3 HOW TO TRUST ▷ YOUR PLAYER

Presented by BITMOVIN Friend MTS intertrust

# BUILDING AN OTT SERVICE FOR TODAY'S WORLD

## **Article 3** – Tips and Tricks: How to Secure Your Content in Challenging Streaming Environments

Published Date: October 6, 2020

*Joshua Shulman, Digital Marketing Specialist, Bitmovin*

Piracy occurs at all levels of video streaming, from illegal downloads to screen captures. How can an OTT provider overcome these issues? Fortunately, there's a good answer: with a mixed balance of back-end solutions including digital rights management (DRM), watermarking, and/or client-hardening. As a part of a multi-post series between partners Bitmovin, FriendMTS, and Intertrust Technologies, Bitmovin is here to define some of the top tips and tricks to implementing these solutions into your web-based player.

By the time content arrives at a web-based player, a majority of protection measures should already be in place. Although it's possible to arrive at the player without a concrete DRM, watermarking, and/or client-hardening solution, this is ill-advised, as not all consumer players can be trustworthy enough to simply view content without engaging in some kind of piracy measures.

### How to Secure Video Streaming Content in Web-Based Environments

The browser environment, open by default, is a challenging environment to secure. Delivering high-value premium content to a web browser can be a risky venture, but one that is critical to reach your audience. To reach a maximum audience, the recommendation is to implement a player in as many devices as possible, including app-first or native solutions. Browser environments are amongst the farthest reaching, but least secure, due to their open nature, and will require some extra attention when implementing content protection systems.

Content licensors (or content owners) are increasingly wary of the impact of content theft at user playback, and will often mandate use of certain obfuscation techniques as part of authentication and authorization flows. **As the second article in our "How To Trust Your Player" series highlights**, ensuring that session authorization tokens are securely ciphered, and can prevent attacks against DRM license acquisition servers is critical to developing a truly end-to-end security chain.

For the browser playback environment where website code (JavaScript) is interpreted and executed, masking how to interact with security systems in place is a critical step. This typically takes place through the use of a code obfuscation tool. The goal of this type of tool is to render the source code unintelligible to prying eyes without fundamentally altering how it functions.

Obfuscation entails parsing JavaScript (JS) source code, rearranging the code, and at some points, transforming it by renaming variables and data structures, and refactoring logic structures to mask algorithms. This makes it nearly impossible to understand the code and how data is parsed by it. The result is code that is extremely difficult to read and reverse-engineer, either by a tinker or a more determined actor…such as a content pirate.

## How to Bolster Your Video Streaming Defenses

Techniques such as uglify-ing or minify-ing JS code provide some minimal defenses, but can be reverse-engineered themselves through automated tooling. While it may not be possible to get back to the original source, it is possible to generate much more intelligible code from tools such as a JS beautifier, from which a hacker could discern information beneficial to attacking your code or services.

### Improving on Obfuscation

JavaScript protection solutions, such as **Jscrambler**, provide significant robustness by generating code with polymorphic obfuscation techniques. On top of this obfuscation, code locks are added to restrict the browsers and platforms on which the code can be executed, providing the ability to restrict the code use to a specific user session. They also aid in the generation of self-defending code, where anti-tampering techniques protect functions and objects. These anti-tampering techniques can trigger defenses (such as halting execution and throwing fatal errors), or generate session invalidation events that trigger a service block for future HTTP requests to your security services.

As your code has to execute on a web browser, following open JS standards, it just is not possible to completely secure playback. Obfuscation products are not a foolproof mechanism to create a secure execution environment. Someone with enough motivation, and time to spend gathering intelligence and doing research, will eventually be able to reverse engineer your playback code. However, putting in place multiple layers of JavaScript code obfuscation as part of a complete defense strategy will deter attacks from content pirates.

### Concurrency Management

Many content owners require OTT service providers to limit account oversharing – the number of simultaneous video views that can take place from a single authenticated and authorised user account. While this is primarily to ensure that a household's stream concurrency or device limits are not exceeded, this has the effect of limiting the impact of credential sharing outside of the user's household. Concurrency management typically takes place by keeping a tally of the number of play/pause/stop events that the player framework's analytics data generates.

Below is a standard tally-event measurement system that measures users "Alice" and "Bob" based on overlapping timestamps of video views in similar geographic locations. ("P" indicates a video pause.) Although this helps monitor general concurrent usage across shared accounts, this method has its limitations.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Alice | | | 1 | 1 | 1 | 1 | P | P | P | 1 | 1 | 1 | 1 | 1 | |
| Bob | | | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | | | |
| Concurrent | 0 | 0 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 0 |

*Concurrent management tally sample*

One is that this method often is not robust enough to limit concurrency. This is because the analytics events can be intercepted and blocked – and are not explicitly tied to a service's DRM license issuance, and the user's entitlement store or rights locker. A better practice is to include heartbeat messaging, driven from the player's message bus with the playhead timeline position (or an offset for VoD), that ties to a specific user's session. When a stream entitlement check takes place as part of DRM license issuance, a heartbeat identifier should be set, tied to the user's session, cryptographically signed, and then passed to the player as the heartbeat token.

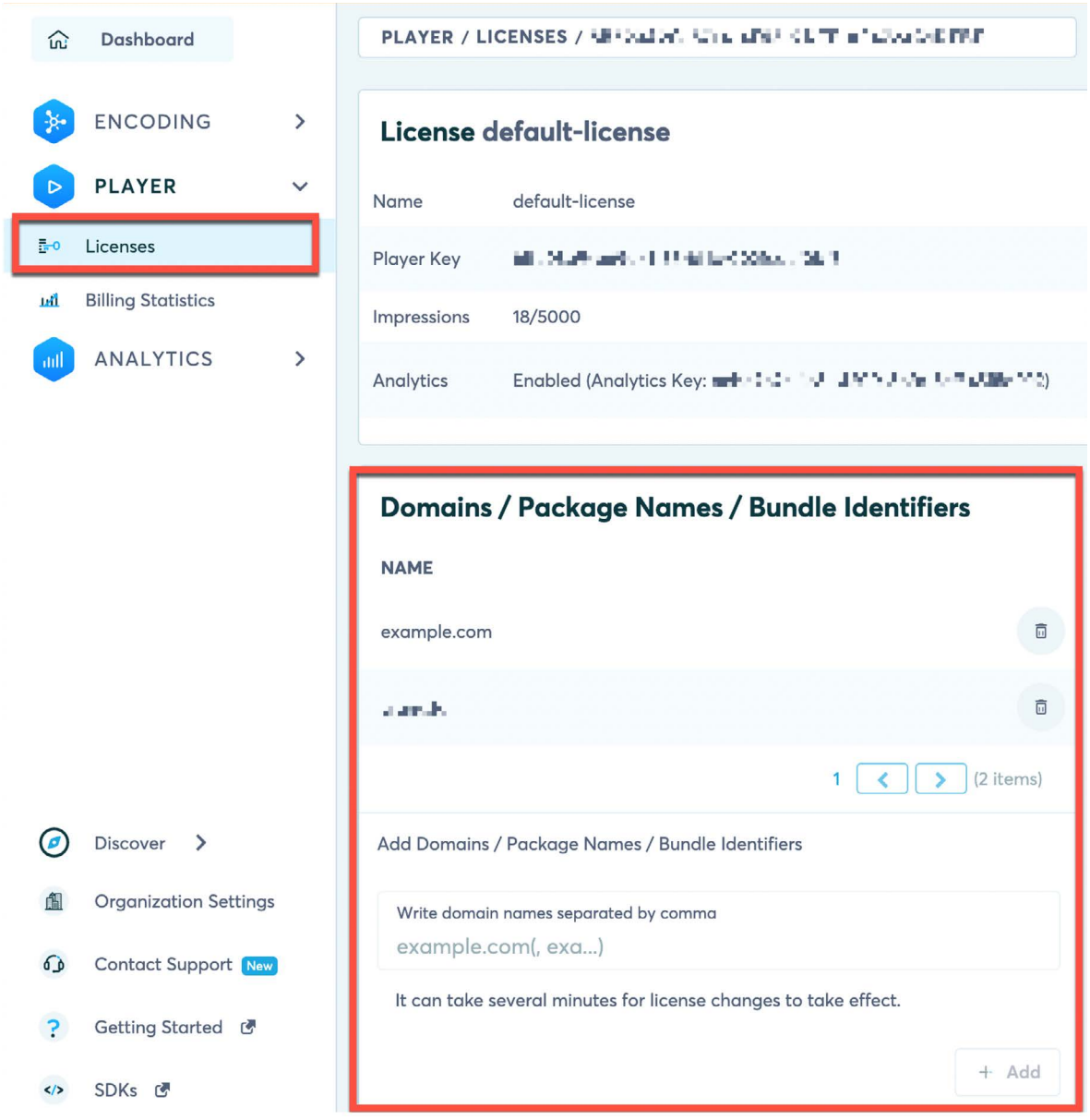
On playback start, and until the end of the session, the player should communicate with the heartbeat service at a predetermined interval to exchange the heartbeat token. At exchange of a valid heartbeat token, the heartbeat service would respond with the next/refresh token, and the user's stream entitlement within the user store would be tallied. If the heartbeat token is not validated within the predetermined interval (+/- n seconds), then the heartbeat service would remove the user's entitlement to play back the content. This, in effect, would remove the ability for the user's session to obtain further DRM licenses until the session had been reset.

When receiving an error from the heartbeat service, the player (through the heartbeat customisation) should invoke the player "stop" functions to tear down the session.

### Domain Locking

Whereas concurrency management is a method of monitoring how many users are viewing the same account, domain locking is essentially a technique to white or blacklist certain websites. It will prevent a player from being embedded on a non-approved site, such as one on which an aggregator might want it to look like they have content available – but in reality are embedding another service's player.

Bitmovin's web-based player, as part of the standard security controls, uses an allowlist for player licensing to prevent misuse. The top-level domain name or host for which the player can be used must be added within the "Player – Licenses" section of the dashboard by selecting "+ Domain" before deploying a player.




*Bitmovin player dashboard*

From this page, it is also possible to add IP address ranges to indicate where the player can be licensed, which can be useful during testing. Localhost is allowed by default. For the mobile/device software development kits (SDKs), the allowlist may also contain the package name and/or bundle ID. In the case of the Roku device, the dev.roku domain is mandatory, along with the Roku channel ID.

Once you've secured your **distribution chain from source to the playback environment**, and have followed best practices to secure the playback experience as much as possible (as above), it's imperative that you follow these rules to boost your users' experience – and ultimately, your brand.

## Rules to Gaining and Retaining Trustworthy Video Players

### 1. Make your content available where your users want to watch it

Combining Bitmovin's encoding and packaging solution to prepare the content for delivery, the robust ExpressPlay DRM system provided by Intertrust to protect delivery, and Bitmovin's Player, it is possible to support a wide range of browser versions and devices to reach your audience.

Bitmovin's multiplayer SDKs streamline the development by bringing your apps to all of the platforms your users would be willing to pay to watch it on – e.g., Smart TV, tablet or mobile device (iOS, Android, etc.). You can find information on the Bitmovin SDK and how to implement it in its **documentation**.

You can also view **all devices and apps supported by the web player**.

**(Web SDK) DRM Support on Desktop Devices**

| Browser | Minimum OS Version | DASH ClearKey | DASH Widevine | DASH/SMOOTH/HLS PlayReady | HLS AES128 | HLS Widevine | HLS FairPlay |
|---|---|---|---|---|---|---|---|
| Chrome (last 3 major versions) | OS versions supported by Chrome | ✅ | ✅ | - | ✅ | ✅ | - |
| Firefox (last 3 major versions) | OS versions supported by Firefox | ✅ | ✅ | - | ✅ | ✅ | - |
| Opera (last 3 major versions) | OS versions supported by Opera | ✅ | ✅ | - | ✅ | ✅ | - |
| Safari 10+ | macOS Sierra | ✖ | - | - | ✅ | - | ✅ |
| MS Edge (last 3 major versions) | Windows 10 | ✅ | - | ✅ | ✅ | - | - |
| MS Edge (Chromium) (last 3 major versions) | Windows 7 / macOS 10.12 | ✅ | ✅ | - | ✅ | ✅ | - |
| MS Edge (Chromium) (last 3 major versions) | Windows 8.1 | ✅ | ✅ | ✅ | ✅ | ✅ | - |
| Internet Explorer 11 | Windows 8.1 | ✅ | - | ✅ | ✅ | - | - |
| Internet Explorer 11 | Windows 7 (Adobe Flash required) | ✅ | - | - | ✅ | - | - |

*DRM Systems supported by the Bitmovin Web Player*

**2. Feature parity with piracy**

Create an impactful and feature-rich player that improves the viewer's quality of experience. Don't punish legit users by restricting how they view their content, such as with **offline play**, time to release and overall quality. In some cases, legitimate content just is not available in high enough resolution, whereas pirated content might offer 4K quality.

**3. Provide your content at a reasonable price point**

Bitmovin's player SDK enables an OTT provider to spend less time developing workflows for each potential player implementation by reducing workflow cost with easy-to-use configurations.

## Summary: How to Secure Video Streaming

The combination of these three rules creates a more favorable user experience than what content pirates can provide. Yet, there is one last problem to overcome once your player is ready: re-streamed content. This is where an effective watermarking service comes in. Not only will it detect, deter and disable leaks, it will work to create a frustrating experience for illegitimate viewers and encourage them to use more legitimate means of consuming content.

***Make it harder to pirate content, but easier to pay for content***

---

To learn more about "How to Trust Your Player," check out the other articles in our series:

- **Article 1 – Tips from the Top: Secure Content Delivery and Playback**
- **Article 2 – Securing Content Access with Digital Rights Management Best Practices**
- **Article 3 – Tips and Tricks: How to Secure Your Content in Challenging Streaming Environments**
- **Article 4 – Beyond Digital Rights Management: Video Watermarking Weighs In**
- **Article 5 – From One End to the Other: Protecting Content From Origination to Playback, Once and for All**

Still want to learn more? View our associated Fireside Chat sessions:

- **Video 1 – Tips from the Top: Secure Content Delivery and Playback**
- **Video 2 – Securing Content Access with Digital Rights Management Best Practices**
- **Video 3 – Tips and Tricks: How to Secure Your Content in Challenging Streaming Environments**
- **Video 4 – Beyond Digital Rights Management: Video Watermarking Weighs In**

Check out the recording of our How To Trust Your Player Webinar: **View Recording**.

For information on redistributing this content, please reach out to **pr@friendmts.com**.

**How To Trust Your Player** is a collaborative effort between Bitmovin, Friend MTS and Intertrust. Our goal is to educate media and content providers on the importance of delivering streaming content in the most secure ways possible from the video player to the end-consumer while protecting both their content and revenue.

## Bitmovin

Bitmovin is a developer of video streaming technology. Built for technical professionals in the OTT video market, the company's software solutions work to provide the best viewer experience imaginable by optimizing customer operations and reducing time to market.

Bitmovin's solution suite – a video encoder, player, and analytics platform – lets content owners redefine the viewer experience through API-based workflow optimization, fast content turnaround, and scalability.

Founded in 2012, the company is based in San Francisco, with offices in major cities in Europe, North America and South America. With more than 250 enterprise customers around the globe, Bitmovin helps power clients like BBC, fuboTV, Hulu Japan, RTL, and iFlix.

## Friend MTS

Friend MTS helps media and entertainment businesses secure content so that revenue can grow and creativity can thrive.

With advanced services that measure, monitor, detect and disable content piracy, Friend MTS provides a 360-degree
view of the constantly shifting content piracy protection ecosystem and stays a step ahead of ever-advancing and sophisticated content piracy behavior and technology with a sharp, deliberate, laser-focused commitment to continual monitoring and innovation.

Businesses and nonprofit organizations throughout the world recognize Friend MTS as the leading authority for content and revenue protection. The company also has donated its digital fingerprint technology to the International Center for Missing and Exploited Children to tackle child abuse content online.

Founded in 2000, Friend MTS is headquartered in Birmingham, England, with operations throughout Europe, the Middle East, Africa, Latin America, and North America. Friend MTS is the recipient of an Emmy® Award for Technology and Engineering, presented by the National Academy of Television Arts and Sciences (2018).

## Intertrust Technologies

Intertrust provides the world's leading digital rights management (DRM) cloud service with a complete ecosystem of security and rights management products.  We empower businesses to securely manage all of their data and devices, regardless of location, format, or type–enabling innovative multi-party apps and services.

Intertrust Media Solutions provides robust content protection solutions for Media and Entertainment. Intertrust ExpressPlay consists of a cloud-based multi-DRM service, broadcast TV security and anti-piracy services with proven scalability in the largest OTT streaming platforms globally.

ExpressPlay DRM™ is today's most complete multi-DRM monetization service for OTT streaming supporting Apple FairPlay Streaming, Google Widevine, Microsoft PlayReady, Adobe Primetime, and the open-standard Marlin DRM. Intertrust also offers ExpressPlay DRM Offline to enable secure streaming of premium content through an offline multi-DRM platform.

Founded in 1990, Intertrust is headquartered in Sunnyvale, California, with regional offices in London, Tokyo, Mumbai, Bangalore, Beijing, Seoul, Riga, and Tallinn.