CLEARTRUST • BOTMAN PRODUCTION

Fingerprint Traps — Technical Reference Guide

An overview of automated device fingerprint analysis and fraud signals

1. What is a Fingerprint Trap?

A **fingerprint trap** is an automated fraud detection mechanism that analyzes the behavior of a device fingerprint (represented by the `ac` parameter) over time across incoming requests. Rather than relying on static signatures, a trap identifies anomalies and behavioral patterns that are statistically improbable for a standard human user.

These evaluation routines actively monitor for key invalid traffic (IVT) indicators, including:

- A single device fingerprint rapidly generating requests across multiple distinct IP addresses.
- A single IP address cycling through an unnaturally high volume of distinct device fingerprints.
- Impossibly compressed event rates that exceed human physiological capabilities.
- Inconsistent browser, operating system, or environmental signals paired with an identical fingerprint.
- Artificially uniform or exact mathematical patterns in mouse movement or touch coordinates.

When an evaluation routine identifies that a threshold has been breached, the request is flagged as **Invalid Traffic (IVT)**, and labeled with a specific trap code and severity score.

2. Properties Used in Evaluation

Traps are contextualized and evaluated at the `Business ID + Traffic Source + Placement Value` level. This ensures thresholds adapt dynamically to specific traffic segments rather than applying broad, global limits across distinct customer environments.

Property	Field Source	Description
Fingerprint	`ac`	The unique hardware/browser device token extracted via the client-side tag.
IP Address	`ip`	The network client IP address transmitting the event.
User Agent	`ua`	The raw browser/device user-agent string.
Business ID	`bizid`	The unique identifier for the customer campaign or business entity.
Source / Value	Session Scope	The traffic source taxonomy and granular placement context.
Device Type	`uaparams.DeviceType`	Categorization (e.g., Desktop vs. Mobile) used to bifurcate detection logic.
ASN	`ASN`	The Autonomous System Number associated with the routing network.
Coordinates	`x`, `y`	Granular pointer or touch telemetry coordinates (when available).

3. Severity and Scoring Framework

Trap Result Format

Every triggered evaluation yields two standardized outputs:

- **Trap Code:** A distinct 3-to-4 letter identifier pointing to the active logic family (e.g., `SFM`, `SIM`).
- **Trap Score:** A numeric integer defining the severity of the threat classification.

Severity Scale

Score	Classification	System State Identifier
0	Clean Traffic	`CL` (No threats identified)
1	**Category 1 — High Risk**	`CAT1_TRAPID` (Critical Invalid Traffic)
2+	Lower Severity Threat	Configured warning or sub-threshold indicators

> **System Logic Note:** If multiple distinct traps execute simultaneously on a single request, the production pipeline implements a `min(score)` selection rule. Consequently, **Score 1 represents the highest-priority, high-risk validation tier**.

4. Production Execution Flow

The clear, sequential processing pipeline for real-time risk assessment runs as follows:

Step 1: Ingestion	The incoming API request payloads are accepted and fully parsed.
Step 2: Whitelist	Check Fingerprint Whitelist (`FIPWL`). If matched, execution terminates early with a **Score 0 (Clean)**.
Step 3: Blocklist	Check Fingerprint Blocklist (`FIPBL`). If matched, previously logged threat profiles and high-risk flags are assigned instantly.
Step 4: Evaluation	Active behavioral routines (SFM, SIM, HF5R, etc.) evaluate transactional state limits concurrently.
Step 5: Consolidation	Anomalies merge with broader traffic profiling modules, selecting the minimum score before serving the final decision matrix.

5. Fingerprint Trap Catalog

5.1 SFM — Same Fingerprint, Multiple IPs

Code: `SFM` | **Severity:** Score 1

Detection Logic: Monitors for an individual fingerprint establishing connections via an excessive number of unique IP addresses within the configured tracking window.

Context: Legitimate physical hardware relies on stable network backbones. Frequent, rapid redistribution across distinct networks implies proxy manipulation, active routing rotation, or structured bot infrastructure.

Device Class	Triggering Condition
Desktop	≥ 7 unique IP addresses mapped to an identical fingerprint.
Non-Desktop (Mobile/Tablet)	≥ 7 unique IP addresses **AND** ≥ 3 unique ASNs mapped to the fingerprint.

Default Window: 24 Hours. Configurable via parameters `SFM_COUNT`, `SFM_ASN_COUNT`, and `SFM_DURATION`.

5.2 SIM — Same IP, Multiple Fingerprints

Code: `SIM` | **Severity:** Score 1

Detection Logic: Tracks instances where a singular network IP address advertises an uncharacteristically diverse collection of unique hardware identities.

Context: Standard household or workplace networks support a predictable, finite scope of individual devices. High-density fingerprint churn out of an isolated IP implies device-id spoofing or running headless browser emulators.

Default Window: 24 Hours. Triggers when unique device fingerprints exceed the profile base limit (Default: `2`). Overridden dynamically via `SIM_THRESHOLD`.

5.3 HF5R — High Fingerprint Rate

Code: `HF5R` | **Severity:** Score 1

Detection Logic: Triggers when event velocities originating from a singular device fingerprint exceed safe parameters.

Context: Human interactions contain baseline latency. Execution spikes scaling past **5 events per second** signal structural script invocation, replayed sessions, or tag misconfigurations.

Default Window: 1 Second sampling window. Configurable via `HF5R_THRESHOLD` and tracking duration `HF5R_DURATION`.

5.4 SFMU — Same Fingerprint, Multiple User Agents

Code: `SFMU` | **Severity:** Score 1

Detection Logic: Checks for structural integrity by matching a static hardware token against changing User-Agent strings.

Context: A legitimate browser environment has fixed canvas, hardware, and OS properties. Seeing different User Agents under the exact same hardware hash points directly to user-agent rotation scripts.

Default Window: 24 Hours. Standard threshold limit is set to `2` unique User Agents before firing.

5.5 SID — Same Fingerprint, Same IP, Different UA

Code: `SID` | **Severity:** Score 1

Detection Logic: A strict validation subset of SFMU. It tracks events where both the device fingerprint and the IP address remain completely frozen, yet the User-Agent values change between requests.

Context: This is a clear indicator of client-side request injection where an attacker spoofs browser headers across sequential requests without modifying the underlying network route or hardware signature.

Default Window: 24 Hours. Fires immediately when `≥ 4` distinct User-Agent strings are mapped to the stable IP-Fingerprint pair.

5.6 SFXY — Same Fingerprint, Same Coordinates

Code: `SFXY` | **Severity:** Score 1

Detection Logic: Evaluates mouse clicks or touch-point telemetry to find repeated, pixel-perfect matches on specific coordinate vectors.

Context: Natural human motor patterns exhibit structural variability. Generating exactly identical `(x, y)` inputs across multiple events reveals automated coordinate playback tools or macro scripts.

Default Window: 24 Hours. Fires on the 10th consecutive event containing identical coordinates.

5.7 SF24 — Scripted Coordinate Progression

Code: `SF24` | **Severity:** Score 1

Detection Logic: Analyzes spatial input arrays to identify flawless mathematical progression sequences.

Context: Human input is naturally organic and imperfect. Linear or algorithmic coordinate progressions indicate automated cursor paths.

Evaluation Criteria: Captures coordinate sequences (Default: `10` points). The mathematical engine tests for three progression behaviors, allowing a tight 10% deviation tolerance:

- **Arithmetic Progression (AP):** Fixed, uniform pixel step changes.
- **Geometric Progression (GP):** Exponential pixel step scaling ratios.
- **Harmonic Progression (HP):** Structured reciprocal curve progressions.

6. Configuration Parameters (Trap Map)

Threshold limits can be customized per customer profile via **Trap Map Settings** (`tset`). Values are maintained in real-time within highly responsive memcache structures as `TRAPVARS-{bizid}-{source}-{value}` .

Parameter Family	Operational Control Profile
`SFM_*`	Manages IP boundaries, ASN count limits, and evaluation time horizons.
`SIM_*`	Adjusts allowed unique fingerprint densities relative to target IP spaces.
`HF5R_*`	Fine-tunes sampling windows and event counts for rate-limiting.
`SFMU_*`	Sets permissible User-Agent variance limits per device token.
`SF24_*`	Configures coordinate array sizes and strictness tolerances for progressions.

7. Operational Customer Guidelines

1. **Profile Activation:** Ensure target traps are explicitly set to active within the administrative customer profile for the required source/value channels.
2. **Whitelist Management:** The `FIPWL` bypass takes absolute priority. Restrict this configuration strictly to internal automated QA infrastructure or verified testing units.
3. **Telemetry Dependency:** Spatial coordinate routines (`SFXY` and `SF24`) require complete integration telemetry. If coordinate properties are omitted from client payloads, these routines will remain inactive.