SALT

SOLUTION BRIEF • COMPETITIVE ANALYSIS

Salt Security vs. Cequence

Cequence grew from bot detection into API security, bringing its inline enforcement architecture with it. Salt was built for behavioral correlation across the full agentic fabric.

CATEGORY	PLATFORMS COMPARED	YEAR
Agentic Security	Salt Security & Cequence	2026

OVERVIEW

Bot detection and agentic security are fundamentally different problems

Cequence's core product places a Defender reverse proxy in the path of API traffic. It inspects requests inline, enforces policies at the control point, and blocks malicious traffic before it reaches your application. That model excels at detecting high-volume, repeating bot patterns. It was not designed for AI agents executing low-frequency, multi-step business logic abuse across distributed API sequences.

Salt Security operates entirely out-of-band. No proxy in the request path. No data leaving your environment. The Agentic Security Graph continuously maps and correlates LLM connections, MCP servers, APIs, identities, and sensitive data — detecting attack campaigns that no single inline inspection point ever observes whole.

0

Milliseconds of latency added. Salt operates fully out-of-band with no reverse proxy required.

3

Layers covered: LLM, MCP, and API fabric. Cequence covers the API layer only.

0

eBPF sensor deployments required. Salt needs no privileged infrastructure agents.

PLATFORM APPROACHES

Inline enforcement vs. out-of-band behavioral correlation.

SALT SECURITY

Behavioral correlation across the full agentic fabric

Salt Security operates entirely out-of-band. No proxy in the request path, no sensors requiring elevated privileges, no policy tuning scripts. The Agentic Security Graph continuously maps and correlates LLM connections, MCP servers, APIs, identities, and sensitive data — detecting multi-step attack sequences that no inline tool ever sees as a whole.

Salt Surface discovers external exposure. Salt Collect monitors live traffic across 70+ technologies. Salt Code governs repositories pre-production. The result is full agentic coverage without adding a single millisecond to your transaction latency.

CEQUENCE

Inline bot detection and API protection

Cequence deploys a Defender reverse proxy in the path of API traffic. It applies behavioral fingerprinting and policy enforcement at the request level, with endpoint-scoped mitigation controls configured per logic risk. Their AI Gateway extends this model to MCP — adding another proxy layer for agent interactions.

The constraint: Cequence's coverage is limited to what flows through its control points. Internal east-west APIs, behavioral sequences that span multiple sessions, and agent actions that unfold across days are invisible to a model built around individual request inspection.

THE COVERAGE GAP

Cequence enforces at the control point. Salt sees the full picture. AI agents execute attacks through individually valid, authenticated requests that traverse Cequence's proxy without triggering any rule. Salt's Agentic Security Graph correlates the full sequence — across LLM requests, MCP tool calls, API executions, and downstream service interactions — to surface what no per-request inspection model can detect.

CAPABILITY COMPARISON

Salt Security vs. Cequence: 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** Cequence discovers APIs and MCP servers within its monitored traffic paths. Assets that never reach its Defender proxy may not be 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** Cequence has no Agentic Security Graph equivalent. 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** Cequence 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** Cequence does not feed runtime findings back into developer workflows.
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** Cequence applies behavioral fingerprinting 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** Cequence detects anomalies using behavioral fingerprints and ML models. Business logic abuse composed of valid requests across multiple endpoints is difficult to detect inline.
Internal & East-West Coverage	**STRONG** Protects internal APIs and downstream service interactions that edge-only and model-only tools miss entirely.	**PARTIAL** Cequence monitors traffic through its proxy points. East-west internal APIs that bypass the Defender proxy 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** Cequence tracks data at monitored proxy inspection points. Sensitive data in motion across the full API fabric is not mapped.

CAPABILITY	SALT SECURITY	COMPETITOR
No Reverse Proxy Required	**STRONG** Salt delivers full agentic protection without placing any proxy in the request path — zero deployment friction, zero latency impact, zero single point of failure.	**REQUIRES PROXY** Cequence's Defender proxy sits in the request path, adding latency per transaction and creating a dependency on proxy availability for protection.
No Privileged Sensor Requirement	**STRONG** Salt requires no privileged eBPF sensor deployments to broaden visibility beyond inline control points.	**REQUIRES SENSORS** Cequence uses privileged eBPF sensor deployments to extend visibility beyond what the Defender proxy alone can see.
No Fingerprint Scripting or Mitigation Tuning	**STRONG** Salt detects complex behavioral abuse without requiring teams to write and maintain endpoint-scoped mitigation controls per logic risk.	**REQUIRES TUNING** Cequence uses Lua-based fingerprint customization and endpoint-scoped mitigation-policy tuning, creating ongoing operational overhead for security teams.

Questions to ask in your evaluation

The most productive 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 are detecting multi-step behavioral attacks

Bot fingerprints detect individual requests. Salt detects the campaign behind them. Ask Cequence to demonstrate detection of a multi-step business logic attack where each request is individually valid and the attack pattern only becomes visible across a sequence of 20 or more API calls spread over hours. This is how AI agents execute attacks. This is what Salt detects and what Cequence's per-request model cannot.

If latency and application risk are concerns

Every organization that has deployed an inline proxy has experienced a moment where that proxy caused application downtime or performance degradation. Salt's out-of-band architecture eliminates that risk entirely. Zero latency added. Zero application dependencies on Salt's availability.

If you need east-west and internal API coverage

Cequence's Defender proxy is positioned at defined ingress points. East-west traffic between microservices, internal service calls triggered by agent actions, and API interactions that bypass the proxy perimeter are invisible to Cequence. Salt covers these paths without requiring additional deployment.

If your team is managing tuning overhead

Cequence's endpoint-scoped mitigation controls require security teams to write and maintain per-endpoint rules for each business logic risk pattern. Salt's behavioral models derive baselines automatically from runtime data. Ask both teams to walk through what ongoing operational overhead their platform requires after initial deployment.

Key differentiators

SALT SECURITY IS THE STRONGER CHOICE WHEN…

- Zero-latency, out-of-band operation is required — no proxy in the request path
- Multi-step behavioral attack detection across sessions and services is needed
- Agentic Security Graph coverage across LLM, MCP, and API layers is required
- East-west and internal API coverage without inline deployment is needed
- Eliminating ongoing fingerprint tuning and mitigation-control overhead is a priority
- No privileged eBPF sensor deployments can be accommodated
- Runtime-to-code remediation loop is a requirement

CEQUENCE MAY BE CONSIDERED WHEN…

Cequence's inline model may be considered when:

- High-volume bot detection and blocking at a defined perimeter is the primary use case and inline latency is acceptable
- All API traffic routes through a manageable set of controlled ingress points with no east-west or agent-driven traffic to protect

For organizations protecting AI agent environments, the multi-step and east-west attack surface that Cequence cannot see is exactly where agentic risk is realized.

THE COMBINED PICTURE

Bot detection and agentic security solve different problems. Cequence is effective at blocking known volumetric attack patterns at inline control points. Salt detects the behavioral sequences — composed of individually valid, authenticated requests — that AI agents execute across distributed API infrastructure. In agentic environments, those sequences are the primary attack surface, and inline inspection cannot reconstruct them.

SEE THE ACTION LAYER FOR YOURSELF

Salt Surface can scan your environment with zero deployment and show you the APIs and MCP servers your agents are connecting to. Then run Salt Collect to see what behavioral patterns those agents are generating — without placing a single proxy in your traffic path.
Request a demo at salt.security