

Stopping Airline Seat Spinning

How Cequence Defends Booking Revenue Against Automated Inventory-Hold Fraud

How Seat Spinning Undermines Revenue and Customer Trust

Seat spinning drains airline revenue by abusing a routine feature of the booking flow: the temporary hold that keeps a seat reserved while a customer finishes deciding. Attackers use automated clients to place seats into that pending state, let the holds expire without paying, and repeat the cycle thousands of times a day. The goal is profit: manufacture enough scarcity that fares climb, then cash in through resale channels, competitive positioning, or simply denying inventory to a rival. This isn't a nuisance-bot problem — it's a direct attack on revenue integrity and customer trust, exploiting how an application is supposed to work rather than any flaw in the code. That's why most security tooling never catches it.

When bots lock inventory at scale, revenue systems see the activity as genuine demand and raise fares. The distortion spreads in three directions:

- **Revenue** — seats that should sell at the right time and price never do, and pricing engines chase demand that was never real.
- **Operations** — look-to-book and booking-velocity metrics get skewed, so forecasting and inventory decisions run on bad data.
- **Customer trust** — travelers watch "sold out" flights reappear at a higher price and lose confidence in the brand.

In markets with long no-cost hold windows — common across Asia Pacific — a spinner's window is even wider.

Why Traditional Controls Miss It

Seat spinning is business logic abuse powered by automation, and traditional defenses were built for a different fight. CAPTCHA is largely irrelevant here; the abuse happens before the login and payment stages where challenges fire, and AI-assisted bots solve them anyway. IP blocking barely dents it: spinners rotate across residential proxies and cloud infrastructure, so reputation-based blocking mostly generates false positives. Static rate limits miss the intent — a bot that throttles itself to look human never trips them. And web application firewalls (WAFs), tuned for syntactic attacks like SQL injection or cross-site scripting, can't tell a shopper browsing fares from a bot holding seats it will never buy.

Every one of these controls judges traffic one request at a time. Seat spinning only becomes visible across the full booking journey.

What Customers See

- Seats available, then gone at payment
- "Sold out" flights that reappear hours later
- Fares that swing up and down in minutes

Why The Usual Controls Fail

- **CAPTCHA:** triggers too late, solved by AI anyway
- **IP blocking:** proxies rotate endlessly
- **Rate limits:** bots self-throttle to blend in
- **WAFs:** no sense of business context

The Defense That Works: Behavioral Intent Analysis

Stopping seat spinning means analyzing booking behavior and purchase intent across the entire booking flow, not classifying individual requests as human or machine. Cequence models how real customers research, select, and buy, then flags the journeys that don't add up — repetitive holds that never reach payment, searches and fare recalculations unlike genuine shopping, geographically impossible sessions. These surface only as coordinated analysis, not isolated checkpoints.

Once an entity is identified as spinning inventory, Cequence prevents seats from being held and taken out of inventory in real time, protecting both revenue and the customer experience. Human-in-the-loop enforcement short-circuits the impossible journeys bots attempt, so airlines get precision instead of friction. Since the entity is now known, other harmful activities like price scraping are prevented, dramatically reducing traffic volumes that otherwise needlessly strain infrastructure.

Cequence was built from the beginning to handle sophisticated business logic attacks, not just volumetric floods. That matters even more now, as AI agents originate a growing share of traffic — some of it legitimate activity operating on a customer's behalf that airlines want to allow. Telling human from synthetic is no longer enough; the platform has to tell good intent from bad. That deep understanding of user journeys and application logic is what lets Cequence catch unlikely and impossible journeys that request-level tooling never registers.

CASE STUDY

A Leading International Airline

A leading international airline engaged Cequence to put an end to suspected seat spinning on its booking APIs. The attackers had automated an entire booking flow where their scripts solved a CAPTCHA puzzle, grabbed a session token, searched flights, pulled seat prices and availability, and filled in fictional passenger details, creating a temporary booking that then held the seats for 30 minutes. The bots let each hold expire unpaid, then started again.

Cequence began in monitoring mode, analyzing and scoring traffic but not blocking it. The team used that time to map how real paying customers move through the booking flow. Compared with real customers, the bots' "impossible journeys" stood out. They called booking APIs directly and didn't trigger the normal calls that customer browsers make. Single session tokens were observed jumping between IP addresses in different countries within seconds. The confidence-based automatic mitigation policies were quickly tuned, and in the last week of monitoring, Cequence went through daily reports with the airline showing exactly what would have been blocked. Once the airline switched on blocking, it was acting on evidence rather than estimates.

Once blocking started, the attackers fought back. They re-ran their booking flow scripts at four and five times the velocity, tried new behaviors, and attempted bookings jumped to around 100,000 a day. Attack detection held. Within a week, attempts fell to as few as 20 a day. Every blocked hold had cost the operators residential proxy access costs and the effort of getting past the airline's existing defenses, and now it earned them nothing, so they gave up.

Over four weeks, no legitimate customer reported being blocked, with orders and revenue rising while blocking was enabled. The airline's annual seat sale normally brings a 15- to 50-fold traffic surge, in no small part due to seat spinning activity. This time it was the first sale in years to run without infrastructure problems. Without the bots' scripted spike at the start of the sale, traffic built up naturally and the systems scaled as needed, keeping up with demand.

>99.9%

detection accuracy
over a four-week
proof of value

0

false positives
traced to Cequence

~99.96%

reduction in
malicious booking
attempts within a
week of blocking

1st

successful seat sale
event unencumbered
by malicious traffic

The Strategic Imperative

Seat spinning is economically motivated and strategically executed, and its blast radius crosses security, revenue management, digital commerce, and distribution. It degrades business performance and customer perception at the same time.

The answer isn't more friction for legitimate travelers, but deeper visibility into intent. A behavioral engine powered by machine learning, cross-session analysis, and real-time enforcement lets airlines detect malicious booking intent before inventory gets distorted, protect pricing models from manipulation, stabilize availability for real customers, and keep demand signals clean for revenue optimization. In an era of AI-assisted, economically motivated fraud, seat spinning is a preview of how business logic abuse is evolving across digital commerce.