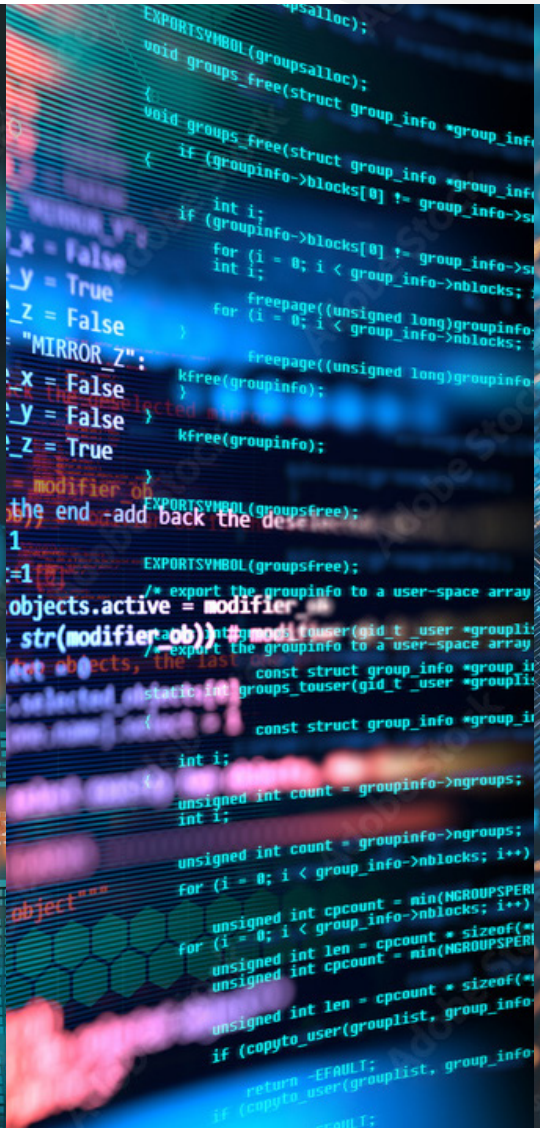
How OpenText addresses current and future application security challenges



OpenText is uniquely positioned to help enterprises meet application security challenges head on with a modern, integrated, and AI-augmented platform. Let’s look at some examples, implementation patterns, and proof points for field and marketing use.

Biggest gains

- Boost developer productivity with AI-assisted triage and remediation
- Reduce AppSec tool sprawl and associated management costs
- Enhance compliance with automated SBOMs and audit-ready workflows
- Improve security across APIs, AI, and the software supply chain
- Future-proof systems for post-quantum and GenAI risks

Tackling the AI explosion in development

Challenge: With more than one-third of enterprises deploying GenAI in production,[1] the attack surface is rapidly expanding—introducing new vulnerability classes across source code, AI models, data, and orchestration pipelines.

Solution

AI-augmented SAST for intelligent triage: OpenText™ Application Security Aviator™ leverages LLM-based auditing to auto-classify findings, add developer-friendly rationale, and recommend secure code fixes—all integrated directly into CI/CD.

Real-time rules for GenAI technologies: Continuously updated SAST rules identify issues like unsafe LLM/agent patterns (AutoGen, OpenAI/ lib misuse), prompt injection risks, and poor trust boundaries.

DAST for LLM-enabled apps: Authenticated scans drive parameterized testing of AI-backed APIs and web flows. Upcoming support includes macro generation for frictionless scanning of GenAI user journeys.

Use cases

LLM output injection: Flags unsanitized use of AI-generated content in commands or templates (e.g., SSTI risks, SQL injection).

Agentic execution risks: Detects tool use patterns without proper auth layering, enabling DevSecOps teams to enforce scoped credentialing.

AI libraries: Highlights risky default settings like verbose debug logs or unrestricted file/network access.

Workflow

1. Activate Aviator in CI pipelines for automated triage and remediation.
2. Enable AI-specific rulesets for your languages (Python, JS, Java, etc.).
3. Record workflows like AI-driven content creation or code generation; run DAST scans against staging environments.

Explore the solution

1 Forrester, *The State Of Application Security, 2025: Yes, AI Just Made It Harder To Do This Right*, 2025

Use case 2

Securing the software supply chain

Challenge: Modern supply chain attacks increasingly target open-source components, AI model artifacts, build systems, and registry integrity.

Solution

SCA across the pipeline: Detect vulnerable OSS dependencies with upgrade paths, enforce license policy, and block critical CVEs pre-merge.

Core Open Source Select: Offers developers package intake health indicators—CVSS age, popularity, maintenance velocity—to prevent risky picks.

Ecosystem risk visibility: Through Debricked integration, security teams gain dashboards and reports to drive allow/deny governance.

Controls and legal governance: Repository firewalls and legal pack rules automate policy enforcement for CVEs and licensing.

Use cases

Typosquatting Detection: Fails PRs on suspicious low-reputation packages, auto-suggesting safer alternatives.

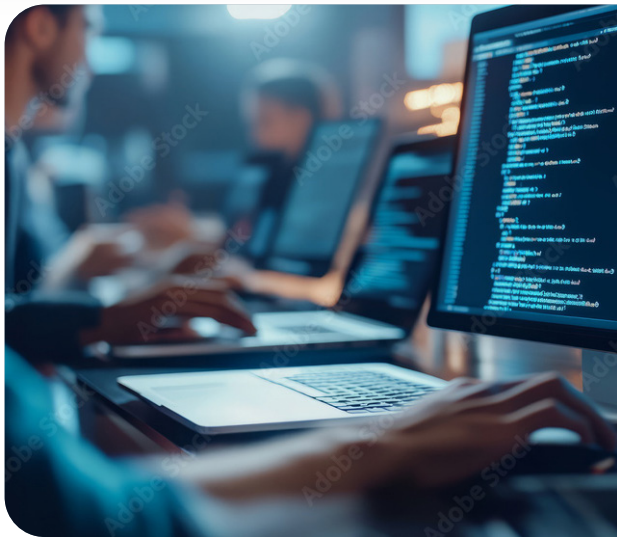
AI model integrity: Blocks unverified model downloads unless signed and checksummed.

License policy enforcement: Prevents builds that violate enterprise license constraints like GPL/CDDL, with remediation options surfaced to developers.

Workflow

1. Define allow/deny rules and minimum health scores in Core Open Source Select.
2. Integrate SCA policies with CI for proactive dependency enforcement.
3. Configure perimeter defense via repo firewalls, sync exceptions in AppSec platform.

Explore the solution



Use case 3

API security as a core priority

Challenge: API-first development exposes teams to complex risks—authorization flaws, over-permissioned third-party APIs, schema drift, and abuse vectors.

Solution

Full-spectrum API testing: SAST and IAST scan source-level logic, while DAST tests REST, GraphQL, and gRPC APIs with schema-based and workflow-driven coverage.

Aligned to OWASP API Top 10: Policies and reports focus efforts on the highest risk areas with mapped mitigation tracking.

Auth-aware scanning: Support for OAuth2, OIDC, mTLS, and complex login flows ensures deep testing of real-world access controls.

Use cases

BOLA/BOPLA: SAST flags missing access checks, while DAST ensures users can't access others' resources by ID manipulation.

Schema drift: DAST discovers undocumented or shadow APIs, prompting inventory updates and improved rate limiting.

GraphQL exposure: Identifies unbounded resolvers and sets query complexity limits to prevent abuse.

Workflow

1. Import OpenAPI and GraphQL schemas; add traffic-inferred endpoints.
2. Record login flows and key API use cases.
3. Tie API scans to release gates and deploy-blocking policies.

Explore the solution

Use case 4

DevSecOps at scale with a developer-first focus

Challenge: Developers influence AppSec tooling decisions and demand embedded, seamless experiences inside IDEs, SCMs, and CI pipelines.

Solution

Native integrations: Plugins for IntelliJ, VS Code, Eclipse; connectors for GitHub, GitLab, Azure DevOps, Jenkins, and ServiceNow.

Orchestration via ScanCentral: Enables concurrent SAST/DAST scans across massive monorepos and microservice fleets.

In-context learning: Secure Code Warrior links findings to short, interactive labs that upskill developers where it matters.

Use cases

Pull request guardrails: On every PR, SAST runs + Aviator auditing push only verified issues to Jira.

Elastic scan scaling: Hundreds of concurrent SAST jobs dispatched during nightly builds avoid CI latency.

Targeted remediation training: Persistent React-based XSS leads to training activation; recurrence rate drops over three sprints.

Workflow

1. Start with a golden repo and standard CI template.
2. Set thresholds and SLAs in SSC policy.
3. Enable Aviator for automated triage and feedback.

Explore the solution

Use case 5

Managing tool sprawl and operational complexity

Challenge: AppSec leaders deal with fragmented tools, inconsistent metrics, and growing pressure for centralized ASPM views and governance.

Solution

Single platform governance: SSC offers policy control, scan orchestration, and compliance alignment from one hub.

Modular coverage: Combine SAST, DAST, IAST, SCA, mobile, and monitoring under one roof—deployable across SaaS, private cloud, and on-prem.

Data portability: REST APIs and export options enable seamless integration with SIEMs, BI tools, and GRC platforms.

Use cases

Unified policy plane: SAST, DAST, and SCA share one rule set—enabling executive dashboards and compliance rollups.

Automated governance: Exceptions created in SSC populate in ServiceNow with tracking and expiry for full audit visibility.

Workflow

1. Map applications to SSC business services.
2. Standardize CI/CD templates for release tagging.
3. Export metrics to enterprise GRC and BI dashboards.

Explore the solution

Use case 6

Responding to regulatory and compliance pressure

Challenge: Security teams must now meet SBOM mandates, incident disclosure timelines, and industry-specific attestation requirements, often under tight audit deadlines.

Solution

Compliance-mapped policies: Align scanning and reporting to frameworks like NIST, FedRAMP, and STIG with minimal overhead.

SBOM automation: Generate and associate SBOMs and evidence (timestamps, remediation steps) to builds and release tickets.

Deployment flexibility: Support for secure SaaS and hardened private deployments, meeting needs of federal and high-regulation sectors.

Use cases

STIG-ready reports: Use default policies to produce DAST results tailored to STIG, easing federal compliance.

Disclosure assurance: Respond quickly to 8K or internal incidents with timestamped CVE impact assessments, patch status, and audit logs.

Workflow

1. Select compliance policies aligned to sectoral standards.
2. Auto-publish SBOMs during builds; link to release records.
3. Use SSC workflows for exception approvals, expiration, and audit trails.

Explore the solution

Use case 7

Elevating detection accuracy and reducing false positives (FPs)

Challenge: Frequent false positives slow developer response and erode trust in AppSec tools.

Solution

AI-assisted triage: Aviator and Audit Assistant use ML to auto-categorize findings, attach rationale, and suggest fixes.

Policy tuning: Lean scan policies, curated filter sets, and tailored rules reduce noise across diverse app portfolios.

Operational guardrails: Auto-suppress low-risk issues; escalate only those likely to be exploitable.

Use cases

Noise reduction: Enabling Aviator cuts PR audit volume by 70 percent, streamlining developer queues.

Precision focus: Updated DAST policies prioritize exploit-relevant issues and remove legacy check clutter.

Workflow

1. Deploy a security-focused baseline policy.
2. Enable Aviator auto-prediction in CI pipelines.
3. Quarterly FP reviews feed custom filters and refinement rules.

Explore the capability

8) Preparing for the post-quantum era

Challenge: The transition to post-quantum cryptography demands crypto agility without disrupting deployment velocity.

Solution

Crypto hygiene SAST: Flags legacy/insecure algorithms, hardcoded keys, improper TLS, and expired cipher suites.

Inventory crypto usage: Integrated scanning inventories crypto artifacts in code, dependencies, and services—essential for phased upgrades.

Use cases

TLS hardening: CI/CD fails builds on use of deprecated cipher suites, developers get auto-suggested upgrades.

Key management: Detects hardcoded secrets and missing rotation policies, driving KMS adoption.

Workflow

1. Enable crypto rules to establish a current baseline.
2. Define PQC migration plans in SSC with ownership and timelines.
3. Use SBOM and SCA to track crypto dependencies across third-party packages.

Read the blog

9) Defending against AI-powered and AI-targeted attacks

Challenge: Attackers are weaponizing automation and targeting AI systems directly—via prompt injection, data poisoning, and LLM abuse.

Solution

AI-specific rule coverage: Detect insecure tool/model interactions, unsanitized AI I/O, and overly permissive configurations.

Scenario-driven DAST: Authenticated scans simulate real business flows and negative AI usage cases.

Human-in-the-loop feedback: Aviator delivers enriched findings with remediation suggestions, helping developers build secure AI experiences.

Use cases

Prompt injection: Flags user prompt paths that reach system tools/instructions unfiltered.

Tool exploitation: Highlights LLM-driven tool access lacking authorization gates.

Workflow

1. Inventory AI endpoints and internal tooling interfaces.
2. Model misuse cases and record DAST scenarios against those paths.
3. Enforce merge gates requiring input validation and allowlisting for AI I/O.

Explore the research

10) Securing smart contracts and blockchain applications

Challenge: Smart contracts are immutable and often handle financial value—vulnerabilities are unrecoverable and costly.

Solution

Smart contract SAST: Detects reentrancy, unchecked external calls, and role misconfigurations in Solidity and Web3 code.

DAST for blockchain frontends: Covers dApp UIs and APIs connected to smart contracts.

Unified governance: Web3 and legacy services live in the same SSC portfolio for consistent policy, audit trails, and risk tracking.

Use cases

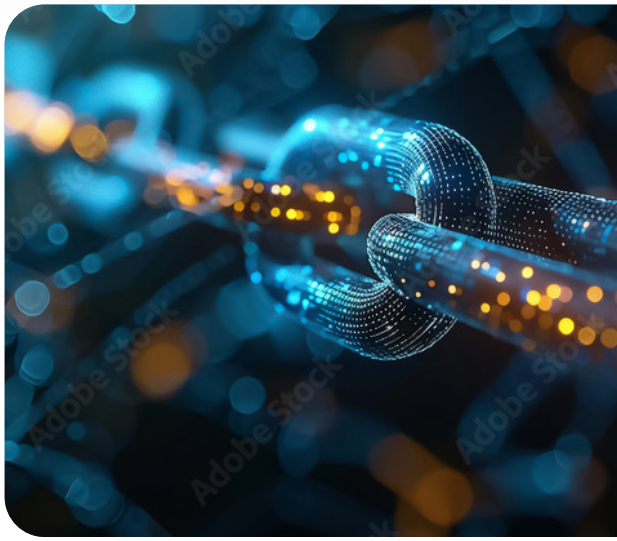
Financial guardrails: Blocks deployment of token contracts missing proper access control.

Admin panel security: Finds stored XSS in NFT management UIs that could target wallet-linked admins.

Workflow

1. Run SAST on smart contract repositories pre-deploy.
2. Test blockchain UIs and admin APIs via DAST.
3. Manage exceptions and approvals in SSC for centralized visibility.

Explore the capability



OpenText Application Security empowers security, DevOps, and engineering teams with a unified, scalable platform built for today's evolving threat landscape. From securing AI-driven applications and APIs to hardening the software supply chain, enabling crypto agility, and unifying AppSec governance, OpenText helps teams detect faster, remediate smarter, and scale securely—without slowing innovation.

opentext™