spec | Spec ID

Executive Summary

Modern online attackers have become increasingly sophisticated in evading detection by "sharding" their identity signals. They rotate through countless IP addresses, user agents, email addresses, physical addresses, and other data points to appear as entirely different users with every interaction. This fragmentation tactic undermines traditional fraud defenses, which rely heavily on linking consumer behavior through persistent identifiers.

Historically, companies have used JavaScript-based device fingerprinting to create unique device IDs, allowing them to track and correlate user sessions. However, this approach is rapidly deteriorating. Privacy-aware consumers are increasingly blocking JavaScript, and modern browsers are restricting fingerprinting capabilities by default. Compounding the challenge, a growing volume of site traffic now originates from AI agents and headless browsers that don't execute JavaScript at all which leave traditional device-based identity systems blind.

Spec addresses this visibility gap with the Spec ID: a powerful, JavaScript-free identifier generated from network heuristics observed passively at the network edge.

Without relying on client-side execution, Spec can assign consistent, durable identifiers to traffic even in headless browsing contexts or when JavaScript is disabled. This enables organizations to tie together thousands of seemingly unrelated events to reveal the coordinated activity behind fragmented attacker identities.

By providing a reliable and persistent view of identity, Spec helps online brands and marketplaces detect and disrupt sophisticated fraud patterns that would otherwise go unnoticed. In an era where attackers exploit every blind spot, Spec restores clarity and control.

Introduction

In today's digital landscape, identity is everything. The ability to reliably associate user activity across time, sessions, and devices underpins a broad range of online protections from fraud prevention to account security, personalization, and trust. Yet this foundation is increasingly under attack.

Sophisticated adversaries have developed techniques to fragment their digital footprints, a tactic we refer to as "identity sharding". By rotating through an array of IP addresses, user agents, email addresses, physical addresses, and other data elements, attackers intentionally avoid persistent detection. The result is a set of seemingly disconnected interactions that evade traditional fraud models, allowing malicious activity to slip past unnoticed.

Companies have traditionally leaned on device fingerprinting via JavaScript to reconnect these fragments. These techniques, which create a unique identifier based on device and browser characteristics, once served as a critical tool in tying together user behavior. But the environment has shifted. Consumers and browsers alike have moved to block fingerprinting technologies in the name of privacy. At the same time, the rise of AI agents and headless browsers, which don't run JavaScript at all, has rendered client-side fingerprinting incomplete or even obsolete in many cases.

This growing visibility gap creates a dangerous blind spot for digital businesses. Without a way to consistently identify users across intentionally fragmented signals, fraudsters operate with near impunity.

Spec's platform offers a new path forward. Unlike traditional methods, Spec ID does not rely on JavaScript. It's a server-side identifier derived from passive network heuristics, meaning it works even when client-side code is blocked or absent.

Spec ID gives online brands and marketplaces the power to stitch together thousands of events that would otherwise look unrelated, enabling them to detect attackers who are deliberately hiding in plain sight.

This white paper explores the mechanics of identity sharding, the failure modes of legacy detection strategies, and how Spec's JS-free identity infrastructure restores the integrity of session linking and threat detection in a fragmented, headless world.

The Problem: Identity Sharding as a Fraud Strategy

Sharding, in the context of online abuse and fraud, is the intentional fragmentation of an attacker's identity across multiple sessions or touchpoints. The goal is to appear as many different users or devices, thereby bypassing systems that rely on user provided elements to detect fraud patterns.

IP Rotation via Proxy Networks

What it looks like:

A fraud ring using residential proxy services like Bright Data or mobile emulators to rotate IP addresses every few seconds or per attempt.

Why it works:

Detection systems that look for repeat IPs across suspicious behavior are blind because each request appears to come from a different part of the world or user base.

Email Address Cycling

What it looks like:

Attackers create hundreds of throwaway Gmail or ProtonMail accounts using automation, with slight variations like jane.doe+1234@gmail.com, janedoe5678@protonmail.com. Increasingly, they also use email relay services like Apple's Private Relay or Sign in with Apple, which generate anonymous addresses like xya1b2c3@privaterelay.appleid.com tied to real inboxes.

Why it works:

Fraud systems that rely on email address correlation to tie behavior across orders or sessions are easily bypassed. Relay addresses are unique per site and session, so the same actor can appear as many distinct users without managing multiple inboxes.

Device Fingerprint Tampering

What it looks like:

Automated bots randomize browser headers and spoof screen resolution, language, and plugin values, creating new device fingerprints per request.

Why it works:

This breaks linkability via traditional JS-based fingerprinting.

Headless Browsers with No JavaScript Execution

What it looks like:
AI agents or scripted bots operate using tools like Puppeteer or Playwright in full headless mode, skipping JS execution entirely.

Why it works:
No JS means no device ID, so every session appears as a clean slate.

Payment and Shipping Address Permutations

What it looks like:
Attackers use real stolen addresses but vary formatting (e.g., “123 Main St Apt 4B” vs. “123 Main Street #4B”).

Why it works:
Weak normalization logic treats each address as unique, preventing detection of reuse.

Mobile App and Web Context Switching

What it looks like:
A single attacker toggles between mobile app and mobile browser (or desktop), breaking continuity between session data pipelines.

Why it works:
Many brands silo traffic from different platforms, losing the cross-channel view.

In isolation, each of these tactics may seem minor. But when combined into a cohesive attack that rotates IPs, spoofs devices, uses different emails, and jumps between user agents, attackers make it nearly impossible for traditional detection systems to connect the dots across the myriad touchpoints that make up a typical website or app. The attacker effectively becomes a thousand ghosts, each operating below the radar.

Why Sharding Works

Sharding works because it exploits blind spots in traditional fraud models and disrupts the linkability between signals that detection systems depend on. Most fraud and risk frameworks are built around assumptions of continuity, shared sessions, persistent identifiers, or clear login boundaries. But today’s attackers are adept at severing those links, making themselves appear as hundreds or thousands of independent users.

At the same time, most brands fail to comprehensively leverage the full spectrum of behavioral and contextual data available to them. This isn’t just a visibility issue, it's a modeling issue. Attackers increasingly operate across both logged-out and logged-in states, probing the system for vulnerabilities, rate limits, or signal response behaviors before launching actual credential-stuffing or account takeover (ATO) attacks.

A common pattern looks like this:

- An attacker floods the login page or password reset flow with thousands of credential attempts using rotated IPs and email addresses.
- They observe how the site responds, identify where captchas are triggered, reverse login engineer errors, observe delays, reverse engineer the thresholds at which risk mitigations will trigger, all without ever logging in.
- Once they find working credentials, they shift to logged-in experiences: viewing stored cards, initiating withdrawals, or changing account info to prepare for full takeover.
- These interactions, while part of a single attacker’s journey, appear fragmented to most detection stacks because they span across IPs and even site subdomains or mobile vs. web.

Fraud platforms that only analyze post-login behavior or rely on JavaScript-based identity signals miss the pre-attack reconnaissance phase entirely which is where behavioral anomalies are often most evident. And because these systems rarely link probing behavior to attack execution, detection rules remain narrow, reactive, and easily bypassed.

To stay ahead, online brands must adopt identity models that:

- Operate across both authenticated and unauthenticated user states
- Connect events across time and across channels
- Don't depend on fragile signals like cookies or device fingerprints
- Can tie together fragmented behaviors into coherent attacker profiles

Sharding only works when your system can't see the whole picture. Spec's architecture is built to close that gap.

Impact on Online Brands and Marketplaces

The rise of identity sharding has significant consequences for online brands, especially those operating at a large scale. As attackers fragment their digital identities to bypass traditional detection methods, brands and marketplaces lose visibility into the true nature of user activity. The downstream impacts are wide-ranging and expensive.

Increased Fraud Losses

Without the ability to tie together fragmented behavior, fraud and abuse go undetected. Credential stuffing, account takeovers, and payment abuse succeed more frequently because the attacker is misclassified as a new, legitimate user each time. The result is:

- Escalating fraud losses
- Growing chargeback rates
- Decreased customer satisfaction
- And added operational burden for fraud and trust & safety teams.

Operational Mayhem

When detection models repeatedly miss coordinated attacks or trigger false positives against legitimate users, it damages internal confidence in the system. Teams become hesitant to act on signals, leading to:

- Manual review bottlenecks
- Reduced automation
- Introduction of friction experiences such as captchas
- And an over-reliance on reactive measures

Poor Customer Experience

As companies tighten controls to compensate for detection gaps and deploy solutions such as requiring additional verification, slowing down onboarding, or introducing friction, legitimate users are caught in the crossfire. High-friction experiences cause:

- Abandonment of sign-ups or checkouts
- Damage to brand perception
- And reduced customer lifetime value

Unique Risks for Online Marketplaces

Marketplaces face a uniquely amplified version of these challenges because they must balance the needs of two distinct types of users on their platform: service providers (sellers, freelancers, hosts) and clients (buyers, hirers, guests), which present a unique challenge to preserving trust and fairness between them.

Reputation Hijacking and Sharded Abuse

Attackers posing as service providers may shard their identity to:

- Create multiple “clean” accounts to bypass new-user scrutiny and blocked accounts
- Farm fake reviews or artificially inflate their ratings
- Contact buyers in prohibited ways or to drive business off platform
- Test platform policies and pricing structures without being detected as the same entity

Clients utilizing the platform may:

- Use relayed or disposable identities to repeatedly exploit promotional offers
- File fraudulent complaints or payment disputes without accountability
- Contact providers in prohibited ways or perpetrate off platform scams

Broken Trust Loops

In marketplaces, trust is multi-directional. If sharded identities result in a provider being defrauded or a client having a dangerous interaction, the platform itself is held responsible. These breakdowns manifest as:

- Public reputation damage
- Loss of platform stickiness
- And reduced participation from high-quality providers

Operational Escalations

Marketplace platforms often build manual or semi-automated workflows to manage trust and safety. When identity fragmentation increases, so do:

- Support tickets and escalation volumes
- Dispute resolution times
- Friction for all providers and clients
- And internal costs to “patch” detection gaps manually

Traditional Solutions and Their Breakdown

Online fraud detection systems were designed in an era when most traffic came from full-featured browsers, with cookies enabled and JavaScript reliably executed. Identity was relatively stable and attackers weren’t using AI-driven automation or residential proxies at scale. This section explores how traditional fraud prevention mechanisms are now routinely bypassed, creating massive blind spots for modern security teams.

JavaScript-Based Device Fingerprinting

Marketplaces face a uniquely amplified version of these challenges because they must balance the needs of two distinct types of users on their platform: service providers (sellers, freelancers, hosts) and clients (buyers, hirers, guests), which present a unique challenge to preserving trust and fairness between them.

How Device IDs Are Generated

Fraud systems have used JavaScript to gather entropy from the user's browser and device such as screen resolution, fonts, timezone, and installed plugins to create a persistent device fingerprint. This fingerprint is then hashed into a unique device ID, which can be used to recognize users across sessions, even if cookies are cleared.

Why Companies Have Relied on It

Device fingerprinting was a powerful tool for linking behaviors across accounts, detecting emulators or automation, and recognizing repeat attackers. It allowed brands to track users without requiring logins, enabling real-time detection of anomalies like rapid sign-ups or payment retries from the same machine.

But this approach depends entirely on one thing: JavaScript execution. And that dependency is now a liability.

The JavaScript Blockade

Consumer Privacy Tools

More users than ever are deploying privacy-enhancing tools like browser extensions, VPNs, and anti-tracking software that actively block JavaScript or fingerprinting libraries. Some privacy-focused browsers even block them by default.

Browser-Level Changes

Modern browsers have implemented structural protections:

- Safari’s Intelligent Tracking Prevention (ITP)
- Firefox’s Enhanced Tracking Protection (ETP)
- Brave’s aggressive anti-fingerprinting stance

These features reduce the entropy available to fingerprinting scripts or block them entirely.

Headless Browsers & AI Agents

Increasingly, traffic is coming from automated clients: headless browsers, AI agents, and scrapers that don’t run JavaScript at all. They skip rendering, run faster, and don’t generate a device ID, making them invisible to fingerprint-based solutions.

The result? A vanishing footprint, where legacy systems lose their primary method of establishing identity.

Broken Feedback Loop

Modern fraud detection relies on the ability to correlate an outcome like a chargeback or account takeover to earlier signals of suspicious behavior. This is critical for training models, tuning thresholds, and creating automated defenses.

- Device IDs are missing
- IPs are rotated every session
- Emails are disposable
- JavaScript is blocked

The system can't link the symptom (chargeback) back to the cause (attacker behavior). It's as if every bad event came from a different, unknown actor. This breaks the feedback loop that most fraud systems rely on to learn and adapt.

In other words, even when the business eventually detects fraud (e.g., through a payment dispute or user report), the signals leading up to it are fragmented and anonymized thus rendering them useless for prevention or investigation.

Spec's Solution: Spec ID

In a threat landscape increasingly defined by fragmentation, headless browsing, and signal obfuscation, traditional methods of linking user behavior are falling short. Device fingerprints fail without JavaScript. Cookies vanish in incognito sessions. IP addresses rotate by the second.

Spec was built to solve this new generation of detection problems and at the heart of its approach is Spec ID.

What is Spec ID?

Spec ID is a persistent, JS-free identifier that allows Spec to link together user behavior even when attackers are intentionally fragmenting their signals. It is generated using proprietary network-level heuristics and metadata observations. By removing the dependency of javascript, attackers see no indication that they are being monitored.

This allows Spec to assign a unique identifier to every actor, even when:

- They're operating in headless browsers
- JavaScript is disabled or blocked
- They switch devices or user agents rapidly
- They never log in or set a cookie

It works passively, from the very first network request even on the very first page loads.

How It Works

Spec ID is derived from a combination of behavioral, temporal, and network-layer signals that attackers typically overlook or can't manipulate easily. These include but are not limited to:

- IP address patterns (not just the IP, but its movement and resolution history)
- HTTP header ordering and normalization characteristics
- Temporal cadence and navigation patterns
- Cross-request behavioral context

These signals are combined and processed in real time by Spec's detection engine to create a stateless, yet deterministic identifier that can persist across sessions even when all traditional markers are gone.

The result: a lightweight, robust ID that is nearly impossible for an attacker to evade, but invisible and frictionless to the end user.

Why It Matters

In practical terms, Spec ID allows online brands to

- Reconstruct attacker journeys across login states, devices, and endpoints
- Link together thousands of otherwise disconnected events to turn noise into signal
- Expose coordinated fraud campaigns that would appear benign in isolation
- Respond faster and more precisely, with minimal false positives

Where most fraud systems are limited to post-login behavior or dependent on browser execution, Spec's edge-native architecture gives it visibility into every interaction, whether it's from a legitimate consumer or a script probing your forgot-password endpoint.

This also enables a more accurate feedback loop because outcomes like account takeover or payment fraud can now be retroactively tied to the originating actor, even if that actor masked their identity across dozens of micro-sessions.

Designed for a Headless Future

The shift toward AI agents, server-side automation, and non-traditional clients means that the future of traffic is increasingly headless, stateless, and adversarial. Spec ID was built from the ground up for this world. It doesn't rely on the browser to behave predictably. It doesn't assume cookies will persist. It doesn't need the user to be logged in.

It simply watches. It learns. And it links.

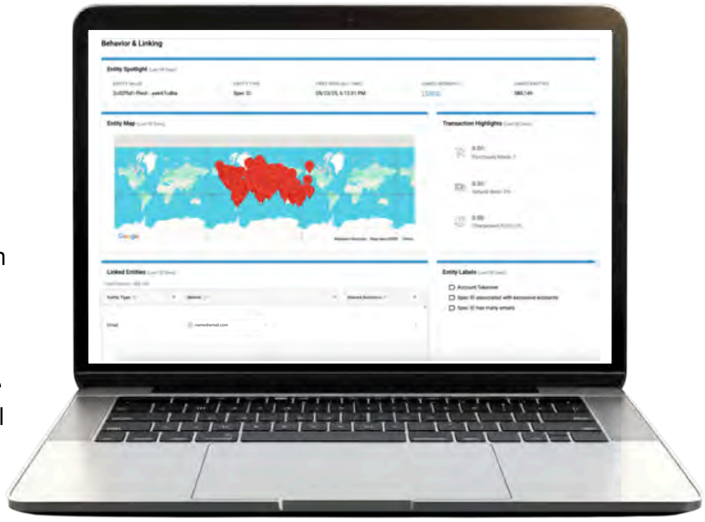
Real-World Impact: Rebuilding the Graph

When attackers intentionally fragment their identity by rotating IPs, spoofing devices, using disposable emails, and automating actions across global infrastructure, most fraud systems lose the thread. They treat each interaction as a separate actor, unable to connect the dots between reconnaissance and exploitation, or between early-stage probing and high-value compromise.

Spec changes that.

By generating a persistent, JS-free Spec ID based on passive network signals, Spec rebuilds the connective tissue between events that would otherwise look unrelated. This allows fraud teams not only to see through the attacker's camouflage, but also to reconstruct the full graph of malicious behavior across endpoints, user states, and timeframes.

Case Study:
One Attack, One Spec ID



In a single large-scale account takeover (ATO) campaign, Spec observed what initially appeared to be uncorrelated traffic: login attempts, password reset requests, and session anomalies coming from around the globe.

Traditional tools would have treated each request as a separate incident, missing the coordination entirely. But Spec resolved all of these events to a single Spec ID to reveal the true scale and structure of the attack.

Here's what that one Spec ID was responsible for:

- **471,309 unique email addresses** were used, likely sourced from recent breach lists
- The traffic came from **44,799 unique IP addresses,** all rotated aggressively using proxy services
- These IPs were spread across **5,491 unique ISPs,** a clear sign of residential proxy abuse
- Activity originated from **205 different countries**. The only country from which no events were seen? North Korea

From the outside, it looked like hundreds of thousands of users. From Specs platform, the attack patterns were clear: it was one attacker, operating through a globally distributed, fully automated infrastructure.

Spec didn't need JavaScript. It didn't need the attacker to log in. It didn't need to observe device characteristics or read cookies. It simply listened at the edge, captured the raw network-level signals, and stitched the attacker's identity back together in real time.

What This Unlocks

In practical terms, Spec ID allows online brands to

- See the full extent of the campaign in real time, not weeks later via manual review
- Block the attacker's infrastructure preemptively, without waiting for user complaints or chargebacks
- Recalibrate risk scoring for sessions that matched the attacker's behavioral signature
- Create linkages across other attack attempts in different parts of the product (sign-up, checkout, support)

Beyond Detection: Operational Clarity

This example highlights how Spec transforms the fraud detection process

- What once looked like statistical noise becomes a high-confidence signal
- What once required case-by-case manual analysis becomes automated and scalable
- What once felt like a fragmented war against anonymous threats becomes a clear, traceable battle against real adversaries

With Spec, organizations aren't chasing after ephemeral IPs or throwaway emails, they're targeting the persistent behavioral fingerprint of the attacker themselves. By leveraging Spec's honeypots, organizations not only protect their infrastructure and their customers, but they harvest intelligence from the attack to better defend against future ones.

Conclusion

Fraud has evolved. Today's attackers are not only technically sophisticated, they are operationally distributed, privacy-aware, and increasingly automated. They use fragmentation as their strategy: rotating IPs, switching devices, cycling through disposable emails, and leveraging headless browsers or AI agents that evade JavaScript entirely. This intentional sharding of identity breaks traditional fraud systems by design.

At the same time, businesses are struggling with limited visibility, overreliance on brittle client-side signals, and a growing gap between attacker behavior and the signals they can actually see. The breakdown of device fingerprinting, combined with the blind spots in logged-out and pre-authenticated flows, has made fraud harder to detect and even harder to understand.

Spec was built to close this gap.

With its JS-free, server-side approach to identity, Spec ID enables organizations to see what others miss: the continuity behind fragmentation. By linking thousands of seemingly unconnected events into a single actor profile, Spec empowers fraud teams to:

- Rebuild the attacker graph across time and systems
- Detect behavior patterns that start before login and persist across sessions
- And break the feedback loop attackers rely on to remain invisible

The result isn't just better detection, it's a fundamentally stronger fraud defense posture. One that works across channels, platforms, and evasive behaviors. One that holds up even when JavaScript is blocked and browser signals are stripped away. One that's designed for the AI-driven, headless web of today and tomorrow.

Spec gives you back the clarity, continuity, and context you've been missing. In a fragmented world, it brings identity back together.