vicarius

VS

Balbix

From reporting risk to real protection

Balbix delivers risk scores and reports but leaves the real work, patching, up to other tools. Vicarius closes the gap with built-in detection, prioritization, and automated remediation, eliminating reliance on fragile integrations and reducing risk faster, with broader coverage and lower operational overhead.

Vicarius key differentiators

Detection to remediation loop

Balbix stops at reporting. Vicarius detects, prioritizes, and patches vulnerabilities in real time, closing the loop and eliminating the gap between knowing and fixing.

Built-in native patching

No reliance on external tools. Vicarius delivers native patch management for OS and 10K+ apps, while Balbix depends on third-party integrations for basic patching.

Zero-Day and exploit awareness

Vicarius integrates real-time EPSS, KEV and exploit intelligence for proactive risk reduction. Balbix offers no native zero-day patching or exploit-aware prioritization.

Full cross-platform coverage

Vicarius supports Windows, macOS, Linux, legacy systems natively. Balbix's macOS and legacy coverage is limited and requires integrations.

Fast, integration-free deployment

Vicarius is cloud-native and operational in hours. Balbix's heavy reliance on integrations increases setup complexity, slows time-to-value, and inflates TCO.

Complete MSSP-ready multi-tenancy

Vicarius offers true multi-tenancy for MSSPs managing multiple customers at scale. Balbix lacks built-in multi-tenancy, relying on complex external workflows.

How Vicarius compares to Balbix

Capabilities	vicarius	Balbix
Remediation built-in	Native, no-code remediation with automated patching, exploit-aware workflows, and immediate execution.	Focuses on risk reports; "remediation" means creating tickets for other tools to handle. Vulnerabilities remain unpatched unless external systems take over.
Virtual patching	Enables virtual patching for zero-days and unsupported software	No virtual patching, if a vendor hasn't issued a fix, you stay exposed.
Cross-platform support	Full patching for Windows, Linux, macOS, and over 10,000 third-party applications	MacOS only through integrations; legacy coverage is thin. Heavy dependence on other vendors to fill basic platform gaps.
Agent-based & agentless modes	Offers both agent-based and agentless scanning for flexible deployment across asset types	Limited flexibility; requires integrations for some workloads, adding fragility and complexity.
Real-time risk prioritization and validation	Prioritizes vulnerabilities using contextual, risk-based analysis that goes beyond CVSS, and validates exploitability in live environments using telemetry, not simulated exploits.	Scores risk but stops short, no native exploit intel, so high scores don't always mean fast fixes.
Multi-tenancy for MSSPs	MSSP-friendly architecture with tenant isolation, delegated roles, and centralized control	No true multi-tenancy; each customer feels like a separate deployment project.
Workflow Automation	Unified platform for detection, prioritization, and remediation, minimizing tool sprawl	"Automation" is just integration handoffs, patching is never in-platform, slowing down actual risk reduction.
Ease of deployment	Modern, cloud-native UI with fast onboarding and centralized console	Heavy reliance on integrations bloats setup time and introduces more moving parts to manage.
Coverage depth	Covers vulnerabilities across system, apps, network, and even patchless or legacy systems	Limited built-in coverage; every third-party app requires an integration or separate tool to patch.
Customer satisfaction	G2: 4.9 / Gartner: 4.8 praised for ease of use, performance, and support	G2: 4.8 / Gartner: 4.7 – customers cite lack of CIS benchmarks, no API for workflow integration, missing release notes, and weak hardening capabilities.