on the same machine as the rest of the Kaltura system. If you've chosen to install a Kaltura cluster, then please go to the machine where the kaltura-nginx package is installed.*

1. Make sure you have root access (sudo)
2. Make sure you have obtained an Asset ID
3. Open the nginx configuration with your favorite editor
4. Copy and paste the following lines towards the end of the file, replacing the {username} and {password} highlighted below with your EZDRM username and password, and replace the \$hls_drm_asset_id value with the correct guid for your Asset ID obtained in step 1:

```
vod_drm_info_cache drm_cache 128m;
location ~ ^/drmhls/ {
    vod_hls;

    vod_mode remote;
    vod_upstream_location /vod_upstream;

    vod_bootstrap_segment_durations 2000;
    vod_bootstrap_segment_durations 2000;
    vod_bootstrap_segment_durations 2000;
    vod_bootstrap_segment_durations 4000;

    set $hls_drm_asset_id "9clXXXX4-6d6e-4773-bXX2-c86XXXXXX5cl";

    vod_hls_encryption_method sample-aes;
    vod_hls_encryption_key_uri "skd://fps.ezdrm.com/;$hls_drm_asset_id";
    vod_hls_encryption_key_format "com.apple.streamingkeydelivery";
    vod_hls_encryption_key_format_versions "1";

    vod_drm_enabled on;
    vod_drm_single_key on;
    vod_drm_upstream_location /hlsdrminfo;
```

```
vod_drm_request_uri "a=$hls_drm_asset_id&u=username&p=password";
add_header Last-Modified "Sun, 19 Nov 2000 08:52:00 GMT";
add_header Access-Control-Allow-Headers "*";
add_header Access-Control-Expose-Headers "Server,range,Content-Length,Content-Range";
add_header Access-Control-Allow-Methods "GET, HEAD, OPTIONS";
add_header Access-Control-Allow-Origin "*";
expires 100d;

}

location /vod_upstream {
    internal;
    rewrite /drmhls/(.*)/$1 break;
    proxy_pass http://hostname;
}

location /hlsdrminfo {
    internal;
    proxy_pass https://cpix.ezdrm.com/kalturafps.aspx?;
}
}
```

5. Restart NGINX

```
sudo service nginx restart
```

- To test your configuration, you will need to load an HLS playlist in a Fairplay player that can be provided by EZDRM. An example HLS playlist URL is:
https://<your_nginx_server_hostname>/drmhls/path/to/video/file.mp4/index.m3u8

Note: make sure to replace the highlighted parts with the relevant values.

References

NGINX VOD Module Github Page: <https://github.com/kaltura/nginx-vod-module>

Additional Information

For additional questions and comments please contact: sales@ezdrm.com