SAUTERA · FIELD GUIDE

The Compliance Tooling Evaluation Guide

Attestation vs. enforcement - and a rubric to score any vendor

DETECT
↓
DECIDE
↓
ACT
↓
PROVE
↓
IMPROVE

Where infrastructure trust is determined.

Most compliance tooling proves your controls existed. Almost none of it proves they still work. This guide is about telling the two apart before you sign.

The gap between an attestation and reality

Your controls passed on Monday. By Tuesday, an engineer widened a security group to ship a hotfix, a service account grew a new scope, and half your evidence describes a system that no longer exists.

That is not a tooling bug. It is a design choice baked into most of the category. A SOC 2 Type II report attests that controls operated over a period — typically 3 to 12 months — as sampled by an auditor. A CMMC assessment compares declared state to actual state on the days the assessor looks. Both are snapshots extrapolated across a window, and the extrapolation rests on an assumption that stopped being true years ago: that the state between samples is stable.

The market has grown enormous around that assumption. Compliance automation broke a real bottleneck — manual evidence collection that took weeks now takes days — and the spending shows it. But spending is up and so are losses: IBM's Cost of a Data Breach Report puts the global average breach at $4.88 million, and Verizon's DBIR keeps tracing the majority of breaches to misconfiguration and human factors — precisely the drift that appears *between* evidence-collection intervals.

The gap sits between attested compliance and enforced state. This guide gives you a way to measure that gap in any vendor: one central distinction, a scoreable six-dimension rubric, and honest buying guidance by role — including an honest account of where our own product fits and where it doesn't.

The distinction that decides everything: attestation, observation, enforcement

Vendors blur three different capabilities under labels like "continuous monitoring." Separating them is the single most useful thing you can do during an evaluation.

Attestation is a claim. A control was configured correctly when someone checked. A screenshot proving MFA was on last Tuesday, a policy document that says least privilege is required, a checklist reconciled on a schedule. It is a promise about intent — and intent decays fast. A control that was right at design time is routinely wrong by Tuesday.

Observation is a measurement. The tool read the real system state — the identity provider's live configuration, the actual bucket policy, the host's actual patch level — and recorded it. This is where most "continuous" tools actually live. It is strictly better than attestation because it reflects reality rather than intent. But it is still after the fact.

Enforcement is authority. The tool sits in the decision path and can deny, revert, or block a non-compliant action before it takes effect. It is the only mode of the three that prevents rather than reports.

The difference is easiest to see in one image: attestation tooling is a camera pointed at your systems; enforcement tooling is a gate inside them. Both can produce a clean report. Only one can stop the action that would have failed the audit if anyone had been watching at the moment it fired.

Why does this matter for trust? Because attestation and observation both produce evidence *you then have to trust*. Enforcement produces a state you don't have to trust — the compliant state becomes the only state that persists.

There is a simple tell you can apply on any vendor call: **ask them to break a control live and show you what happens.** If the answer is "you get an alert" or a red box on a dashboard, you are looking at an attestation or observation product wearing an enforcement label. If the tool denies or reverts the change and logs the decision, you have something that survives Tuesday.

Neither mode is worthless. Attestation gets you the certificate. But enforcement is what survives the assessor — and, increasingly, what survives your own automation.

Why agentic AI ended the point-in-time era

Human operators are slow, and that slowness was a hidden safety margin. A change request sat in a queue; a reviewer glanced at it; drift accumulated at human speed, roughly in step with quarterly sampling. Agentic AI removes that margin entirely.

An AI agent can open a hundred pull requests, rotate credentials, and reconfigure networking in the time a nightly scan waits to run. And the actor population has already inverted: industry identity research consistently finds machine identities outnumbering human ones by dozens to one in the enterprise — most of them with none of the controls the humans get.

Here is the scenario that breaks sampling for good. An AI agent is asked to ship a hotfix. It spins up a temporary compute instance, attaches a broad IAM role because the narrow one lacked a permission, pulls a secret, and tears the instance down forty seconds later. Every step is logged. None of it is caught. Your next quarterly sample lands on a different day; the over-privileged role existed for under a minute — long enough to matter, far too short for sampling to see. Multiply by hundreds of agent actions a day and point-in-time evidence becomes fiction.

Three things change when machine actors enter the loop:

- **The actor count explodes.** Identity is no longer 1:1 with a human, and a standing AI agent token with broad scope is a standing liability, not a one-time mistake.
- **Change frequency jumps by orders of magnitude.** A control checked once a day is unverified for 23 hours — and an AI agent acts in milliseconds.
- **Intent is inferred, not declared.** "Who did this" stops being the same question as "should this have happened."

That last point exposes the limit of zero trust as commonly deployed. NIST SP 800-207 is explicit that access decisions must be dynamic and per-request — but identity verification answers *who is asking*, not *whether this action, on this infrastructure, should proceed*. Zero Trust tells you who. It cannot tell you whether. An authenticated AI agent reaching a compromised or drifting host is a breach every identity control will wave straight through, because nothing about the requester is wrong.

The correct model is not "give the AI agent broad access and trust it." It is to bound each agent explicitly and verify its actions against policy at decision time. Which brings us to the rubric.

The evaluation rubric: six dimensions, scoreable

Run every vendor on your shortlist through the questions below. Score each question 0 (no / can't answer), 1 (partial / roadmap), or 2 (demonstrated live). Twenty-four questions, maximum 48. The absolute number matters less than the spread: a tool that scores well on provenance and continuity but zero on enforcement and unknowns is a reporting product — fine, as long as that is what you meant to buy.

`1. Evidence provenance`

Does the evidence describe reality, or a declaration about reality?

- Is evidence drawn from **observed system state** (live configs, live host telemetry) rather than declared intent (policies, questionnaires, screenshots)?
- Does each assertion **name the specific resource** and timestamp, or generalize across an account?
- Is the evidence **tamper-evident and reproducible on demand** — can you re-derive the same answer from the record?
- Can the vendor trace **one control end to end**, live: subject, observed state, policy applied, verdict? If they can't trace one, they can't govern a thousand.

2. Continuity vs. snapshot

"Continuous" collapses to near zero value if it means "nightly."

- What is the **evaluation latency** — the gap between a real change and the tool's response? Minutes, hours, or next quarter?
- What is the **population coverage** — what percentage of in-scope resources is actually evaluated, not sampled? Ask for the denominator.
- Does it evaluate **on relevant events**, or only on a polling schedule that leaves exploitable windows?
- Is audit evidence a **byproduct of normal operation**, or a separate collection job you run before the auditor arrives? If pulling proof takes a week, it isn't continuous.

3. Device-layer truth

Most compliance tooling reads control-plane configuration and stops. Risk also lives on the hosts themselves.

- Does the tool read **actual host state** — patch currency, disk encryption, exposed ports and management interfaces, endpoint protection — or only cloud-account metadata?
- Does it evaluate the **resource being accessed**, not just the requester? Identity posture on the client says nothing about the server on the other side.
- Does it track **lifecycle and supportability** — supported, approaching end-of-life, already unsupported — rather than treating version age as a proxy?
- Are known vulnerabilities **weighted by exploitability** (for instance against CISA's Known Exploited Vulnerabilities catalog), not just counted?

4. Honest-unknown handling

This dimension sounds soft. It is the sharpest trust test in the rubric. A tool that silently passes what it cannot verify is worse than no tool: it manufactures false confidence and hides risk behind a green light.

- When the tool **cannot determine a state**, does it report "unknown" — or silently pass?
- Does ambiguity **fail closed**? An unverifiable action should deny, not default-allow.
- Is "unknown" **visible in scoring and reporting**, or quietly averaged away into a green aggregate?
- Can you **enumerate your blind spots** — ask the tool what it cannot see, and get a real answer?

5. Remediation loop closure

Detection without a path to a verified fix is a very expensive alarm.

- When drift is found, is there a **defined path from finding to fix** — a recommendation with rationale and evidence, not just a severity color?
- Are consequential fixes **gated by human approval**, with a record of who approved what and when?
- Is remediation **verified after the fact** — does the tool confirm the fix actually landed and the control re-passes, with rollback if it didn't?
- Does the loop **produce its own evidence** — a closed finding that carries detection, decision, action, and proof as one record?

6. AI and agentic readiness

If a vendor pitches agentic AI, the question is not "what can it do?" but "what is it *bounded* from doing?"

- Are decisions made **per action, not per session**? Session-level trust is stale the instant scope changes.

- Are **non-human identities handled natively** — can it distinguish a human operator from an AI agent, and bound each agent to an explicit scope rather than an inherited broad role?
- Can you **replay every decision** an AI agent made and the evidence it acted on?
- Is there **reversibility** — can an AI agent's action be undone, and is there a human checkpoint before consequential changes?

Bring the forty-second-role scenario from the previous section to every demo. A pure attestation tool will tell you the role *appeared* compliant on the days it sampled. A tool worth buying will show you the exact request to attach the broad role, the boundary it was checked against, and the verdict — regardless of how long the instance lived.

A buyer's guide by persona

The right tool depends on your binding constraint, not the market's longest feature matrix. Name the constraint first.

The platform team: drift is the risk

You own live infrastructure, and your problem is that controls decay the moment infrastructure changes — which is constantly. An assessor (a C3PAO under CMMC, a Type II auditor under SOC 2) will compare declared state to actual state, and if the two diverge, the finding lands on you no matter how good the paperwork looked at scan time.

Weight dimensions 2, 3, and 5 heaviest. You need evaluation latency measured in minutes, full population coverage, host-level truth, and a remediation loop that closes with proof. The concrete test: practice-level least privilege. An attestation tool records that a role had the right scope during the nightly scan; six hours later an on-call engineer grants it admin and forgets to revoke it, and the dashboard stays green until the next scan. You want the tool that catches or reverts the grant when it happens — and treat every automated actor in your boundary, including AI agents, as a privileged identity with bounded, audited access.

The compliance owner: don't overbuy, don't stop early

If your constraint is *getting your first SOC 2 fast*, the compliance-automation platforms — Vanta, Drata, Secureframe, Sprinto — do that job well. Pick on auditor network, framework breadth, and price, and don't overbuy enforcement you won't operate yet. That is honest advice, and few vendors will give it to you.

But know what you bought: a reporting layer over sampled evidence. Weight dimensions 1 and 4 as you grow. The moment your board's question shifts from "did we pass?" to "can this happen again?", sampled attestation cannot answer it — only a continuous record of decisions can. The practical milestone to aim for: evidence as a byproduct of operations, so recertification stops being a quarterly project and becomes a report you run on demand.

The AI agent builder: the boundary is the product requirement

You are wiring non-human actors into production, and traditional compliance software assumes a human at every keyboard. That assumption fails on day one.

Weight dimensions 4 and 6 above everything. Every AI agent needs an explicit, auditable permission boundary; every action needs a per-request decision against current context, not a cached grant; and "unknown" must be a valid, blocking answer — because your agents will regularly generate states no policy author anticipated. Alerts do not scale to machine-speed actors: a human reads an alert in minutes, by which time the agent has made a thousand more calls. If a vendor cannot show you the enforcement boundary around their own agents, the automation is a liability wearing a demo.

Where SAUTERA fits — and where it doesn't

We built SAUTERA around one conviction: infrastructure trust is a running property, not a certificate. So here is our own product, scored honestly against the rubric we just handed you.

Where SAUTERA fits. SAUTERA is an infrastructure-trust platform. It scores the trust of the infrastructure itself — servers, workstations, network devices — continuously, from what is observed, never assumed. That maps directly to dimensions 1 through 4: evidence comes from live device observation and is tamper-evident; scoring is continuous rather than sampled; the device layer — patch currency, encryption, exposed surface, lifecycle status, exploitability-weighted vulnerabilities — is the core of what we read, not an afterthought; and when we can't see enough to conclude, the device reads **Unknown** — an honest gap, never a confident guess dressed up as a green light.

On dimension 5, SAUTERA runs **closed-loop remediation, human-approved**: the platform detects the issue, recommends the fix with rationale and evidence, a human approves it, the fix is applied with rollback available, and the re-scored control is the proof. The human gate on consequential action is architectural, not a feature flag — we don't sell fully autonomous remediation, and you should be suspicious of anyone who does. On dimension 6, the platform's own AI is agentic reasoning with LLM-backed analysis per device — detection and recommendation agents that decide and advocate, with a human approving consequential action. The trust engine is also callable by AI systems over MCP as a design-partner preview; there is no public endpoint yet, and we say so.

Coverage, stated plainly. Installed sensors ship for Windows and Linux today — not macOS or mobile; those are roadmap. Agentless collection runs over SSH, WMI/WinRM, and SNMP for hosts and network devices you can't install on. Cloud-side, connectors cover AWS, Azure VMs, and GCP compute. Compliance evidence packages map to SOC 2, NIST CSF, ISO 27001, and FedRAMP-aligned frameworks and export on demand.

Where SAUTERA doesn't fit. If your constraint is a first SOC 2 on a deadline, a compliance-automation platform is the faster path — buy one, and come back when drift, device truth, or agentic actors become the risk. SAUTERA is not antivirus and doesn't replace your EDR. It does not monitor facilities, power, or thermal conditions — that is not what infrastructure trust means here, and no probe of ours collects it. And our Zero Trust Completeness Audit is an expert-led **self-assessment** aligned to NIST SP 800-207 — a scored posture report, gap analysis, and remediation roadmap. It is not an accredited certification, and we will never call it one.

Zero Trust tells you who. SAUTERA tells you whether. It doesn't replace your identity layer; it completes it — Identity Trust plus Infrastructure Trust is the complete trust decision.

The takeaway

Compliance tooling earns its keep only if it proves controls are true continuously, reads real system state, and can act — or at least verify — at the moment things change.

- **Attestation is intent, observation is measurement, enforcement is authority.** Only enforcement changes outcomes; know which one each control actually gets.
- **Score vendors on the six dimensions:** evidence provenance, continuity, device-layer truth, honest unknowns, loop closure, agentic readiness. Demand the live demo where a control breaks.
- **Agentic AI is the forcing function.** Machine actors change infrastructure faster than any sampling interval; bound every agent per action and let "unknown" fail closed.
- **Buy for your binding constraint.** First framework: automation platforms are fine. Live drift, device truth, or an AI workforce: you need continuous, observed evidence — and a boundary that holds between assessments.

Infrastructure trust is not proven once a year. It is decided every time an action fires — and your tooling should be built for that reality.

Take this with you: **download the guide as a PDF** — rubric included, ready to run against your shortlist.

See how SAUTERA scores infrastructure trust in real time — explore the product, or request a demo.