SOLUTION BRIEF • COMPETITIVE ANALYSIS

Salt Security vs. Wallarm

Wallarm built its platform around inline filtering nodes and payload inspection. AI agents execute attacks through valid, authenticated API calls that contain no malicious payloads. Salt was built to detect the behavioral sequences that payload inspection cannot see.

CATEGORY	PLATFORMS COMPARED	YEAR
Agentic Security	Salt Security & Wallarm	2026

OVERVIEW

Payload inspection finds what attackers no longer bother to hide

Wallarm deploys NGINX-based filtering nodes inline with API traffic, applying ML models and signature matching to detect and block malicious payloads. That approach is effective for injection attacks, known exploit patterns, and volumetric threats. It was not designed for AI agents executing business logic abuse through individually valid, authenticated API calls that contain no detectable payload.

Salt Security operates entirely out-of-band across the full API fabric. No filtering node in the request path. No latency tradeoff. The Agentic Security Graph correlates LLM activity, MCP server connections, API sequences, and identity behavior to surface attack campaigns that look clean at the payload level but reveal themselves across the behavioral sequence.

0

Filtering nodes required. Salt operates fully out-of-band with no inline deployment.

0

Per-endpoint mitigation tuning required. Salt derives behavioral baselines automatically.

8

Years of production API behavioral analysis. Wallarm: payload and signature-based detection.

PLATFORM APPROACHES

Payload blocking at inspection points vs. behavioral correlation across the fabric.

SALT SECURITY

Out-of-band behavioral correlation

Salt Security operates entirely out-of-band. No filtering node, no inline enforcement, no per-endpoint policy tuning. The Agentic Security Graph maps and correlates LLM connections, MCP servers, APIs, identities, and sensitive data across code, cloud, and runtime — detecting multi-step attacks that no single request inspection point can see.

Eight years of production API behavioral data means Salt's anomaly detection understands what normal agent behavior looks like and surfaces deviations with precision. No Lua scripting. No manual mitigation controls. No latency added to your production traffic.

WALLARM

ML-enhanced WAF and inline filtering

Wallarm deploys NGINX-based filtering nodes inline with API traffic. It applies ML models and signature matching to detect and block attacks at the request level, with endpoint-scoped mitigation controls tuned per logic risk. Their API security capabilities extend traditional WAF protection to the API surface.

The constraint: inline filtering sees individual requests. It cannot reconstruct intent across multi-step sequences. And endpoint-scoped mitigation controls require ongoing security team effort to define and maintain for each business logic risk pattern — creating operational overhead that scales with your API footprint.

THE COVERAGE GAP

Wallarm blocks payloads at filtering nodes. Salt detects behavioral campaigns. AI agents do not send SQL injection strings or known exploit payloads. They send valid, authorized API calls — at scale, at speed, in sequences that reveal intent only when correlated across the full fabric. Wallarm's filtering node sees none of that sequence. Salt's Agentic Security Graph sees all of it.

CAPABILITY COMPARISON

Salt Security vs. Wallarm: side by side

CAPABILITY	SALT SECURITY	COMPETITOR
Unified Agentic Discovery	**STRONG** Discovers APIs, MCP servers, and AI-driven assets across external exposure, cloud, code repositories, and runtime as a unified environment.	**PARTIAL** Wallarm discovers APIs within monitored traffic paths. Assets that bypass the filtering node are not discovered.
Agentic Security Graph	**STRONG** Correlates LLMs, MCP servers, APIs, identities, and sensitive data in one action-layer context — the only platform with this cross-fabric view.	**NOT AVAILABLE** Wallarm has no Agentic Security Graph. LLM, MCP, API, and identity correlation in a unified model is not available.
Salt Code Governance	**STRONG** Governs API and MCP creation in repositories and developer workflows before risky logic ships to production.	**NOT AVAILABLE** Wallarm does not govern API and MCP creation in developer repositories.
Runtime-to-Code Remediation	**STRONG** Feeds runtime findings back into DevOps workflows and AI coding assistants to fix root causes — closing the loop between detection and fix.	**NOT AVAILABLE** Wallarm does not feed runtime findings back into developer workflows or coding assistants.
Identity-Aware Sequence Correlation	**STRONG** Tracks unique agentic identities and multi-step intent across sessions, tools, and services to detect campaigns, not just individual events.	**NOT AVAILABLE** Wallarm applies ML detection at the request level. Multi-step identity-aware sequence correlation across sessions and services is not available.
Behavioral Action-Layer Protection	**STRONG** Detects machine-speed business logic abuse beyond signatures, schemas, or known patterns.	**PARTIAL** Wallarm detects known attack patterns and ML-identified anomalies at the request and payload level. Business logic abuse through valid requests across multiple endpoints is difficult to detect.
Internal & East-West Coverage	**STRONG** Protects internal APIs and downstream service interactions that edge-only and model-only tools miss entirely.	**PARTIAL** Wallarm monitors traffic through its inline filtering nodes. East-west APIs that bypass the node are not covered.
Action-Layer Data Security	**STRONG** Maps sensitive data in motion across APIs, MCP servers, and agent actions across the full fabric.	**PARTIAL** Wallarm detects sensitive data exposure at monitored inline inspection points. Sensitive data in

CAPABILITY	SALT SECURITY	COMPETITOR
		motion across the full API fabric is not mapped.
No Filtering Node Required	**STRONG** Salt delivers full protection without placing any filtering node in the request path — zero latency impact, zero application dependency, zero single point of failure.	**REQUIRES INLINE NODE** Wallarm's NGINX-based filtering node sits in the request path, adding latency and creating an application availability dependency on Wallarm's infrastructure.
No Manual Mitigation-Control Scoping	**STRONG** Salt detects complex abuse without requiring endpoint-scoped mitigation controls built per logic risk. Behavioral baselines are derived automatically.	**REQUIRES MANUAL TUNING** Wallarm requires building and maintaining endpoint-scoped mitigation controls for each business logic risk — a process that scales with API complexity and creates ongoing operational overhead.
Beyond Local Rogue MCP Audits	**STRONG** Salt governs MCP and API risk across code, cloud, external exposure, and runtime — not only inspecting what passes through local filtering infrastructure.	**LIMITED SCOPE** Wallarm's MCP coverage focuses on inspecting traffic at local node level. Rogue MCP servers and shadow integrations that bypass the node are outside scope.

Questions to ask in your evaluation

The most productive evaluation question is not which platform has more features, but which platform covers the layer where your risk is highest. Use these scenarios to drive the conversation.

If you need to detect multi-step behavioral attacks

Ask Wallarm to demonstrate detection of a multi-step business logic attack where every individual request is authenticated, valid, and passes all payload inspection checks. This is how AI agents execute attacks — no malicious payloads, only behavioral sequences. Wallarm's filtering node model has no architectural answer for this detection problem. Salt's Agentic Security Graph was built for it.

If operational overhead from tuning is a concern

Wallarm's endpoint-scoped mitigation controls require security teams to define and maintain per-endpoint policies for each logic risk pattern. As your API footprint grows and agents create new connections, that overhead scales. Ask Wallarm what ongoing maintenance a deployment covering 500 APIs requires. Then compare that to Salt's out-of-band approach with automatic behavioral baselining.

If application availability and latency are priorities

Every inline proxy carries availability risk. When Wallarm's filtering node experiences an issue, that issue affects your application. Salt's out-of-band architecture has zero application dependency — protection does not require a node in your traffic path, and a Salt infrastructure issue never affects application performance.

If east-west and internal API coverage is needed

Wallarm's filtering nodes protect defined entry points. East-west microservice traffic, internal APIs that agents call after initial authentication, and services that never expose a public entry point are outside Wallarm's coverage model. Salt covers these paths out-of-band without requiring filtering nodes in every internal network path.

SUMMARY

Key differentiators

SALT SECURITY IS THE STRONGER CHOICE WHEN…

- Out-of-band operation is required — no inline filtering node in the request path
- Multi-step behavioral attack detection beyond payload inspection is needed
- Agentic Security Graph coverage across LLM, MCP, and API layers is required
- Eliminating endpoint-scoped mitigation tuning overhead is a priority
- East-west and internal API coverage beyond filtering node placement is needed
- Zero latency impact to production traffic is a requirement
- MCP governance beyond local traffic inspection is required

WALLARM MAY BE CONSIDERED WHEN…

Wallarm's inline filtering model may be considered when:

- Traditional WAF-style inline blocking is required at a defined perimeter and all relevant API traffic routes through a manageable set of filtering nodes
- Payload-based attack detection and blocking is the primary requirement with no need for behavioral sequence analysis or east-west coverage

For organizations protecting AI agent environments with east-west traffic and behavioral attack patterns, Wallarm's inline model addresses a different problem.

THE COMBINED PICTURE

Payload blocking and behavioral correlation address different attack categories. Wallarm is effective at catching known exploit patterns at inline inspection points. Salt detects the behavioral campaigns — composed of valid, authenticated requests — that AI agents are specifically designed to execute. In agentic environments, the attack surface has shifted from malicious payloads to malicious sequences. Only out-of-band behavioral correlation can see those sequences whole.

SEE THE ACTION LAYER FOR YOURSELF

Salt Surface can show you the full external exposure of your agentic infrastructure with zero deployment, and Salt Collect can demonstrate behavioral patterns in your API traffic that no filtering node inspection model can surface. **Request a demo at salt.security**