

Why we refuse to treat a model’s guess as a finding

“...and how we built a platform around that refusal.”

PUBLIC EVIDENCE-GATED OFFENSIVE SECURITY NOT A SALES DECK, NOT A RESEARCH PREPRINT

SECStrike · EchoStrike

ABSTRACT

This paper describes what EchoStrike is, what it is not, and why we organised the product the way we did. It is written for security leaders, practitioners and technical partners — an attempt to say, in plain language, how we think offensive security should work once language models are in the loop. This version adds a short, intuitive account of the measurable core underneath the architecture, and points to a companion paper that develops it formally.

1	The problem we actually have	7	What the research has shown so far
2	What EchoStrike is	8	Where the work goes next
3	How the loop is supposed to work	9	What we are willing to put our name on
4	Why evidence gates exist	10	What this paper does not contain
5	ATVE, without the mythology	11	Closing
6	The core idea, made measurable		

1. The problem we actually have

Most of the last two years in “AI security” has been a story about speed. Faster recon. Faster write-ups. Faster suggested remediations. That is a real gain. It is also a dangerous one, because speed without a proof standard just produces more confident noise.

Anyone who has sat through a pentest debrief knows the difference between “this looks interesting” and “this is exploitable, here is the evidence, and we can show it again.” Language models are unusually good at the first sentence and unusually willing to dress it up as the second.

We built EchoStrike because that gap was starting to become the product. Teams were being asked to trust generated narratives. Findings were arriving without a reproduction path. Autonomy was being sold as a feature rather than treated as a control problem. None of that is abstract for us. It is the daily failure mode we designed against.

Discovering a possible weakness is not the same as proving a security condition. Proposing an action is not the same as being authorised to take it. Everything in EchoStrike hangs off those two sentences.

2. What EchoStrike is

EchoStrike is an offensive-security platform built around a closed loop: find the attack surface, prove what is actually exploitable, help get it fixed, and then prove the original condition is gone. We write that as **FIND** → **PROVE** → **FIX** → **REPROVE** because the last step is the one most tools quietly skip.

The platform is not a single scanner with a chat window bolted on. It is a set of specialised pieces that share structured evidence and engagement state, and that are deliberately kept apart so no one component can declare victory for the rest.

The pieces, as they stand today

Status matters here. We label capabilities as production, in build, research or planned so a reader does not have to guess which parts are live and which parts we are still arguing with.

STATUS	CAPABILITY	WHAT IT ACTUALLY DOES
● PRODUCTION	ReconPulse	External attack-surface discovery. Produces candidates, not verdicts.
● PRODUCTION	RedEngine / Crimson Nexus	Evidence-backed validation, engagement state, reusable verified knowledge.
● PRODUCTION	Pentest Studio	The human investigation path. Not a fallback. Part of the design.
● PRODUCTION	EchoStrike UI / API	Projects, hosts, findings, evidence, reporting, re-test, access control.
● IN BUILD	BlueFence	Cyber Twin, code-enriched context, remediation practice and replay.
● RESEARCH	ATVE	Evidence-gated reasoning and bounded-autonomy control architecture.
● PLANNED	Extended remediation loop	Broader fix ranking and attack-map-to-remediation work.

If a capability is marked research, it is not a production feature wearing a lab coat. ATVE in particular is a control architecture we are formalising on top of a working platform. Treating it as a finished autonomous attacker would be a misreading of the work.

3. How the loop is supposed to work

Find — ReconPulse

ReconPulse watches the externally observable surface and hands candidates into validation. That is the whole job. It is not allowed to promote a candidate into a finding by volume of output or by how plausible the narrative sounds. Candidates are inputs. They wait for proof.

Prove — RedEngine

RedEngine is the current centre of gravity. A claim that wants to leave the system as a reportable finding is expected to carry the execution context and evidence that produced it, and to survive controls that are allowed to refuse the claim. Fail-closed is not a slogan here. If the verification path cannot stand up, the finding does not ship.

Agents in RedEngine do not keep their own private conversational memory as the source of truth. They work from a shared verified journal. When work needs to move between stages, or across to a human, the handoff is typed state rather than a chat transcript someone has to reconstruct at midnight.

Investigate — Pentest Studio

We did not design humans out of the loop and then add them back as an apology. Some conditions need judgement, taste, and a person who has been in the environment before. Pentest Studio is that path. Escalation is an architectural outcome, not an error state.

Fix and reprove — BlueFence

BlueFence is still being built, and we say so. The intent is straightforward: take structured finding context, practice a fix in a disposable Cyber Twin, then replay the original attack path. Closure needs two things, not one. The original condition should no longer reproduce. The service should still behave as expected. A component that proposed the fix does not get to mark its own homework.

4. Why evidence gates exist

Probabilistic reasoning is useful. It explores. It interprets. It plans. It is not stable enough to be the last word on whether something is exploitable.

We ran a small internal comparison that we do not treat as a scientific result, and we should be honest about the sample size. In identical runs, 6 of 17 LLM-judge verdicts flipped. The deterministic gates did not. Separately, we found that roughly a third of what a naïve tool might report did not survive our verification gates. That is not a marketing statistic. It is the reason the gates are there.

There is a related point that matters more than people expect. Once a technique has been validated, we can hold it as reusable verified capability — an executable node tied to a technical fingerprint. Replaying a held technique can require no new model calls at all. Exploration and already-proven execution are different jobs. Mixing them is how systems start “remembering” things they never actually proved. As the next two sections make precise, this replay is not merely an efficiency: it is the one operation that restores trust once reasoning has eroded it.

5. ATVE, without the mythology

ATVE — the Adaptive Threat Validation Engine — is the research control layer sitting conceptually above observation and validation. Reasoning may recommend an action. Recommendation does not authorise execution. That is the entire idea, and it is surprisingly easy to lose once models start sounding sure of themselves.

The decision path is unglamorous on purpose:

Observe → hypothesise → assess evidence → estimate belief → estimate meta-confidence → estimate uncertainty → select action → apply policy → execute, investigate, or escalate

Three outcomes, not a confidence slider

ATVE is allowed three primary outcomes.

Execute	Investigate	Escalate
Evidence, belief, trust in that belief, causal consistency, risk and policy all support the authorised action.	The information state is incomplete. Collect more decision-relevant evidence at acceptable risk.	There may be a real hypothesis, but trust, risk, authorisation or context needs a person.

Maximum autonomy is not the design goal. Defensible autonomy is. Act when the record supports action. Look harder when it does not. Hand the work over when the boundary is reached. The next section makes “the record supports action” a number rather than a judgement call.

What the state actually contains

We publish the shape of the decision state because the shape is the argument. We do not publish weights, thresholds, customer policy or optimisation functions, and we will not.

At a given step the system holds the active hypotheses, the accumulated evidence, belief confidence, meta-confidence (how much the current belief should be trusted), aleatoric and epistemic uncertainty, novelty relative to validated conditions, causal consistency, operational risk, policy and authorisation, and any relevant adjudicated memory. A decision function maps that state onto execute, investigate or escalate.

The notation looks formal because the distinctions are formal. Belief and trust in belief are not the same number. A pile of correlated observations is not independent confirmation. Memory is context to be checked, not proof that walks in the door.

6. The core idea, made measurable

Everything above is an argument about how the system should behave. This section gives the one quantity that lets someone check the argument on a real run, in enough detail to build intuition. The full treatment — definitions, a calibration guarantee, proofs — is in the companion paper [2]; here we stay at the level of the idea.

We score the path, not the finding

The same result reached two different ways deserves two different levels of trust. A finding is not trustworthy because of what kind of finding it is; it is trustworthy because of how it was proven and how far that proof held on the way to the output. So the thing we attach a number to is the *execution path* that produced a finding, not the finding itself.

For a path π we define a **determinism bound** $B(\pi)$: an upper bound on the probability that the finding is false, given exactly the path that produced it.

$B(\pi) = P[\text{the finding is wrong} \mid \text{the path } \pi \text{ that produced it}]$

Small B means the path established the property and did not lose it. Large B means too much un-checked reasoning stands between the proof and the claim.

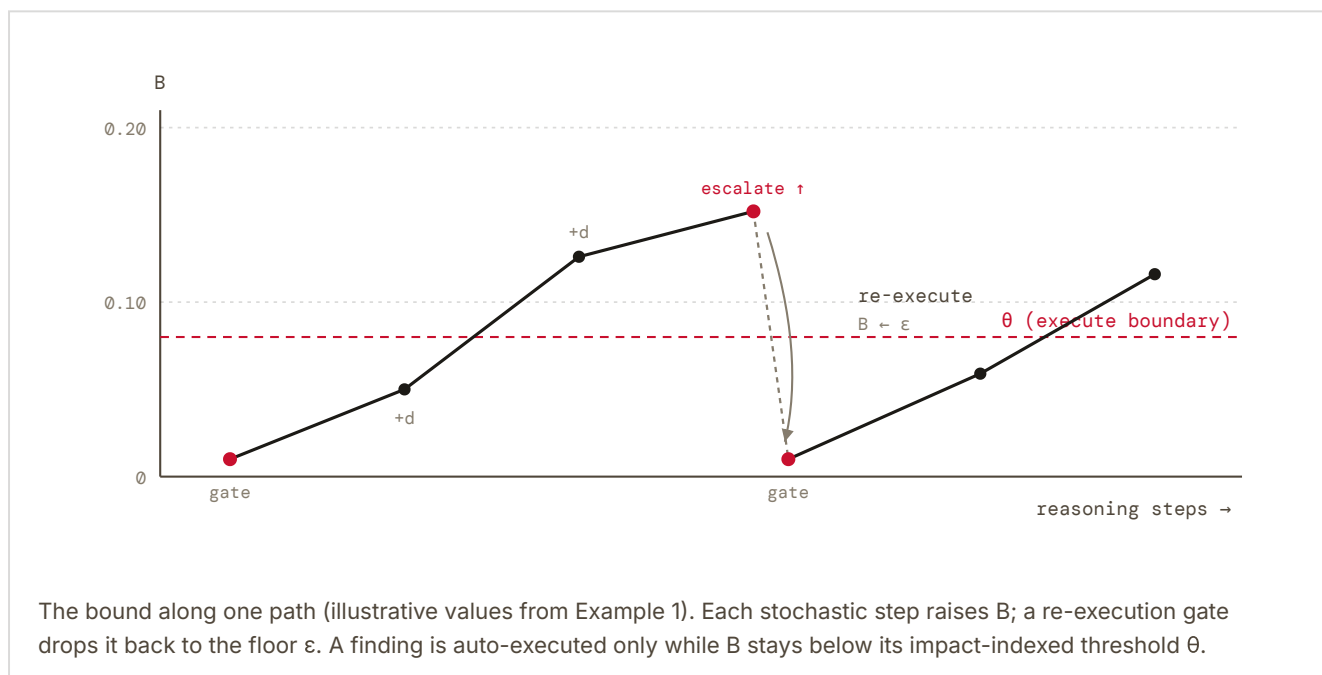
How the bound moves: a gate sets it, reasoning erodes it, re-execution restores it

The bound moves in three ways, and only three.

1. **A gate sets it.** A deterministic check — a re-run, an oracle, a monitor — establishes the property and starts trust near-certain, at a small gate residual error ϵ .
2. **Reasoning erodes it.** Each non-deterministic step adds a measured divergence d — how much that step's output varies on identical inputs — and trust decays with every uncertain step.
3. **Re-execution restores it.** Replaying a held, already-validated check resets the bound back to the gate floor, with no new model calls, however far it had drifted.

Putting those together, over the reasoning steps since the last gate the bound takes a simple multiplicative form:

$$B(\pi) = 1 - (1 - \epsilon) \prod_{\text{steps}} (1 - d).$$



Two things in this picture matter more than the algebra. First, re-execution is the only operator that *resets* B . A committee of models cannot: agreement only stacks more correlated, non-deterministic steps, which can at best fail to make things worse. This is the precise reason the platform's replay of validated checks — not a vote among models — is its real source of trust. Second, the bound makes the right thing visible: a long stretch of un-checked reasoning has a loose bound by construction, which tells you exactly where to add a gate.

From a number to a decision

A measured bound becomes a graduated, auditable authorisation. Calibrated against re-execution ground truth — did the proof actually re-run and still hold — the bound drives the three outcomes rather than being asserted alongside them:

Execute	Investigate	Escalate
The bound clears the impact-indexed threshold θ and the action is in authorised scope — act autonomously.	A real lead, but the bound can still be tightened by collecting more evidence at acceptable risk.	Trust, scope or policy needs a person — hand it to a Pentest Studio expert.

The threshold θ tightens as impact rises: a higher-stakes action must clear a tighter bound. Calibration is tuned so that, among the findings the system auto-executes, the error stays below a target δ — a finite-sample, distribution-free guarantee. When there is too little evidence to calibrate, the policy blocks and says so, rather than reporting a reassuring zero.

WHAT “PROVABLY BOUNDED AUTONOMY” MEANS

Two different guarantees, held apart on purpose. **Scope is deterministic:** an information-flow check decides whether an action is inside what we were authorised to touch, and injected text cannot move it. **Trust is calibrated:** a stated, tested limit on how often an in-scope action may be wrong. Deterministic scope, calibrated trust — not a claim that any single reasoning step is deterministic.

That distinction is also a correction we want to make plainly: the right phrase is not “deterministic safety guarantees.” The safety of the *boundary* is deterministic; the trust in a *finding* is a calibrated probability. “Defensible autonomy, not maximum autonomy” becomes a number — the system acts only where the record earns it.

Why this is the right foundation

The architecture tells you how the system is organised. This measurable core is what lets every claim be checked and every path improved. It is **computable** (the bound comes from measured divergence and gate error, with no human in the measuring loop), **measurable** (a calibrated error limit you can state and test, not a confidence vibe), **auditable** (evidence re-runs on demand; stale proof or a failed re-run forces a stop), and **adaptive** (rank the flaky paths and add a gate where it buys the most trust; thresholds flex with risk). Those four properties map directly onto the controls regulated buyers ask for — ISO/IEC 42001, the EU AI Act’s Article 14 human-oversight requirement, and the NIST AI RMF. The companion paper [2] develops each of these with definitions, a calibration theorem and a governance mapping.

7. What the research has shown so far

ATVE research uses controls that are supposed to be able to fail. That sounds obvious. It is not how a lot of “AI security eval” writing is done.

When we zeroed the calibration proxies, the measured meta-confidence advantage collapsed. When we shuffled proxy features across episodes, the meta-signal disappeared. When historical reliability contained no predictive signal, memory added essentially nothing. Those are the results you want from a null test. They mean the evaluation can detect the absence of a signal, which is the first thing it has to be able to do.

The less flattering result is the one we are willing to lead with. In a controlled PCAP transfer — 5,000 episodes, 160,000 packets — packet-derived meta-confidence did not transfer strongly enough to support selective autonomous execution. Under the configured policy, ATVE executed nothing and investigated

everything. Agreement sat at 67.5 percent. That is not a launch metric. It is a transfer gap, and we are treating it as one.

The companion paper [2] reframes this honestly. A signal that tries to predict trust from features of a belief is exactly the kind of thing that can fail to transfer. A determinism bound grounded in re-execution is anchored to an outcome the platform can actually check, which is why finding telemetry that predicts trust — so the per-step divergence is grounded in evidence, not output variability alone — is the shared open problem both papers name.

WHAT WE WILL NOT CLAIM

These experiments show that the evaluation framework can break ATVE's assumptions on purpose. They do not show production-ready meta-confidence, and they do not show a generally reliable autonomous attacker.

8. Where the work goes next

The immediate research problem is boring and hard: find technical telemetry that actually predicts when belief confidence should be trusted, using kernel-captured traffic and application telemetry from authorised isolated labs. Not toy traces. Not features that only work in the experiment that produced them.

Beyond that, we need evidence families rather than evidence counts — provenance, freshness, reproducibility, state transitions, negative controls, causal preconditions. We need temporal reasoning that can see before and after, not just a snapshot. We need a serious answer to when a human is the highest-value next action rather than the thing you call when the model gets tired. And we need memory that can be stale, mismatched or useless, and is treated that way.

The companion concept paper [2] sets out the formal version of this programme: the determinism bound and its degradation–replenishment calculus, calibration against re-execution ground truth, deterministic scope enforcement, and a re-executable assurance record. Its experiments are deliberately deferred so they can be run against the real ATVE workflow on the EchoStrike codebase rather than a stand-in, and reported with measured values.

9. What we are willing to put our name on

EchoStrike is a working offensive-security platform. Discovery, validation, human investigation, reporting and re-test exist in production. RedEngine can refuse a finding. There is substantial automated engineering coverage and a mechanism for reusable verified knowledge. BlueFence is in build and should be described that way.

ATVE is an experimental control architecture inside that programme. It formalises evidence-gated reasoning, explicit uncertainty, meta-confidence, causal consistency, policy gating and human escalation. It is not a production-ready autonomous attacker, not a substitute for professional judgement, and not proof that machine reasoning is generally reliable.

The relationship between the two is simple. EchoStrike is the operational platform. ATVE is the control architecture we are building so that, as reasoning becomes more autonomous, it still has to ask what the evidence supports and how much it should trust its own conclusion. The determinism bound in Section 6 is how that question becomes a number.

10. What this paper does not contain

We have left out exploit procedures, adapters, prompts, orchestration logic, scoring functions, internal schemas, model identities, customer policy, infrastructure and anything else that would make the implementation reproducible from the document. That is deliberate. A public architecture paper should explain the argument. It should not hand someone the machinery. The companion paper is the same: it publishes the *shape* of the formal layer — the definitions and the decision rule — not the operating thresholds, weights or policies.

ATVE is patent pending with the USPTO. This paper is a public technical disclosure.

11. Closing

The industry is going to keep putting language models next to offensive work. That is already happening. The question is whether the systems around those models can tell the difference between a fluent hypothesis and a validated condition, and whether they will stop when they cannot.

We organised EchoStrike around that question. Find is not prove. Propose is not authorise. Fix is not closed until the original path fails to reproduce and the service still stands. Autonomy, if it arrives, arrives behind evidence, policy and a person who can take the work when the boundary is reached — and now behind a number that says how far the evidence actually held.

That is the work. The rest is implementation.

FIND • PROVE • FIX • REPROVE
Observe • Reason • Validate • Control

COMPANION PAPER

Determinism-Bounded Authorization

The formal core: the determinism bound, its degradation–replenishment calculus, calibrated authorisation, and the governance mapping. A concept and preliminary-work paper.

[READ THE COMPANION PAPER →](#)

RESEARCH PROGRAMME

ATVE research

Measure → Decide → Prove: the research agenda, the data engine behind it, and how to work with the programme.

[SEE THE RESEARCH →](#)

Cite as: EchoStrike White Paper, Version 1.1. SECStrike, October 2026.

ATVE™ (Adaptive Threat Validation Engine) – USPTO patent-pending, App. No. 63/814,466.

Companion: P. Taeprasartsit and V. Khera, *Determinism-Bounded Authorization*, concept note, 2026.

This paper describes research direction and current architecture as of October 2026. Capabilities marked “research” or “planned” are not shipped product features. It is kept intentionally separate from commercial claims made elsewhere on this site.

