

A SASE Vendor Turns to AppSentinels to Secure Its Own APIs



How a SASE vendor secured its own APIs at scale, without slowing releases or breaking its air-gapped boundary.

EXECUTIVE SUMMARY

A leading SASE/SSE vendor needed to protect its own APIs at the same standard it sells to customers, without slowing down releases or breaking its air-gapped production boundary. AppSentinels met both needs on one platform, replacing an incumbent API security product that could not: SaaS in UAT for speed, fully air-gapped on-premises in production for control, with the same policies and detection logic across both.



The Challenge: Speed and isolation, at odds with security

Releases outran manual review, and an air-gapped production estate ruled out most cloud-only tools, leaving the riskiest flaws in business logic unchecked.

01 Fast-Moving APIs

New services and endpoints shipped continuously, so manual review and periodic testing were always looking at a version that had already changed.

02 Air-Gapped Production

Production could not send traffic, payloads, or telemetry to a vendor cloud, which ruled out most API security products by design.

03 Business Logic Risk

The risks that mattered most were authorization and workflow errors, such as one tenant reaching another tenant's object or a step skipped in a multi-step flow. Tools that check schemas and signatures do not see these.

04 Critical Path Limits

Anything inline on production traffic added latency and a failure mode, so security could not come at the cost of availability.

“Most API security products asked us to choose between speed and our data boundary. AppSentinels ran as SaaS where we wanted to move fast and air-gapped where the data had to stay: same platform, same policies, no compromise on either side.”

”

— Head of Security Engineering, Global SASE/SSE Vendor

The Solution

Discovery & Posture

Continuous discovery of every API in UAT and production, including shadow and undocumented endpoints, with sensitive data flows mapped to the APIs that carry them.

Automated Testing

Tests are generated from observed traffic and real business workflows rather than a static list, and re-run as APIs change; multi-step and authorization abuse is exercised the way an attacker would.

Runtime Monitoring

Production traffic is monitored out of band on a separate plane. The platform sees every call and flags abuse in real time without sitting in the request path or adding latency.

Dual Deployment

SaaS in UAT for speed and fully air-gapped on-premises in production, with all data staying inside the customer's boundary and the same detection logic and policies in both.

5,000+

APIs discovered and under continuous coverage

15 billion+

API calls monitored every month

Zero

bytes of production data leaving the environment

Business Outcomes

Full API coverage

5,000+ APIs discovered and under continuous coverage across UAT and production, including endpoints that were not in any existing inventory.

Runtime scale with zero exposure

15 billion+ API calls monitored per month at runtime, out of band, with no latency added to production traffic and no customer or tenant data leaving the environment.

Air-gapped production, fully covered

Production ran under a fully air-gapped deployment with complete coverage and no data leaving the customer's environment.

Faster feedback loops

Testing moved from periodic and manual to continuous and automated, so findings arrive while code is still moving rather than at the next review cycle.

Deeper risk coverage

Authorization and workflow flaws surfaced that schema and signature checks do not reach, the class of issue that only shows up when calls are chained in sequence.

About AppSentinels

AppSentinels is the Business Logic Security platform for modern applications. Built around the Business Logic Graph, it delivers continuous discovery, automated stateful red-teaming, and runtime protection across AI agents, MCP servers, APIs, and the workflows that connect them. First-generation API security tools inventoried endpoints; AppSentinels secures the logic behind them, the layer where breaches happen.