Nick Clark · Published July 29, 2026 · PDF

HUMAN Security: Bot Networks and Metering by Source

A single adversary with a script can present a thousand fresh identities to an autonomous agent, and if that agent charges its refusal counter per assertion, the thousand identities buy a thousand chances to drive it to withhold action. Bot mitigation platforms such as HUMAN Security address a closely related concern at the traffic layer, and publicly describe their purpose as separating automated traffic from human traffic. U.S. Provisional Application No. 64/117,812 describes a different placement for that concern: an agent that derives origin-equivalence classes from its own lineage and identity records, charges the refusal counter per source of assertion rather than per assertion, and consults no registry, directory, or shared scoring service to do it.

One Adversary, One Thousand Names

An autonomous agent that accepts assertions from outside parties needs some way to stop acting when those assertions stop resolving. In the filed architecture that job belongs to a refusal counter, and a separate chapter of the same filing governs the metering procedure that drives it. Chapter 2 recites only what its own problem statement requires: increments accumulate upon both refusal paths, since artifacts that

are false produce the rejected determination and artifacts unresolvable against the agent's append-only lineage field produce the not-determinable determination, and both paths accumulate increments.

Chapter 2 then names the soft spot plainly. Volume alone drives the authorization gate of the semantic agent to the withheld state, and the volume available to an adverse party is limited only by the cost of presenting further identities. Where identities are cheap, the adversary need not defeat the agent's reasoning at all. It need only arrive often enough, under enough names, that the agent's own protective mechanism carries it to the withheld state.

That is the denial-of-service shape of a Sybil attack, and it is awkward precisely because the protection is the exposure. The question worth asking is not how large the tolerated volume should be, but what the counter should be charging against.

What HUMAN Security Does

HUMAN Security is a cybersecurity company that publicly describes its business as distinguishing automated traffic from human traffic and stopping automated traffic used for fraud or abuse. Its public materials address the categories in which bots do commercial damage, among them advertising fraud, credential stuffing and account takeover, fake account creation, scalping and inventory abuse, and malicious activity in mobile and web applications. That is a substantial and well-earned position in a hard field.

Collective observation is the idea the company's public materials put at the center of the approach. As described publicly, the platform evaluates requests against signals gathered across the properties it protects, so that what is learned from an attack pattern observed at one property can inform protection at others. The company also maintains a threat research function that investigates and publishes on botnet operations and

fraud schemes, and that published work has contributed materially to public understanding of how ad fraud and bot infrastructure operate. Nothing beyond those public descriptions is asserted here about how any of it is implemented.

Why the approach is sound on its own terms deserves saying plainly. Recognizing an adversary at one vantage point and again at another requires something that occupies both, and a shared observation layer is a legitimate and effective way to obtain that. Practitioners choose this category deliberately, wanting the benefit of everyone else's incident before it becomes their own.

What follows is not a better version of that. It concerns a different placement under a different constraint, one in which the deciding party has no shared vantage point and must reach a defensible answer from records it holds itself.

Charging the Counter Per Source

The filed chapter forecloses that condition by charging the refusal counter per source of assertion rather than per assertion. A source, in the chapter's terms, is a set of asserting parties between which the semantic agent demonstrates a relation from its own records. Under that structure the contribution of a single origin to the withholding of an action is bounded irrespective of the number of identities the origin presents.

The derivation is an ordered procedure. On receiving a conduct evaluation artifact, the agent retrieves the counterparty identity record of the asserting party, or instantiates one if none exists. It then retrieves, from the signed policy object in force, an enumeration of declared relation types, each specifying a class of lineage entry and a matching condition over such entries. For every asserting party already assigned to an origin-equivalence class, whether in the current window or a preceding one, the agent evaluates each declared relation type against the present party.

The chapter enumerates three such relation types:

- **Shared dispatch lineage**, evidenced where the append-only lineage field contains an entry recording a dispatch to the present asserting party and an entry recording a dispatch to the compared asserting party, and both entries record a common parent dispatch entry as their immediate antecedent.
- **Co-signature**, evidenced where a single lineage entry bears a signature verifiable against an identity primitive of the present asserting party and a signature verifiable against an identity primitive of the compared party.
- **Common introduction path**, evidenced where the counterparty identity records of both parties each record an introducing party and the recorded introducing parties are identical.

Each declared relation type is independently sufficient for assignment. Where any is evidenced, the present party joins the compared party's class; where more than one class is identified, the classes merge; where none is evidenced, the party is assigned to a new class. The assignment, the relation types evaluated, the entries relied upon, and the resulting class identifier are appended to the lineage field and written into the counterparty identity records of the class. The chapter is explicit that the identifier records the class to which the party is assigned and is not a determination concerning that party's conduct.

Two consequences matter for the threat model. Normalization runs on receipt of the artifact, before the increment register is consulted, so a party newly presented within a window is assigned to the class of its related parties before any increment attributable to that artifact applies. Presentation of a further identity within a window is therefore incapable of yielding a further increment where the relation is evidenced from records the agent already holds. And assignment persists across windows: only the per-class increment register is window-scoped and reset.

A harder residual case gets its own treatment: a party arriving with many identities the agent has never encountered, against which no relation can yet be evidenced. The chapter defines a severance-survival test over the edges of a class, designates a class an

untested class where every constituent edge resolves not-typeable, and separately computes whether the recorded introduction paths of the class converge upon a common ancestor entry within a depth declared in the signed policy object. Where a class is designated an untested class **and** its introduction paths converge, a cost multiplier declared in the policy object, greater than zero and less than unity, applies to that class's contribution. Where either condition fails, the class contributes at full weight. The multiplier does not exclude the class, does not suppress the determinations produced for its artifacts, and does not prevent full-weight contribution once recorded severance events accumulate.

One illustrative trace in the chapter uses a rate threshold of five, a cost multiplier of one fifth, and an introduction-path depth of two. Those figures are given by way of illustration and are not defaults. Elsewhere the quantities are policy-declared and only bounded, the declared depth being at least one.

Where the Two Architectures Part

Effectiveness is not the axis of divergence. What each design may consult, and what each produces, is.

A shared observation layer draws its strength from correlation across parties. The origin-equivalence class, by contrast, is computed by the semantic agent from its own records: without reference to a centralized registry, without query to a directory, without participation in a consensus procedure, and without coordination with a further execution node. The chapter draws that contrast against Sybil-resistance schemes that resolve identity against a shared registry or a coordinating authority, and states that no shared authority is required or consulted.

The second difference is what the class is worth outside the agent that made it. Nothing. A second semantic agent holding a different lineage field and different counterparty identity records derives, from the same population of asserting parties, a partition that

need not agree with the first. No procedure reconciles the two, no class identifier is transmitted between them, and neither agent admits a class identifier derived by the other. The class identifier is therefore not an identity attested by a third party and confers no portable standing on the asserting party. Nothing crosses between agents that could accumulate into portable reputation.

Third, normalization governs the metering alone. An artifact from a party in a class already recorded in the register still produces a determination and is still appended to the lineage field; it simply applies no increment. In the chapter's language, normalization suppresses no determination and withholds no adjudication. Controls placed at the traffic layer are generally described as deciding whether a request is served at all, which is a different and frequently desirable thing. The two behaviors are complementary rather than substitutable.

Reading the Two Together

Nothing here asks an operator to choose. A deployment can put a bot mitigation layer in front of an application and still want a per-agent answer behind it, because the two answer different questions. One asks whether a given request should be served at all. The charging procedure asks how much a given source may contribute toward carrying a specific agent's authorization gate to the withheld state, and it must answer that even for parties that were legitimately served.

Availability of shared context is the practical dividing line. Where an operator has a shared observation footprint and is content to depend on it, correlation across parties is the stronger instrument. Where an agent must instead reach a defensible answer from records it holds, whether because it runs in isolation, because its counterparties are other agents rather than browsers, or because a cross-party scoring dependency is unacceptable, the filed procedure describes how that answer is derived and recorded in the agent's own append-only lineage field.

One limit is stated in the filing rather than hidden by it. The enumeration of declared relation types is policy-declared and comprises at least one type, and the chapter notes that an empty enumeration assigns every asserting party to a distinct class, restoring the condition the chapter set out to foreclose. The protection is therefore no better than the relation types an operator declares.

Disclosure Scope

This article describes subject matter disclosed in Chapter 2 of U.S. Provisional Application No. 64/117,812, and is published in part as a defensive disclosure. It reflects one chapter of a pending application. It is not legal advice, not an offer to license, and not a representation as to the scope of any claim that may issue.

References to HUMAN Security are to public materials and are used for comparison only; no relationship, endorsement, or infringement is asserted.

Statements about HUMAN Security are limited to what the company has publicly described about its purpose and approach. No assertion is made about its internal implementation, and no comparison here should be read as a statement that any product does or does not implement any particular mechanism.

Origin-Equivalence Normalization (/origin-equivalence)

All 49 steps → (/inventive-steps)

Metering that can't be gamed by splitting one accuser into many.

Provisional application

PRIMARY TECHNICAL DISCLOSURE

– Origin-Equivalence Normalization: Metering Refusal Per Source, Not Per Assertion (/articles/origin-equivalence/origin-equivalence-normalization)

SECONDARY TECHNICAL

- Content-Blind Encounter Commitment and the Paired Encounter Receipt (/articles/origin-equivalence/content-blind-encounter-commitment)
- Cross-Receipt Equivocation Detection: Catching an Agent That Tells Two Stories (/articles/origin-equivalence/cross-receipt-equivocation-detection)
- The Untested-Origin Saturation Bound: Pricing a Flood of Freshly Minted Identities (/articles/origin-equivalence/untested-origin-weighted-saturation-bound)
- Counterparty-Side Fork Detection: Recognizing a Split Identity From Outside (/articles/origin-equivalence/counterparty-side-fork-detection)
- The Attestation Origin-Equivalence Gate: An Anti-Collusion Bound (/articles/origin-equivalence/attestation-anti-collusion-bound)
- The Bridging Party and the Deferred Merge: Charging One Windowed Increment Against Every Origin-Equivalence Class a Single Asserting Party Bridges (/articles/origin-equivalence/bridging-party-deferred-merge)
- The Common-Execution-Node Relation Type: Charging Co-Hosted Asserting Parties as One Origin (/articles/origin-equivalence/common-execution-node-relation-type)
- Receiver-Side First-Encounter Budget Decrement: Pricing a Stranger's Arrival Per Presented-Material Origin Class (/articles/origin-equivalence/receiver-side-first-encounter-budget-decrement)
- Mutual First Encounter Without Cross-Budget Coordination: Two Strangers, Two Matched Pairs, Two Separate Budget Decrements (/articles/origin-equivalence/mutual-first-encounter-no-cross-budget)

APPLICATIONS · GENERAL

- Sybil-Resistant Agent Networks: When One Accuser Becomes Many (/articles/origin-equivalence/sybil-resistant-agent-networks)
- Fraud Rings in Autonomous Commerce: Metering Per Source, Not Per Claim (/articles/origin-equivalence/fraud-rings-in-autonomous-commerce)
- Invalid Traffic and Click Fraud: Metering by Source, Not Impression (/articles/origin-equivalence/adtech-invalid-traffic)

APPLICATIONS · SPECIFIC

- W3C Decentralized Identifiers and Verifiable Credentials: Verifying an Identifier Versus Metering Its Origin (/articles/origin-equivalence/w3c-decentralized-identifiers)
- Okta Auth for GenAI: Agent Identity and Origin Equivalence (/articles/origin-equivalence/okta-auth-for-genai)
- WorkOS: Agent Identity Infrastructure and Source Metering (/articles/origin-equivalence/workos-agent-identity)

- Entra Agent ID: Directory Identity and Derived Origin Classes (/articles/origin-equivalence/microsoft-entra-agent-id)
- SPIFFE and SPIRE: Workload Identity and Origin Equivalence (/articles/origin-equivalence/spiffe-spire)
- Cloudflare Bot Management: Blocking Agents and Metering Sources (/articles/origin-equivalence/cloudflare-bot-management)
- **HUMAN Security: Bot Networks and Metering by Source (/articles/origin-equivalence/human-security)**

TERMINOLOGY

- asserting party (/glossary#asserting-party)
- origin-equivalence class (/glossary#origin-equivalence-class)

Origin-Equivalence Normalization overview → (/origin-equivalence)