IBM Cloud Video

Presented By

K.Anu

Developer Tools

Control the viewing experience for your audience. Create your own streaming applications, customize interfaces, analyze engagement and access telemetry data with the APIs and SDKs of the IBM Video Streaming platform.

Viewer Authentication API

Connect IBM Video Streaming to your own user database or directory information service and manage access control using your existing security Tools for both live and recorded content.

Introduction

Viewer Authentication API lets you implement a custom authentication process. This prevents live and recorded videos on your account from being accessed until the viewer authenticates. The authentication process is based on JSON Web Token (JWT) standards (JWE and JWS).

Flow

To implement viewer authentication flow you need to create an application that includes the following:

- Screens your viewers need to go through to authenticate
- Backend code that validates these authentication requests
- The process to redirect back to / or embed IBM Video Streaming components (player, QnA, Chat) with authentication token

Embed with Token

If you have protected pages where you want to use our embedded components, you can pass the authentication token directly to the embed URL in the vt query parameter.

Redirect Flow

If you embed the player to an unprotected page or use IBM's channel page, the player will redirect the viewer to your authentication endpoint. You need to redirect back to the ref URL (passed in query parameter) with authentication token in the vt query parameter.
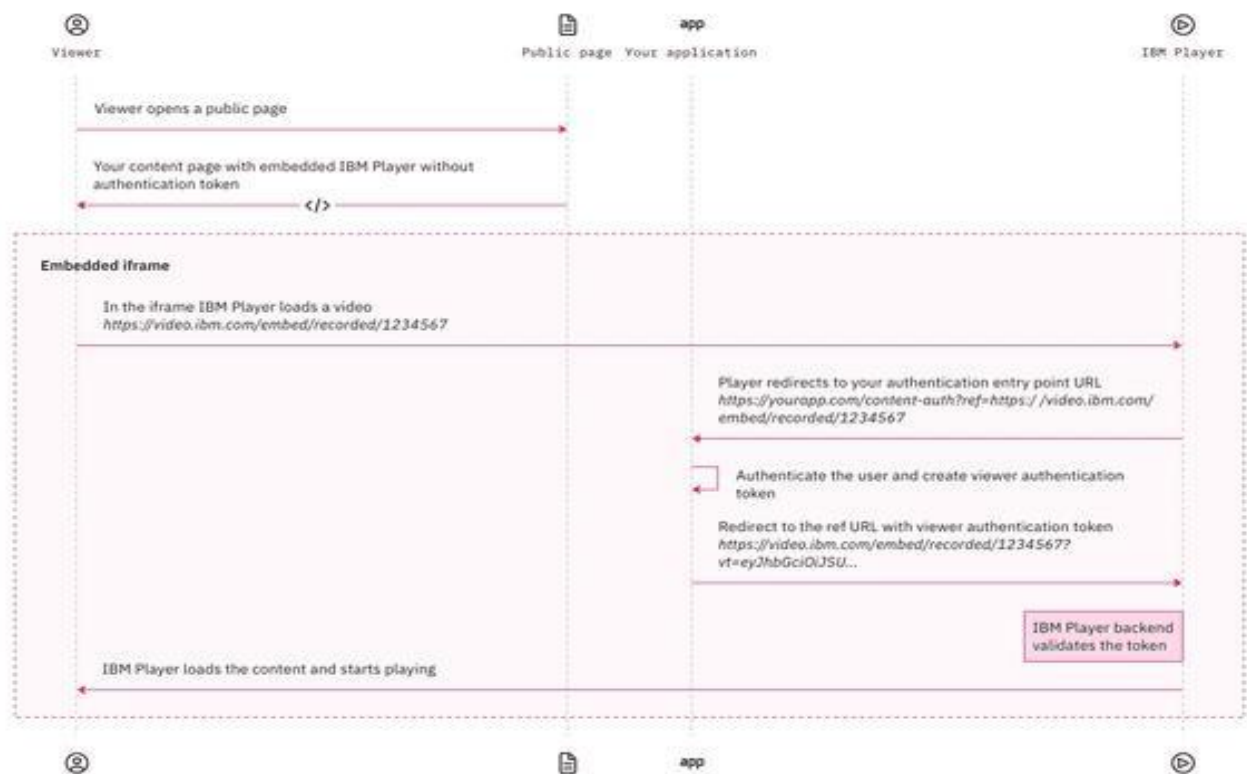
Viewer Authentication Settings

You can enable viewer authentication via API.

- Generate an RSA key pair with 2048 bits key size
- Set the public key and authentication URL to the content (channel or video) via API.

We recommend using different key pairs for every content.
By default, every video inherits these settings from the channel but if you want to use different authentication for a video you can set it.

Example to key generation:

```
Ssh-keygen -t rsa -b 2048 -E SHA512 -m PEM -P "" -f content_1234567.key

Openssl rsa -in content_1234567.key -pubout -outform PEM -out content_1234567.key.pub
```

Viewer Authentication Token

Viewer Authentication Token token is an encrypted JWT (JWE) that contains a signed JWT (JWS) with the following mandatory claims:

Sub: Subject, the token subject (identifier of the viewer, we recommend e-mail address)
Iat: Issued at, Unix timestamp in seconds when the token was generated. We will accept the token in the next 60 seconds from this timestamp. (We don't accept tokens with future issued at timestamps)
The JWS is signed with your private key that you set to the content. The JWE is encrypted with Video Streaming's public key that you can get from the API.

Viewer Authentication Token Lifecycle

You can use the same token for the different components (Player, Chat, QnA)
You can use every token once for a component: when the component embeds reloaded (eg: viewer reload the page) you need to generate a new token.
The token lifetime is 60 seconds after issued

Token generation method

Create JSON Web Token Payload with manadatory claims

```
{
  "sub": "viewer@yourdomain.com",
  "iat": 1624363990 //current timestamp
}
```

Sign JWT with your content private key using *RS512* algorithm

The key you've generated when you set the viewer auth to the content

```
eyJhbGciOiJSUzUxMiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJwbGF5ZXIiLCJzdWIiOiJ2aWV3ZXJAeW91cmRvbWFp
bi5jb20iLCJpYXQiOiJE2MjQzNjM5OTAsImV4cCI6MTYyNDQ0MzIyNX0.0krgCWmCykz7mXipXyb0GCa5hwEiXjW4M
i218M7Eo1VUe5oyMVbEoWUETBcJZiUQ_e7gfh16OxsCsTAzz3rkC8GmuLxjjKjfC_hDLitg3PMpTCYqZJE_oFKiJi
b_vReH8rOpIKAeaB0ZpLvI3ddTQyPKBzPHOCTdWlrVFyVxthCwbMwRBWKmx18MUZ5VMHW3IehstryhksXab0O3oT9
Gr3dQt_X3rjMYbxDYnT4uq9ZaOaaxhTvoKcnwO8FgmFWqD6RTctBBeoz5LUjvRoK7B9C5IPpH68GG5Jr96qlDWY2b
gxQNmVyAaZZh6ymV6QEvukEcnOGviZ_RI2aN7KltXhWFfV6v1odEHAQneiqDqKe3NuSdPFHJxvDTm_LMzTnVr7Vf4
SpPevZ8qmhG6aPyGBGebqIZLZvzLQS8tDyjrHPBNq3hSvplkPOhm9RKQijEH0IYRqWyxr9MmlurEWYROud2tlk2kv
FSAVGzDcig4e3MRtComFHE6m980YJ4lHEAWVn5qvkOXEs7tBDkdFS1eiVpJjK2wfnoW5drJhNCIp0RCw4kmGVDgpW
28sUdGLRinVVAYL2_eiwDCTWOvexv8_ZA8RVPGji3Nf09EwHcMWi1X5113rynaYqTSmwEarPCP5J1oCuOsxkKScLY
yv73yL8aMU7JK6dvSZg
```

Get Watson Media Viewer auth public key from API

https://api.video.ibm.com/viewer-auth-public-key.json

The respose is in JWK format.
You can cache this response on your side (and we recommend to do so) but from time to time we will change this key. After the key change we will accept the deprecated key for a time.

```
{
    "keys": [
        {
            "kid": "a4a2c9240f22947f326278bb3f70c307b14b7ffca0c0cddcc5e89c63b26b08b6",
            "kty": "RSA",
            "use": "enc",
            "n": "42EVWz1rgkavbpOIAyGv8_AC0jDDJF10KqPzwIExTvyscOIZTNGe9M-GdsfJqlybfO
                  6eAaHd-fRkVQJJO2m5WbIJBSPfayDfEhFNp0dS_AiYd70u0yve51g6im5a1jFzW6miUgwtorGvS
                  LUyAogdmygydguUncitxx42VaWanXpeZ6vuUbq6Ztj85Vp-HIaHuSnxSsNrTgPBm1DlUIiJrqTI
                  IqmBf3B9uLhAcRYXm2cXkJQl-T6VkoRDHKNtdl_AWmVi61Bh8lp975dMsujpKwVzyOba2-XfSsg
                  OI9NzoutEOBJGVECZLhkIZ-TRXIYqgFp15iEHz4e2SM9a19r5TdM6zYdJmCOcWER2RPDlR6dvnc
                  Ay2dStP9pAC4S6mqKsityath26YSO3QxU2AvGlUYEMdStGwTHWMJSKNXqmlfSRk_qkcwLeqkgI
                  PtS6jm_9yq5k8qKgtHbujrcBtWAp0SQJIIb_7NnK1M4fytPg80wxhW8NzYnMKRL2VDaS6CQj9Y
                  Jmbu02k1Sk_oS_AZxBl JFlpNQYvV FQfWb0HOsZsR0qbee2fEFiYvDqXUMR FxGjOwLoxLTfS
                  6eHpQaRAGvatql6BRoDHwZ6zczLERH3fEcfgdabOLJyJAZurLD901oH3qpwvbwESpcF6zLJxqP
                  R7T7amLGgaz6lIYp3rWQ0",
            "e": "AQAB"
        }
    ]
}
```

Set JWS to the JWT payload

Set JWT's *kid* header from Watson Media viewer auth public key response

Encrypt JWT with the public key from JWK response with *RSA-OAEP* sign algorithm and *A256CBC-HS512* encryption.

```
eyJhbGciOiJSU0EtT0FFUCIsImVuYyI6IkEyNTZDQkMtSFM1MTIiLCJ0eXAiOiJKV1QiLCJraWQiOiIxOGEwMDgyY
WUwODU5N2Y5ZWU4OWRkZTYzMWExZjA2ZmQ3YWUyZTFmYWY0YjJkZTFjMTQwNDhlYmI1NTEzNjZiIn0.B7XEDM9pJ2
nFAzIUQDXyi9j4O4AoRnxX8trlkP6bGaddJHl3JKVlLkClWs3cqqDewtETs3yT-7oJhYIj9Cgt0g2hJrcAKkgNBUw
MBgHcCQ6kgfyq1lApB6AEM4jI0nTiNyyLRzujp5zIrz4lIhQyBxVhVY0yoKPAjvcEzUJYIfPTWYkTSBeLZ0NNGAGK
nIqIPKIluegGBfh8I3NzSHNsbTpxS8VcGpnXAg5xdGRwGe5kpKDGlSh5LGaadd414yscfNrfu5ayKgtMnipKOLbtl
CDRZ_vWQf13dI_MEkR9kMjT4iu3n0AslJtBPzMtTbigYh3a7MdLFk50Xe6VtH3nuzzoyw3D2vC73inH7w7QBx_2r6
BYtvkV1anLXJ0cJCfQNlkNu92yUlHUjNjDcqSY9KU4xbIis8s_h40UCLKv65KkZ9qBxD2STYrNFbIJCDE6mFxcLmc
qpMROLi10KQAh8h8iRMZ8SfzmCx0_aN99tu7VVnIaHCDSxmP7NoVIn1U8p3Be4d4wc7Zt69IDDk9RVmSZum4hdOQz
dmyHDIAlAYmQljUu4lxaVJs0WU_wJHEWMAfNKragXKPzM8IBZ301MTGt9jgnYQKaeB_1qL90tzShfToNMs8fB69oi
BRPBkpSsp78hku1p5GFDQQo0zrs5vfCtrAt_aZ1ZI.Q4_Jrl5JpI31hMbVjm6HFQ.GqxE59rS7UPLwYQCxeMyePuK
VYJlSPDlj6Cd6xHTY_S0nN7GqR2uZB736hwD6me4zUGh25_MleYzPi26lln_2yztgk3WudaEBnn8UwKJmrOygRmoa
MgURTLufHCXjTbpjvfRHJW0rjosSUphYJNEK9CIQZlhlDx0uZYnId61MoJFHuEhrk1Y5l9L6uABhC_TX989x1dggJ
ceOo4F8MvlgBWSuw601lytlM509xBjHM5snIaduHpl0M4bkaZAi2m4QaqNG7CkZ3qlVkuTj18Tn4z7XFWjC0uJ7IC
bLKPAuiNDnJMDcvFhqBvqdtEOBJGMmi96pAv_z54pd57O7eaSu_rxMZ5Me4CH_jHVLpXpFrnXcSvrP86kmNXO2xrC6LL
SSXJQFcSFVA2SUFnNSWBhoCSf5bzc7p70WiEcqKpCf1BxTDtrYM1AVXbd_jqzy1DOtUH4BSxX1zz_gqaqlMWJTU5m
wSXK052HJiRsHIOXP2B_pdpfN3p6B0m2vDwx0uQuEp8iBe3wXuMlqfi85yJvYHQLrXNJqvgXjLMmS_aJxsR57JlVI
KVTs4iPa_ELFKAy62S0ScBsl_xoG4eo2_8hGYDkVNSg5NvwPRezfWkzrqLK63TS9UGMnk
```

Use the JWE result as viewer authentication token