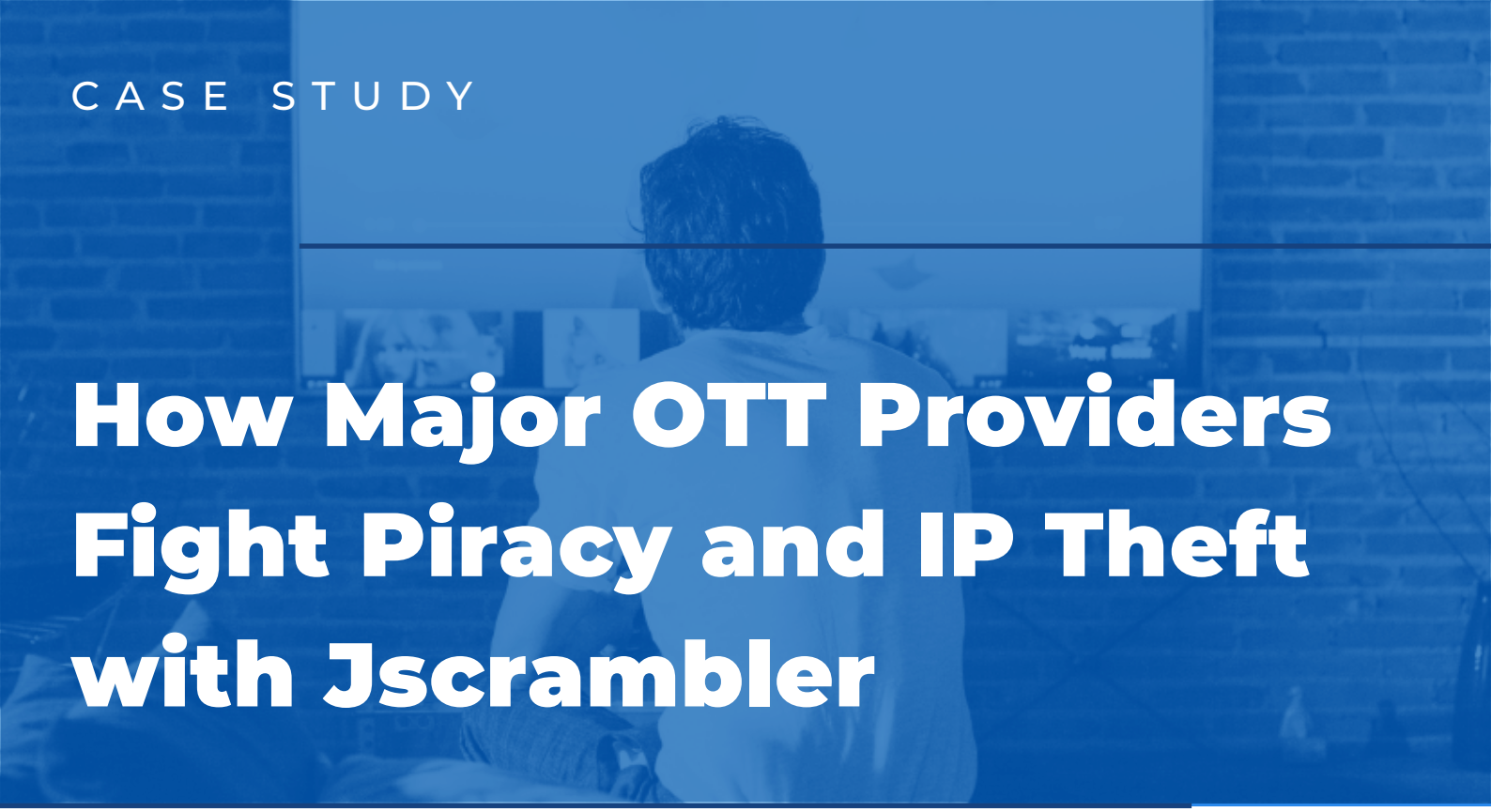


CASE STUDY

How Major OTT Providers Fight Piracy and IP Theft with Jscrambler

BEHIND THE STREAM

"We can't afford to lose months of R&D by leaving it in easy reach. We had to get the best possible code protection."

jscrambler

OVERVIEW

jscrambler

OTT
THE NEW STANDARD OF ENTERTAINMENT

For decades, cable was king when it came to entertainment. But with the Web explosion in the '90s, it became possible to deliver content over the top. As broadband Internet quickly spread during the 2010s and users wanted to access content from their mobile devices, OTT quickly outpaced cable.

Today, there are over **600 million** OTT subscribers globally. This industry generates over **$100 billion** in annual revenue, a figure that is expected to top **$332 billion** by 2025.

In this fast-paced industry, providers face two key challenges. On one hand, to retain customers, they develop proprietary solutions that must be kept safe from the prying eyes of competitors; on the other, piracy is also growing, putting providers at risk of noncompliance with content rights owners and costing them a good chunk in lost revenue.

THE CRITICAL VALUE OF INTELLECTUAL PROPERTY

Customer behavior analyses show that one of the main reasons why users cancel a certain provider is the experience of using the player. At a time when competition in OTT is at an all-time high, **customer retention** is the order of the day.

This prompted providers to invest in proprietary solutions to enhance buffering, analytics, or even the experience of using the player.

Because modern OTT platforms are developed in **JavaScript and HTML5** — which provides countless benefits for the provider — the **logic of these proprietary solutions is left exposed** on the client-side. Meaning that a competitor can readily debug the code to uncover these novel developments. Providers are taking the necessary steps to protect this exposed JavaScript and HTML5 and avoid losing the upper hand.

STOPPING PIRACY, FILLING THE SECURITY GAP

Losses from piracy are expected to grow from **$9.1 billion** (2019) to **$12.5 billion** (2024), urging providers to employ advanced solutions to minimize their exposure to content leaks.

DRM, a widely used anti-piracy layer, ensures that content can only be accessed by rightful users. However, it doesn't provide any further protection when the content reaches the user's display — meaning that malicious users can capture the content and redistribute it.

Forensic watermarking helps solve this issue by embedding subscriber info into content, enabling providers to block leaking accounts after leaked content is found in the wild. Modern watermarking solutions use a **client-side JavaScript agent**, a much superior approach when compared to server-side watermarking. However, this client-side agent **is exposed to attacks**. By tampering with the logic of the watermarking agent, attackers can **bypass the watermark** and leak content without it being traceable.

SUMMARY

CHALLENGES

Conceal the source code and prevent reverse-engineering to protect innovative algorithms

Reduce exposure to content leaks coming from unprotected client-side watermarking

Increase the overall security posture of OTT platforms

SOLUTION

Enterprise JavaScript Protection:

Polymorphic obfuscation to conceal client-side logic with unpredictable code changes

Self-Defending to mitigate debugging, reverse-engineering and tampering of the watermarking agent

Robust countermeasures to stop attackers in real-time

RESULTS

Seamless app build integration

Increased application security posture as outlined by OWASP

Zero performance issues

Zero leaks coming from tampering with the watermarking agent

CHALLENGES

As early as 2014, numerous players in the OTT industry have reached out to Jscrambler with demanding security challenges. Leading companies were adopting the new HTML5 standard and became aware of changes to their attack surface that could pose key business threats.

Since JavaScript and HTML5 are presented as cleartext in the browser, any end-user is able to retrieve or tamper with this logic. One of our first OTT clients was a top-5 global OTT provider who had the priority of ensuring that its innovative algorithms would be concealed and protected from code reverse-engineering attempts.

The increased competition in this industry has prompted several other providers to look for the best possible solution to keep client-side logic as protected as possible from competitors or even attackers. In many cases, source code protection was a standard security requirement.

"Competition is tough in our space. We can't afford to lose months of R&D by leaving it in easy reach. ***We had to get the best possible code protection."***

The close collaboration between the security engineers from these OTT providers and Jscrambler's own team of engineers led to a common understanding of another key threat — the risk of tampering with forensic watermarking solutions. The unprotected code of client-side watermarking agents could lead attackers to new ways of leaking premium content.

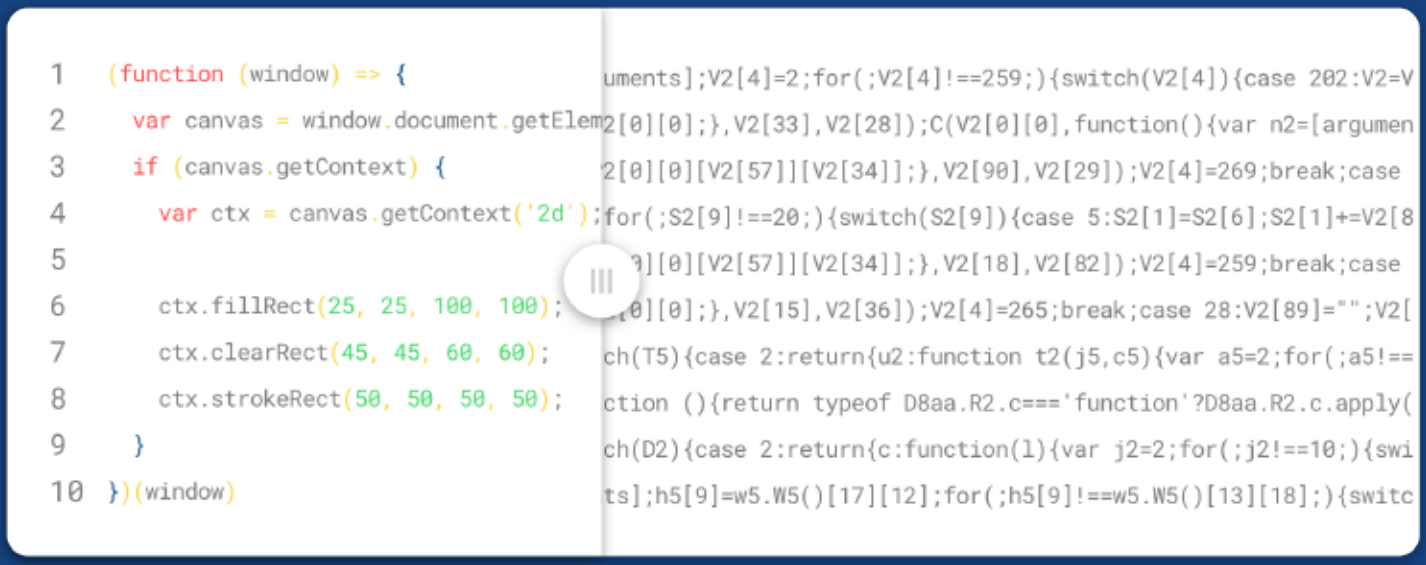
Natural progression led Jscrambler to some of the main providers of forensic watermarking, who were aware of the security implications of having a watermarking agent that was exposed on the client-side.

And because security concerns don't stop at piracy, the close collaboration between Jscrambler and OTT providers also made it clear that their Web OTT platforms had to be protected against the main Web security threats as outlined by leading security standards like those from OWASP and NIST. One of the biggest needs here was keeping user data secure — which highlighted the need for source code protection.

"The only watermarking implementation that makes sense today is client-side. **Adding code protection must be a standard step for maximum resilience."**

Answering all the identified security threats meant getting the best possible **JavaScript and HTML5 code protection**. The clear answer here was the mature and proven JavaScript Protection technology provided by Jscrambler.

At the basis of this technology lies Jscrambler's **polymorphic obfuscation**. With this protective layer, providers concealed their source code beyond possible recognition. By leveraging the Jscrambler **Code Hardening** feature, providers were also able to ensure that no tool was capable of reverse-engineering their protected code. And thanks to the inherent polymorphism, each new protected build yields a different output — an extra line of defense against reverse-engineering and attack automation attempts.



```
1  (function (window) => {
2    var canvas = window.document.getElem
3    if (canvas.getContext) {
4      var ctx = canvas.getContext('2d');
5
6      ctx.fillRect(25, 25, 100, 100);
7      ctx.clearRect(45, 45, 60, 60);
8      ctx.strokeRect(50, 50, 50, 50);
9    }
10 })(window)
```

```
uments];V2[4]=2;for(;V2[4]!==259;){switch(V2[4]){case 202:V2=V
2[0][0];},V2[33],V2[28]);C(V2[0][0],function(){var n2=[argumen
2[0][0][V2[57]][V2[34]];},V2[90],V2[29]);V2[4]=269;break;case
for(;S2[9]!==20;){switch(S2[9]){case 5:S2[1]=S2[6];S2[1]+=V2[8
9][0][V2[57]][V2[34]];},V2[18],V2[82]);V2[4]=259;break;case
[0][0];},V2[15],V2[36]);V2[4]=265;break;case 28:V2[89]="";V2[
ch(T5){case 2:return{u2:function t2(j5,c5){var a5=2;for(;a5!==
ction (){return typeof D8aa.R2.c==='function'?D8aa.R2.c.apply(
ch(D2){case 2:return{c:function(l){var j2=2;for(;j2!==10;){swi
ts];h5[9]=w5.W5()[17][12];for(;h5[9]!==w5.W5()[13][18];){switc
```

Original

Obfuscated

As per the recommendations of Jscrambler Engineers, these OTT providers also applied the **Self-Defending** layer to prevent more determined attackers from debugging or tampering with the code. Jscrambler does this by scattering integrity checks throughout the code that are triggered when attackers open a debugger or make any change to the source code. When that happens, Jscrambler issues countermeasures such as breaking the app, redirecting attackers, or issuing a custom callback.

This Self-Defending layer is crucial both for preventing intellectual property theft and attackers' attempts to bypass the watermarking agent.

> *"It has **the most extensive and powerful set of features** that we saw in any JS protection product."*

Because watermarking bypass poses significant threats to OTT businesses, **Jscrambler has partnered with some of the leading watermarking providers** to ensure that their client-side watermarks are secured against this threat.

Thanks to the maturity of Jscrambler's infrastructure and API clients, all providers successfully integrated Jscrambler into their build process, ensuring that Jscrambler seamlessly protects their code at each new build.

*"The **integration process went much smoother** than we had anticipated. Really appreciate all the help from your support team."*

RESULTS

In all instances, Jscrambler fully addressed the security challenges brought forward by these OTT providers. Their **intellectual property became much more protected** from the prying eyes of competitors; they greatly **minimized their exposure to content leaks** by protecting the client-side watermarking agent; they **maximized the overall security posture of their applications** by making it extremely hard for attackers to plan or automate data exfiltration or scraping attacks.

The whole process, from the testing stage to integration with their CI/CD pipeline, took an average of two and a half weeks and fewer than three meetings with Jscrambler's Engineers. In all cases, Jscrambler successfully passed these providers' QA tests.

Another key concern that several OTT providers had was performance. Source code transformation is usually perceived as bringing significant overhead. However, thanks to Jscrambler performance-focused features like **Code Annotations** and **Profiling**, these providers experienced near-zero performance loss and the protected code passed all requirements.

In instances where providers' security teams were involved in the process, their security requirements were also met. Jscrambler directly answered some key points as outlined in security recommendations such as the ones from OWASP and NIST. Specifically, the source code protection provided by Jscrambler serves as an **extra line of defense against data exfiltration attempts**, as it greatly increases the cost of attack planning and automation.

*"We require **every deployment** of our client-side watermarking to be secured with Jscrambler."*

Ultimately, the most demanding requirement across most providers was ensuring that no attacker could bypass their client-side watermarking agent and leak content. After integrating Jscrambler, all of these providers have consistently reported **zero successful attacks** to their client-side watermarking. By ensuring that every piece of content is watermarked, they are able to guarantee the best possible anti-piracy protection, which spans from DRM all the way down to resilient watermarking.

jscrambler

CONTACT US

If you want to know more about how Jscrambler can help you Secure your JavaScript and HTML5, don't hesitate to contact us

hello@jscrambler.com

+1 **650 999 0010**

jscrambler.com

Gartner

Jscrambler is the leader in Client-Side Application Security
Recognized in Gartner's **Market Guide for Online Fraud Detection**
and in Gartner's **Market Guide for In-App Protection**

Trusted by the Fortune 500 and 43000+ companies and individuals globally.