MSP / MSSP PARTNER BRIEF

AI agent security is a new recurring revenue line for your MSP.

Your clients are deploying AI agents now. Nothing in their security stack covers the agent layer. Crawdad is the managed AI-security service you deliver — a new recurring line layered on top of the relationships you've already built.

The opportunity — a managed-service line no one else is offering yet

Recurring, not one-time. Crawdad is a managed service, not a product sale. Layer it into your existing security practice alongside EDR, DLP, and identity. One more line of recurring revenue on every client, with channel economics built for how MSPs actually sell.

First mover defines the category. AI agent security is nascent. The MSP who brings a real answer to their clients now sets the standard for how this layer gets covered. First movers define the category; everyone else follows.

Every client, universal applicability. Every client deploying AI agents needs this coverage, regardless of size, industry, or maturity. Regulated clients in finance, healthcare, and government feel it first and will ask. You bring the answer.

Why now — your clients are exposed right now

AI agents are moving from pilot to production across every industry, and they act with real authority: your client's credentials, files, and network access. When an agent reads a poisoned document or web page, it can follow hidden instructions silently. This is a net-new attack surface that EDR, DLP, firewalls, and identity tools don't see, because those tools were built to govern *humans and endpoints*, not autonomous agents acting on their own.

Regulated clients are already asking who secures their AI agents. The security buyer is looking for an answer, and the MSP who arrives first with a real one wins the room. The gap between "my clients are deploying agents" and "I can secure them" is your opportunity.

Why you — you already own the security relationship

Your clients trust you with their endpoints, networks, and identity. Agent security is the natural extension. Nothing in the typical MSP stack covers the agent layer, and being the one who covers it makes you the leader in every account. Crawdad doesn't replace anything you run today. It covers the surface nothing else sees: additive, no displacement, no rip-and-replace.

What Crawdad does — three things, at the point the agent actually runs

Crawdad installs on your client's machine, at the boundary between every AI agent and the model it calls. One environment variable routes the traffic; from the agent's perspective nothing changes but the address.

It inspects. Every request and response passes through an on-device, multi-layer detection pipeline — pattern and heuristic layers plus a custom-trained transformer classifier — that catches prompt injection, credential leaks, data exfiltration, and indirect (document- and tool-borne) injection. This is the part that matters most and that most of the category doesn't do: many AI-security tools authorize *whether* an agent may make a call, but never look at *what is actually in the traffic*. An attacker who poisons an agent's input produces a request that still looks authorized. Crawdad reads the content, on the device, and catches exactly that.

It enforces intent. An agent can be issued a signed, cryptographically-bound declaration of what it's *for* — a read-only auditor, a specific set of tools and data. Any action that diverges from that declared mandate is caught and can be stopped, per action. This is enforcement of *purpose*, not just detection of bad content: the failure mode where an agent with real authority quietly does something outside its remit.

It contains. Policy is set per device, per client, or fleet-wide, and enforced locally. A governed agent is held to a sanctioned path; the strictest tiers confine it further.

All three run on the client's machine. **Raw client content never leaves their device by default** — telemetry back to your console is metadata only, and customer-governed. When you pitch Crawdad, you can tell your clients their data stays on their machines, because architecturally it does. That's a conversation most of the category cannot have.

How it fits — fleet-scale agent security, every client, one console

- One-command deployment per endpoint.
- **Fleet Console** to manage AI security across your entire client base from one place.
- Per-client tenancy and isolation.
- Centralized policy distribution and detection monitoring.
- **No device limit** — the Fleet Console is architected to manage any number of endpoints, with bulk operations.
- Local-first: raw client content never leaves their machines by default.
- Metadata-only telemetry, customer-governed.
- Fits alongside EDR, DLP, and identity. No rip-and-replace.

Client's agent (any framework) → Crawdad inspects every request → AI model (any provider). Raw content never reaches the cloud, or you.

New in this release — the enforcement layer

The latest release adds the capabilities that move Crawdad from *detecting bad content* to *enforcing what an agent is allowed to do*:

- **Signed intent enforcement** — issue an agent a signed mandate; divergence from it is caught per action.
- **Observe-first onboarding** — a fresh install starts in Observe mode: it shows you what it *would* act on, blocking nothing, until you explicitly arm enforcement. Your client's first hour is a report they can read, not a wall they fight. This is how a security control should onboard, and it makes fleet rollout across a client base safe by default.
- **Per-process attribution** — every decision names which agent, which process, did what. Stable across agent version updates.
- **Session-aware enforcement** — behavior is tracked across a session, not just a single call, so a slow sequence of individually-innocuous actions is caught when the cumulative picture crosses a line.

Proof — a public, reproducible benchmark

We'd rather you verify than take our word.

- **99.80% detection rate** at a **0.09% false-positive rate**
- **497 attacks** and **1,172 benign samples** — **1,669 total**, 22 categories
- **One** missed attack and **one** false positive, both disclosed by name in the public repo

No other vendor in the category has published a reproducible benchmark. Clone the corpus, run it against your own tool, and compare. Benchmark corpus (CC BY 4.0): github.com/AndrewSispoidis/contemporary-agent-attacks.

Under the hood: a multi-layer detection pipeline built in Rust, with **4,796 automated tests passing across 26 crates**, tamper-evident audit logging, and signed, verifiable releases. The detection figures are engine-capability measurements, stated together and reproducible; we don't quote a latency number publicly until the public latency benchmark is out.

The pitch you can make that no one else can

Raw content never leaves your client's device by default. Detection and enforcement run locally; telemetry to your console is metadata only and customer-governed. For regulated clients whose content *can't* be sent to a cloud service, that isn't a feature, it's the difference between "we can secure your agents" and "we can't." You bring the answer that keeps their data on their machines.

The MSPs who secure AI agents first define the category.

Talk to us about becoming a Crawdad partner. We'll walk you through the deployment model, the channel economics, and how to position agent security to your clients.

contact@getcrawdad.dev · getcrawdad.dev · Crawdad Security, Inc.