FingerprintJS Case Study

How one company stopped review fraud with FingerprintJS

The company was able to catch an additional 500 fake reviews per day with FingerprintJS Pro's highly accurate visitor identification API, even when the perpetrators attempted to conceal their identity using a VPN or clearing cookies.

Results

Caught 500+ fake reviews per day

Switching from FingerprintJS open source to Pro resulted in a massive increase in identification accuracy and put a stop to the vast majority of fake reviews.

Identified returning visitors using VPNs

Pro's VisitorID remained stable even as users attempted to conceal their identity, catching even the most sophisticated fake review services.

Flexible solution for preventing future fraud problems

The company's team plans to use FingerprintJS Pro to solve other fraud use cases in the future due to its flexibility and accuracy.

Customer Overview

FingerprintJS works with an EU-based company that provides a peer-to-peer marketplace for selling online services. The company's website is popular across Europe, with over 3M monthly unique visitors in Italy alone.

Sector: P2P Marketplace

Use Case: Review Fraud

Site Traffic: 5M+ Monthly Visits

FingerprintJS Features

- ☑ 99.5% Accurate Identification
- ☑ Browser Fingerprinting
- ☑ GDPR and CCPA Compliant
- ☑ Incognito Mode Detection
- ☑ Geolocation

FingerprintJS

The Problem

The company was struggling with fake reviews on their marketplace. Many users on their site would sort listings based on total number of reviews and average star rating in order to surface the highest quality services. Due to this, vendors were incentivized to collect as many reviews as possible, and many would try to inflate their ratings with fake reviews. These vendors would either pay for services that would submit multiple fake positive reviews, or would write multiple reviews themselves by creating multiple accounts. The company wanted to ensure that vendors with high ratings were authentically loved by the community, and not merely the most successful at circumventing review limits.

The company found that many fake reviews were being created from the same device, but the perpetrator would use a VPN or clear cookies to conceal their identity. The team decided that the best solution to their fake review problem was a service that could accurately identify visitors even if they changed their IP address or used other concealment methods.

Solution Overview

A developer at the company found out about FingerprintJS after seeing a post on Stack Overflow about identifying anonymous visitors for the purposes of fraud prevention.

The company initially used FingerprintJS' open source library for browser fingerprinting, then decided to trial FingerprintJS Pro for increased identification accuracy.

After investigation, the company found that the FingerprintJS Pro API was able to catch a significant number of fake reviews, improving the integrity of their review system and overall customer experience.

How the company caught fake reviews

The company was already checking whether an account was being created on an email or IP address that had previously created an account, so it was easy for engineering to incorporate FingerprintJS's VisitorIDs as an additional check on account creation. Every time a user attempted to submit a review, the company would check whether that visitorID had previously submitted a review associated with the specific service (or serviceID). If they had, the user would be blocked from submitting another review.

Using this methodology the company was able to catch 500 additional fake reviews a day, making a large dent in their review fraud problem with this one simple solution.

Future growth into other fraud prevention use cases

While the company has been focused on fake review prevention thus far, they plan to use FingerprintJS Pro for other fraud use cases in the future due to its flexibility. One of the main future use cases for FingerprintJS is preventing referral fraud and new account signup fraud, as new customers are incentivized with significant discounts. Another possible use case is in preventing card cracking, card testing and other forms of payment fraud to reduce revenue losses from chargebacks.