WHITE PAPER

Guide to Application Security: What to Look For and Why

DevSecOps and Application Security Best Practices

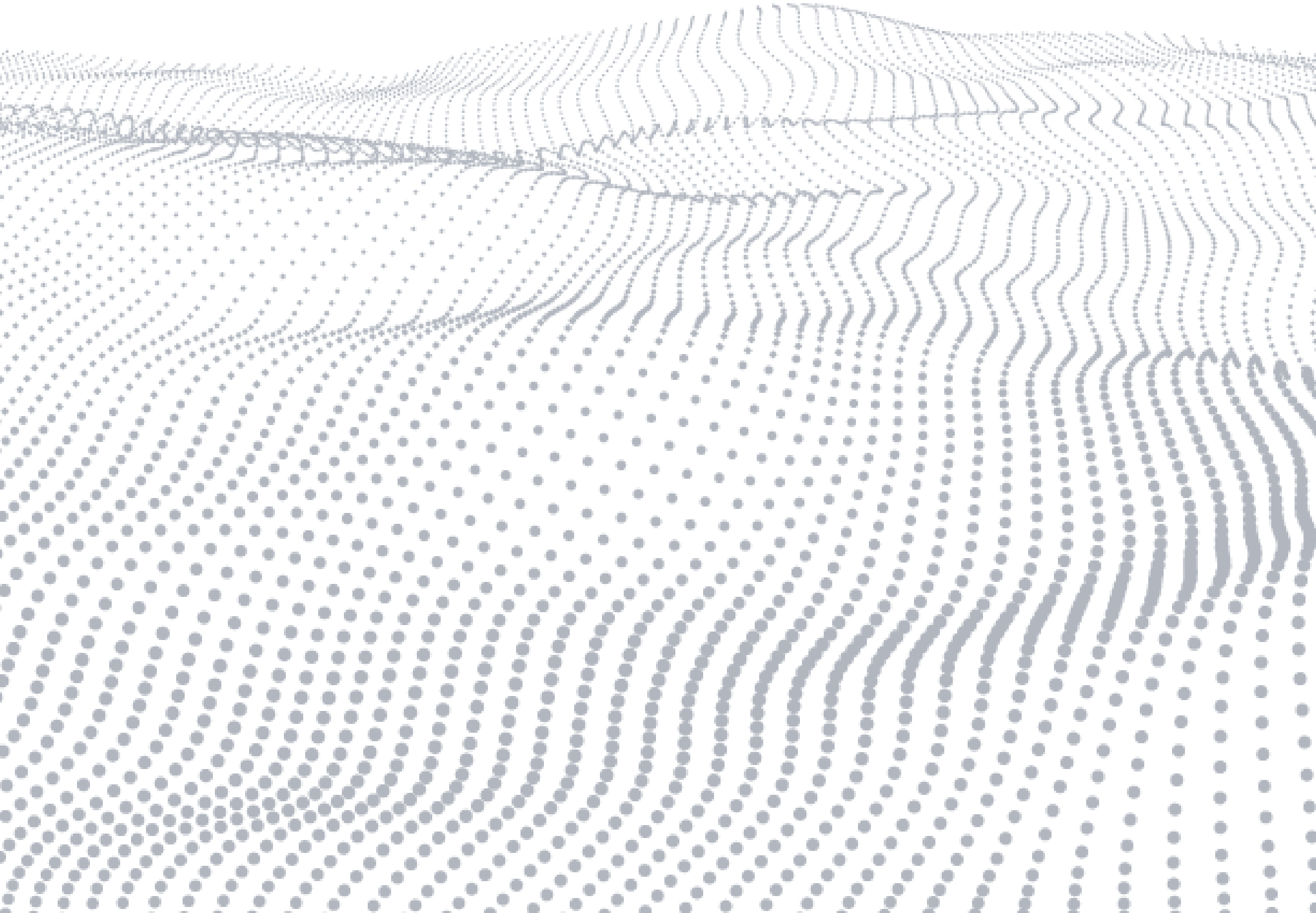


Table of contents

Introduction

If your organization does software development in-house, there are a myriad of development workflows and processes to choose from. Some organizations still implement old-school waterfall development workflows; some are agile shops. In terms of process, some have adopted DevOps, and some integrate security testing into DevOps workflows for DevSecOps.

How and why should you transition from one approach to another? What do you need to change in terms of culture, tools, techniques, and processes? What tools and processes can help you integrate application security into your DevOps workflows and enable seamless testing for developers in your organization? Which tools and processes work in which stages of the software development life cycle (SDLC)? Should you be thinking about containerization? Should you develop code in the cloud? How do you get started? If you're a CISO, how does DevSecOps improve software security and generate tangible savings in lead times to prevent a major exploit? How should you evaluate and select application security tools, and how should you partner with your development teams?

This white paper answers some of these questions and describes best practices for securely accelerating your software velocity.

Waterfall, agile, DevOps, and DevSecOps basics

Many organizations have moved away from the rigid sequential stages of waterfall toward agile development. Agile development gives organizations the flexibility to make changes in any phase, support frequent requirement changes, and perform testing concurrently with software development.

What is DevOps?

DevOps is an ideology that combines cultural philosophies, technical practices, and tools to help development and IT operations teams work collaboratively to build, test, and release software faster and more reliably.

According to The DevOps Handbook, "In the DevOps ideal, developers receive fast, constant feedback on their work, which enables them to quickly and independently implement, integrate, and validate their code, and have the code deployed into the production environment."[1]

What is DevSecOps?

DevSecOps involves integrating security testing into continuous integration (CI) / continuous delivery (CD) workflows. CI/CD is facilitated by automated software development workflows and processes (e.g., in the build, test, and release phases).

Continuous integration (CI) is a software development practice where team members integrate their work frequently. Typically, each person integrates daily, leading to multiple integrations per day. Integrations are verified by an automated build (including test) to detect integration errors as quickly as possible. CI build servers automatically pull in requisite files and dependencies from SCM repositories when developers check their updates into the version control system.

Continuous delivery (CD) is a software development discipline where software is built in such a way that it can be released to production at any time. Continuous deployment (also CD) means that every change goes through the pipeline and automatically gets put into production, resulting in many production deployments every day. Build servers are also used to enable continuous deployment, deploying updates to production when a build succeeds and passes all tests. Following stringent CI/CD workflows is becoming the norm in modern enterprises that develop and deploy software.

Enterprise development requirements

Developers need to be able to write, debug, and test code quickly and easily, before checking it into a central build. They need access to industry-standard integrated development environments (IDEs) that support their style of code development and code integration. Many DevOps tools integrate into the IDE to ensure developers can use them without any friction.

Logically, this trend should also apply to application security testing and AppSec tools. Tools that can integrate into the IDE are preferred by developers, who can then maintain security compliance while they work in their native environments. Better security compliance from developers helps security executives improve operational efficiencies and accelerate release velocities. Improved efficiencies and faster release velocities, in turn, translate into a significant reduction in the total cost of ownership of the application security solution.

Zooming out, we see emerging technologies in container development slowly becoming mainstream. Development teams are adopting container orchestration systems, such as Kubernetes, to automate container and binary cluster deployments and to control and manage the scaling of containerized applications. Large enterprises are moving development to the cloud to offload costs and save resources while benefiting from a faster time to market and the scalability advantages of not having to purchase and maintain servers.

At the end of the day, it's all about increasing software velocity to meet rapidly changing business needs. To do this, organizations need to empower their developers by giving them tools and establishing workflows that make delivering secure, high-quality software faster and more efficient.

Start security testing by shifting left

Many studies have shown that it's faster, easier, and cheaper to find and fix software issues early in the development process, as developers write code, rather than later (e.g., in testing, QA, or production). Organizations often use application security tools, such as static application security testing (SAST) tools, early in development to find and fix both quality issues and security vulnerabilities. Teams often use SAST tools in conjunction with software composition analysis (SCA) tools, which find security vulnerabilities and license issues in open source components.

Static versus dynamic testing

Many organizations use dynamic application security testing (DAST) or penetration testing during QA or just before production. DAST and pen testing dynamically test an application for security vulnerabilities that can't be detected using SAST or SCA tools. Security teams often perform DAST and pen testing periodically on live released applications for security compliance reasons. But running dynamic testing on live applications exposes them to cyber attacks and potential data breaches.

In addition, while DAST and pen testing can identify security vulnerabilities, they can't identify the corresponding lines of code containing the vulnerability—something that SAST tools can do before an application is in production to reduce the attack surface for a potential vulnerability. As a result, critical security issues identified by DAST can be problematic to fix and can take a long time to resolve. With companies now deploying to production multiple times a day (e.g., Netflix), a live security vulnerability lurking in production code can spell disaster, affecting the bottom line instantly. This potential for far-reaching, immediate impact from a single vulnerability is why SAST sees more widespread adoption than DAST.

Shifting left eliminates unnecessary work later

Instead of sifting through and prioritizing long lists of security issues (including false positives) generated by "noisy" DAST tools post-release, security teams should instead work closely with their development teams to eliminate vulnerabilities earlier in the SDLC by using SAST, SCA, and interactive application security testing (IAST) tools. IAST tools dynamically test applications during runtime, typically in the test and QA phases, to identify security vulnerabilities that SAST or SCA tools couldn't find. Unlike DAST tools, which have slower analysis times, IAST tools can integrate seamlessly with build and test automation tools (e.g., Selenium) and quickly generate analysis results that identify specific lines of code where security vulnerabilities reside. As a result, developers can fix identified issues quickly and push their commits to run automated CI/CD workflows.

What to look for in an application security tool

Essential features

Application security (AppSec) tools must have certain basic features to be effective:

- They must be fast, accurate, and comprehensive.
- They must be easy to use and easy to deploy, with support for multiple languages and frameworks.
- They must have powerful automated analysis engines that don't just scratch the surface but dive in deeper to find critical quality and security vulnerabilities that are difficult or impossible to discover through manual code review or penetration testing.
- They must provide support, plugins, and integrations for various tools in the SDLC: industry-standard IDEs, source code and OSS repositories, CI build servers, bug trackers for triaging identified issues, and cloud and container development tools.

Integrations

With the need to stay competitive and achieve a faster go-to-market, organizations are focusing on low-overhead application security testing, which is driving the push to go to cloud. Therefore, application security testing tools must seamlessly integrate with development and DevOps tools used either on-premises or in the cloud. AppSec tools must fit with the tools that developers and DevOps leads are already using or will use in the future. If these tools slow down development, neither development nor DevOps teams will adopt them.

Reporting

AppSec tools must also provide comprehensive and extensible reporting schemes. These should cover not only the tactical aspects for triaging and remediation by security teams but also executive-level dashboards and reports for heads of development and security. For example, they must provide security trend data and compliance information in relation to industry taxonomies (e.g., OWASP Top 10, CWE/SANS Top 25). AppSec tools must be adaptable to organizations' needs and help them reduce compliance overhead. Financial services organizations, for example, need reports in PCI formats, whereas the medical industry relies on compliance reporting with HIPAA.

Team-specific needs

Security leads strive for continuous visibility into the most critical vulnerabilities that could adversely affect their organizations. Development leads, on the other hand, rely on AppSec tools to provide (1) detailed contextual remediation guidance for identified issues and (2) real-time security training so developers don't need to be security experts to fix vulnerabilities and check in cleaner code.

What to look for in an AppSec software-as-a-service (SaaS) platform

Each role in an organization looks for certain factors when evaluating a new platform:

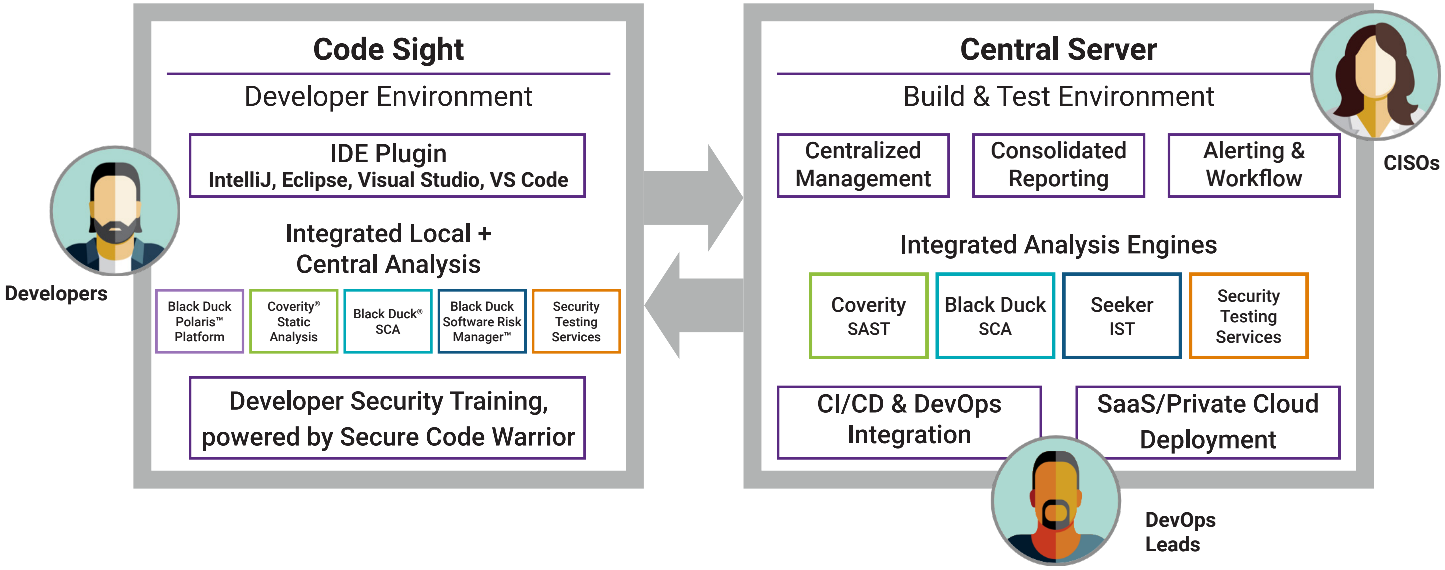
Development leads look for tools that fit seamlessly into their existing workflows, that are accurate, and that don't generate a lot of false positives. They want to avoid tool-related developer frustration and wasted time.

DevOps leads look for tools that integrate with their existing build systems and development and test tools and processes.

CISOs, allied executives, and security leads look for robust, comprehensive reporting capabilities that accurately prioritize critical security vulnerabilities for remediation. They look for time-saving automated tools that can generate executive-level reports that help them maintain compliance with regulatory and security standards.

Black Duck Polaris® Platform is a scalable solution that can address many requirements for each of these roles out of the box.

Figure 1. Black Duck Software Integrity Platform



Code Sight
Developer Environment
IDE Plugin
IntelliJ, Eclipse, Visual Studio, VS Code
Integrated Local + Central Analysis
Black Duck Polaris™ Platform
Coverity® Static Analysis
Black Duck® SCA
Black Duck Software Risk Manager™
Security Testing Services
Developer Security Training, powered by Secure Code Warrior
Developers
Central Server
Build & Test Environment
Centralized Management
Consolidated Reporting
Alerting & Workflow
CISOs
Integrated Analysis Engines
Coverity SAST
Black Duck SCA
Seeker IST
Security Testing Services
CI/CD & DevOps Integration
SaaS/Private Cloud Deployment
DevOps Leads

Developers want ease of use, speed, and accuracy

Developers need AppSec tools that help them do their job without getting in their way or creating extra work. Developers typically use SAST IDE plugins to identify and fix issues before checking in their code for central build analysis. However, many SAST IDE plugins are "lightweight" and can find only a subset of vulnerabilities. Using them can cause builds to break because of security compliance or quality gates.

With Black Duck's Code Sight™ IDE Plug-in plugin for the Polaris platform, developers don't have to sacrifice accuracy for speed. Code Sight is a productivity tool that helps developers identify and triage software defects early in the SDLC, so they don't propagate the same errors further downstream, where it's much more expensive and time-consuming to remediate.

Fast incremental analysis

With Code Sight, developers can get Coverity® SAST incremental analysis results in seconds within their IDE. Code Sight does incremental analysis automatically in the background every time developers open or save a file. For consistent, accurate results, the plugin uses the same Coverity analysis engine used for full baseline central analysis. And Polaris automatically synchronizes incremental analysis results with central analysis scans, so developers can focus on coding without having to invoke scans manually or interrupt their workflow.

Security training and remediation advice

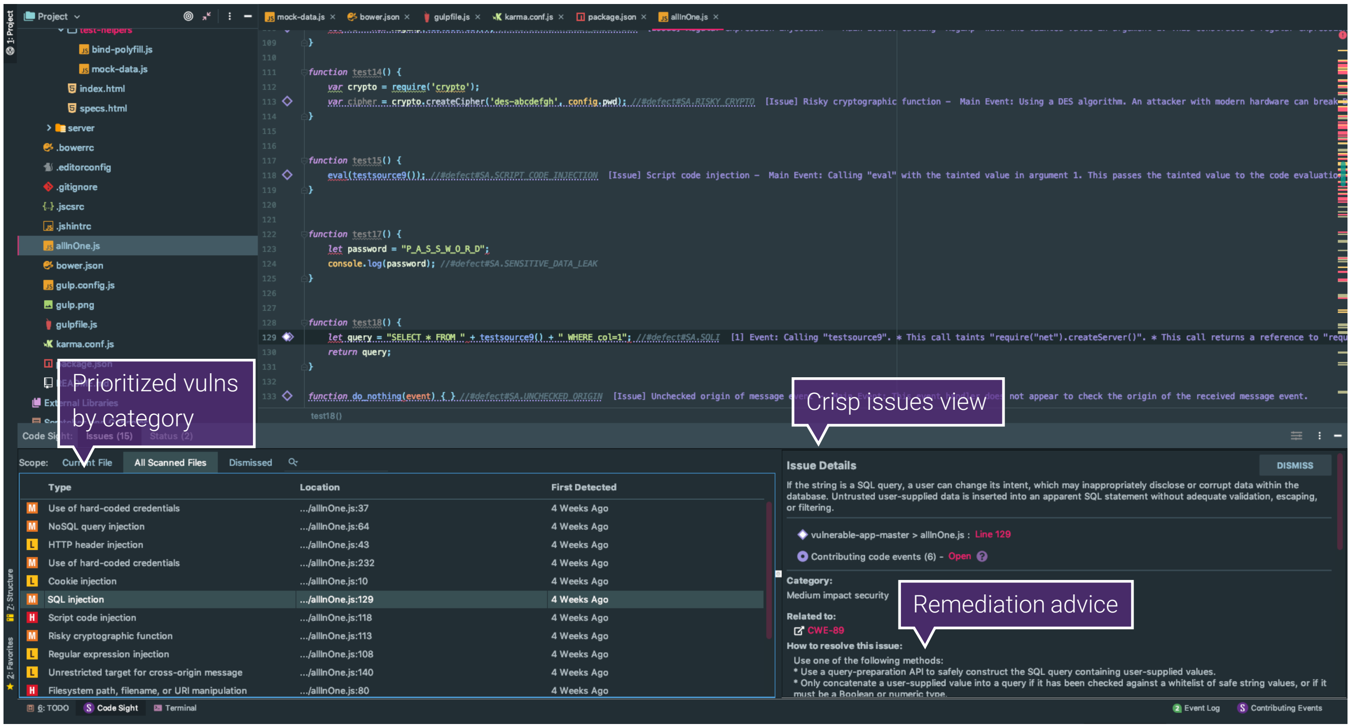
Developers using Code Sight get “in-the-moment” security training and remediation advice on how to fix issues as they are coding, without leaving their IDE. Remediation advice includes detailed issue descriptions, prioritized vulnerabilities by severity, links to CWE information, and dataflows that help them debug their issues and prevent security and quality issues from entering the main codebase.

Code Sight also provides developers with links to eLearning courses related to the CWEs associated with specific issues in their code. When developers receive contextual security training, they don’t have to be security experts to fix new vulnerabilities as they arise.

Consistent user experience

Code Sight’s modern developer interface is consistent across standard industry IDEs (e.g., Visual Studio, Eclipse, IntelliJ, VS Code) and all Black Duck products supported on the Polaris platform. Consequently, developers don’t have to learn a new UI each time they use a new security tool.

Figure 2. Code Sight screenshot examples (in IntelliJ)



```
}

function test14() {
    var crypto = require('crypto');
    var cipher = crypto.createCipher('des-abcdefgh', config.pwd); //#defect#SA.RISKY_CRYPTO  [Issue] Risky cryptographic function –  Main Event: Using a DES algorithm. An attacker with modern hardware can break
}


function test15() {
    eval(testsource9()); //#defect#SA.SCRIPT_CODE_INJECTION  [Issue] Script code injection –  Main Event: Calling "eval" with the tainted value in argument 1. This passes the tainted value to the code evaluatio
}


function test17() {
    let password = "P_A_S_S_W_O_R_D";
    console.log(password); //#defect#SA.SENSITIVE_DATA_LEAK
}


function test18() {
    let query = "SELECT * FROM " + testsource9() + " WHERE col=1"; //#defect#SA.SQLI  [1] Event: Calling "testsource9". * This call taints "require("net").createServer()". * This call returns a reference to "requ
    return query;
}

function do_nothing(event) { } //#defect#SA.UNCHECKED_ORIGIN  [Issue] Unchecked origin of message event – Main Event: This event handler does not appear to check the origin of the received message event.
test18()
```

	Type	Location	First Detected
M	Use of hard-coded credentials	.../allInOne.js:37	4 Weeks Ago
M	NoSQL query injection	.../allInOne.js:64	4 Weeks Ago
L	HTTP header injection	.../allInOne.js:43	4 Weeks Ago
M	Use of hard-coded credentials	.../allInOne.js:232	4 Weeks Ago
L	Cookie injection	.../allInOne.js:10	4 Weeks Ago
M	SQL injection	.../allInOne.js:129	4 Weeks Ago
H	Script code injection	.../allInOne.js:118	4 Weeks Ago
M	Risky cryptographic function	.../allInOne.js:113	4 Weeks Ago
L	Regular expression injection	.../allInOne.js:108	4 Weeks Ago
L	Unrestricted target for cross-origin message	.../allInOne.js:140	4 Weeks Ago
H	Filesystem path, filename, or URI manipulation	.../allInOne.js:80	4 Weeks Ago

Issue Details

If the string is a SQL query, a user can change its intent, which may inappropriately disclose or corrupt data within the database. Untrusted user-supplied data is inserted into an apparent SQL statement without adequate validation, escaping, or filtering.

vulnerable-app-master > allInOne.js : Line 129

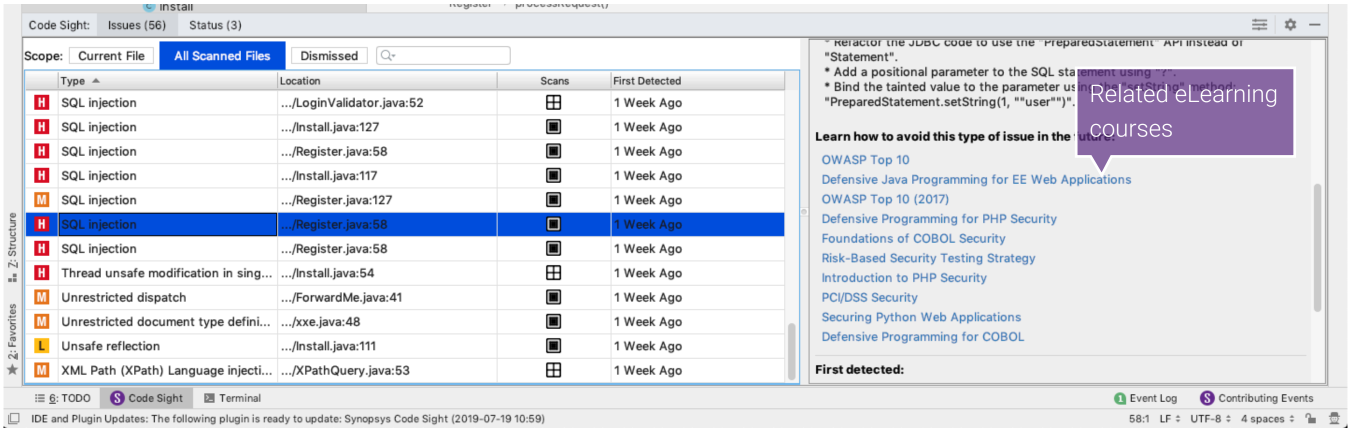
Contributing code events (6) - Open ?

Category:
Medium impact security

Related to:
CWE-89

How to resolve this issue:

Use one of the following methods:
* Use a query-preparation API to safely construct the SQL query containing user-supplied values.
* Only concatenate a user-supplied value into a query if it has been checked against a whitelist of safe string values, or if it must be a Boolean or numeric type.



Install
Register › processRequest()
Code Sight: Issues (56) Status (3)

Scope: Current File | **All Scanned Files** | Dismissed

	Type	Location	Scans	First Detected
H	SQL injection	.../LoginValidator.java:52	⊞	1 Week Ago
H	SQL injection	.../Install.java:127	▣	1 Week Ago
H	SQL injection	.../Register.java:58	▣	1 Week Ago
H	SQL injection	.../Install.java:117	▣	1 Week Ago
M	SQL injection	.../Register.java:127	▣	1 Week Ago
H	SQL injection	.../Register.java:58	▣	1 Week Ago
H	SQL injection	.../Register.java:58	▣	1 Week Ago
H	Thread unsafe modification in sing...	.../Install.java:54	⊞	1 Week Ago
M	Unrestricted dispatch	.../ForwardMe.java:41	▣	1 Week Ago
M	Unrestricted document type defini...	.../xxe.java:48	▣	1 Week Ago
L	Unsafe reflection	.../Install.java:111	▣	1 Week Ago
M	XML Path (XPath) Language injecti...	.../XPathQuery.java:53	⊞	1 Week Ago

* Refactor the JDBC code to use the "PreparedStatement" API instead of "Statement".
* Add a positional parameter to the SQL statement using "?".
* Bind the tainted value to the parameter using the "setString" method: "PreparedStatement.setString(1, ""user"")".

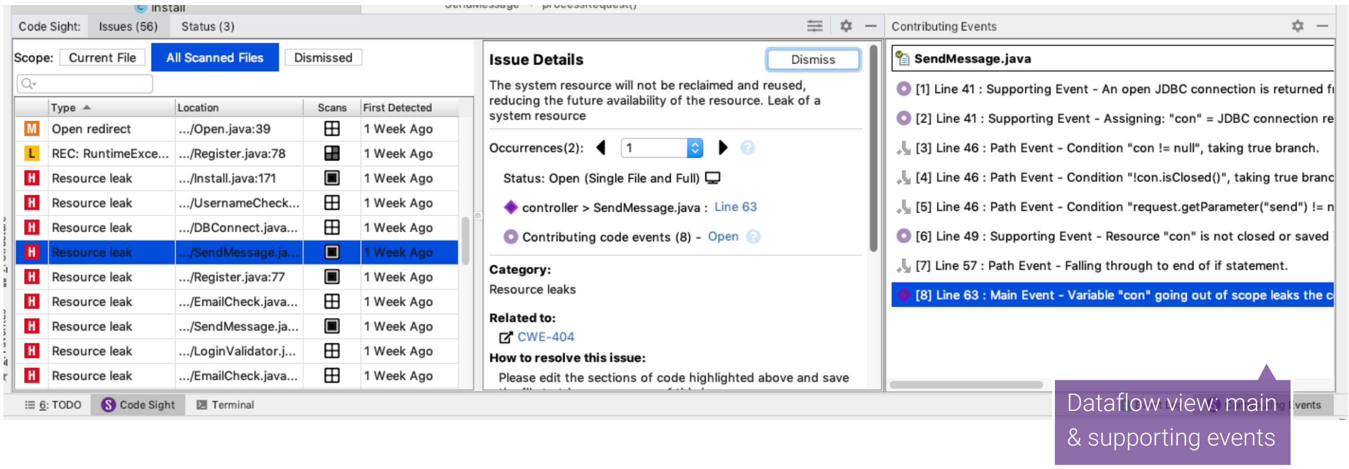
Learn how to avoid this type of issue in the future:

- OWASP Top 10
- Defensive Java Programming for EE Web Applications
- OWASP Top 10 (2017)
- Defensive Programming for PHP Security
- Foundations of COBOL Security
- Risk-Based Security Testing Strategy
- Introduction to PHP Security
- PCI/DSS Security
- Securing Python Web Applications
- Defensive Programming for COBOL

First detected:

IDE and Plugin Updates: The following plugin is ready to update: Synopsys Code Sight (2019-07-19 10:59)

Related eLearning courses



Install
SendMessage › processRequest()
Code Sight: Issues (56) Status (3)

Scope: Current File | **All Scanned Files** | Dismissed

	Type	Location	Scans	First Detected
M	Open redirect	.../Open.java:39	⊞	1 Week Ago
L	REC: RuntimeExce...	.../Register.java:78	▣	1 Week Ago
H	Resource leak	.../Install.java:171	▣	1 Week Ago
H	Resource leak	.../UsernameCheck...	⊞	1 Week Ago
H	Resource leak	.../DBConnect.java...	⊞	1 Week Ago
H	Resource leak	.../SendMessage.ja...	▣	1 Week Ago
H	Resource leak	.../Register.java:77	▣	1 Week Ago
H	Resource leak	.../EmailCheck.java...	⊞	1 Week Ago
H	Resource leak	.../SendMessage.ja...	⊞	1 Week Ago
H	Resource leak	.../LoginValidator.j...	⊞	1 Week Ago
H	Resource leak	.../EmailCheck.java...	⊞	1 Week Ago

Issue Details

The system resource will not be reclaimed and reused, reducing the future availability of the resource. Leak of a system resource

Occurrences(2): ◀ 1 ▶

Status: Open (Single File and Full)

◆ controller > SendMessage.java : Line 63

○ Contributing code events (8) - Open

Category:

Resource leaks

Related to:

CWE-404

How to resolve this issue:

Please edit the sections of code highlighted above and save

Contributing Events

SendMessage.java

- [1] Line 41 : Supporting Event - An open JDBC connection is returned fı
- [2] Line 41 : Supporting Event - Assigning: "con" = JDBC connection re
- [3] Line 46 : Path Event - Condition "con != null", taking true branch.
- [4] Line 46 : Path Event - Condition "!con.isClosed()", taking true branc
- [5] Line 46 : Path Event - Condition "request.getParameter("send") != n
- [6] Line 49 : Supporting Event - Resource "con" is not closed or saved
- [7] Line 57 : Path Event - Falling through to end of if statement.
- [8] Line 63 : Main Event - Variable "con" going out of scope leaks the c

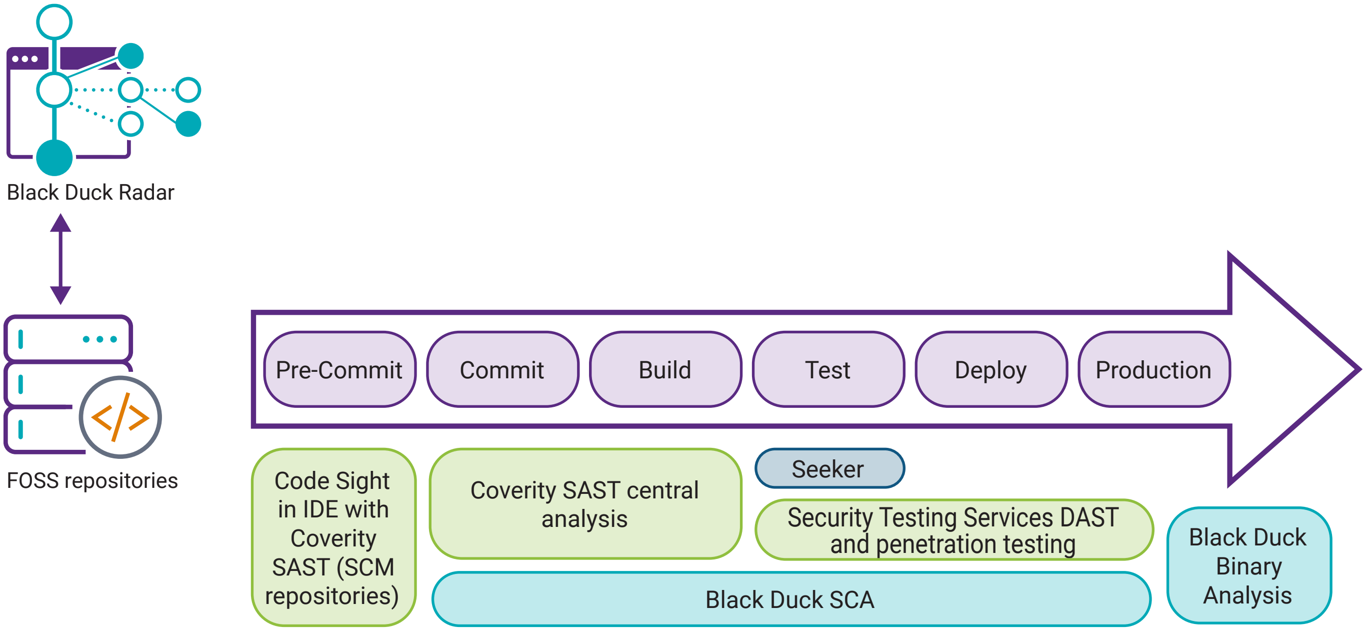
Dataflow view: main & supporting events

Open source management

According to the Black Duck 2019 OSSRA report, the majority of codebases (96%) reviewed by the Black Duck Audit Services team in 2018 contained some open source code.[2] More than 60% of the codebases contained known vulnerabilities, and 68% had license conflicts.[3] Policy violations can occur with security policies, technical/operational policies, or legal compliance policies that cite restricted legal licenses.

With tools such as Black Duck® SCA, developers can get information on the risks associated with their use of open source components—including security vulnerabilities and license policy violations—early in the development process. Black Duck Radar, a native Chrome browser plugin, shifts analysis even further left, as it can identify free and open source software (FOSS) components with known issues based on version numbers before developers download them for use.

Figure 3. How Black Duck products fit into the SDLC


Black Duck Radar
FOSS repositories
Pre-Commit
Commit
Build
Test
Deploy
Production
Code Sight in IDE with Coverity SAST (SCM repositories)
Coverity SAST central analysis
Seeker
Security Testing Services DAST and penetration testing
Black Duck SCA
Black Duck Binary Analysis

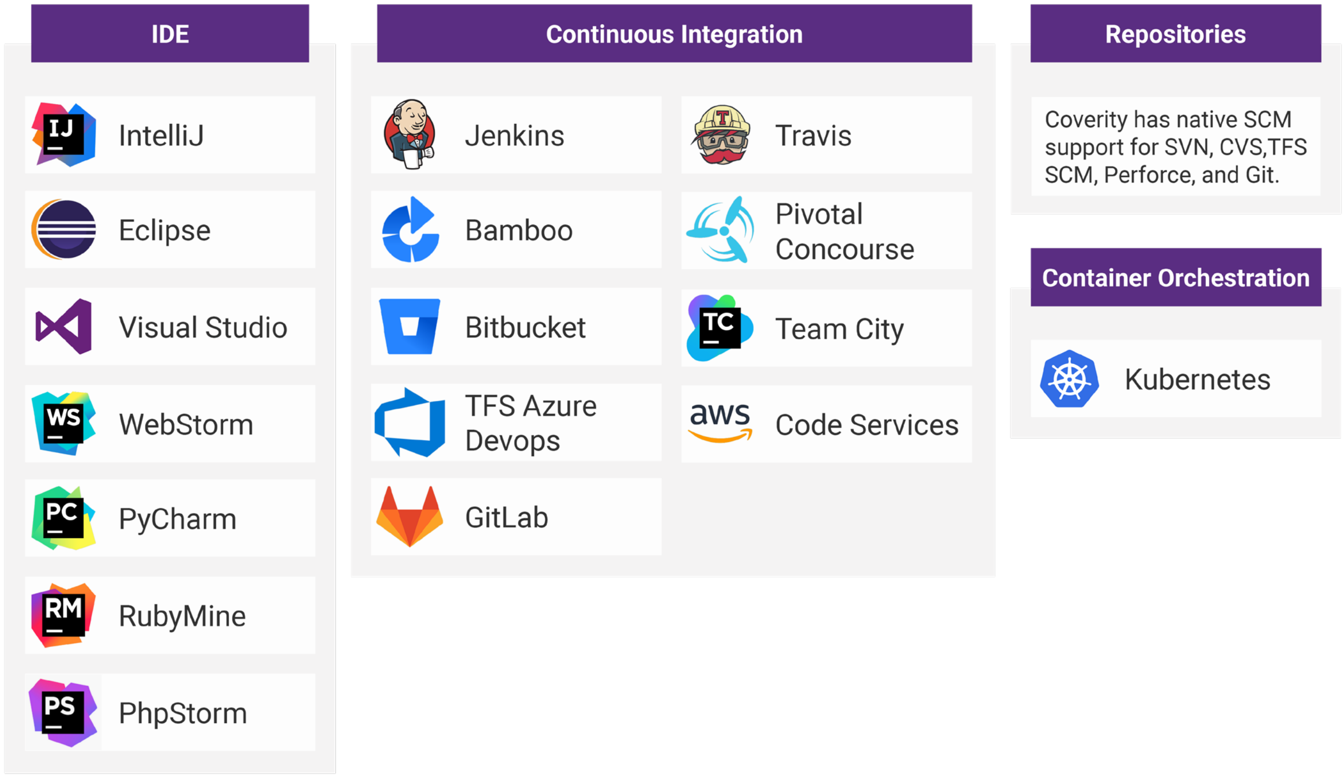
Cross-correlation of results between tools

Using the analysis results from one tool to improve the results from another leverages the strengths of each tool and saves developers time. The Polaris platform will soon support cross-correlation between Coverity SAST and Black Duck SCA. Developers can use Coverity SAST to determine which open source vulnerabilities it identifies are reachable in the code and to help prioritize issues for remediation. They can use Black Duck SCA to filter these Coverity-identified issues to those found in open source components to be triaged and mitigated using Black Duck SCA. They can then use Coverity SAST to focus on issues related to proprietary code in the same application.

DevOps leads need integrations for development environments, tools, and cloud platforms

DevOps leads don't want to have to write scripts to get new AppSec tools to work in their CI/CD workflows. Instead, they want tools that are ready to integrate with their CI build servers, container orchestration and cloud platforms, and repositories. With Polaris, Black Duck products (e.g., Coverity SAST, Black Duck SCA, Seeker® Interactive Analysis) work with a myriad of developer tools, including Code Sight for market-leading IDEs, as seen in Figure 4 below.

Figure 4. Polaris platform support



IDE

- IntelliJ
- Eclipse
- Visual Studio
- WebStorm
- PyCharm
- RubyMine
- PhpStorm

Continuous Integration

- Jenkins
- Travis
- Bamboo
- Pivotal Concourse
- Bitbucket
- Team City
- TFS Azure Devops
- Code Services
- GitLab

Repositories

Coverity has native SCM support for SVN, CVS,TFS SCM, Perforce, and Git.

Container Orchestration

- Kubernetes

Scalability

With Polaris, it's possible to quickly onboard and analyze thousands of applications and support tens of thousands of developers. The platform's support of industry-standard cloud platforms on the Kubernetes orchestration engine enables accurate and comprehensive security scanning that elastically scales with your business requirements.

CISOs focus on security vulnerabilities and policy compliance

CISOs and security leads need to be able to understand and manage their organizations' security risk posture across their application portfolio at any time. They need to be able to identify, prioritize, and resolve the most critical security vulnerabilities that threaten their organizations. Polaris combines best-in-class security tools with robust executive security reporting dashboards to provide cross-product aggregated reporting. The platform also cross-correlates analysis results to ensure reports are highly accurate and comprehensive.

Figure 5. Polaris cross-product aggregated reporting view of Coverity SAST, Black Duck SCA, and Seeker IAST analysis results

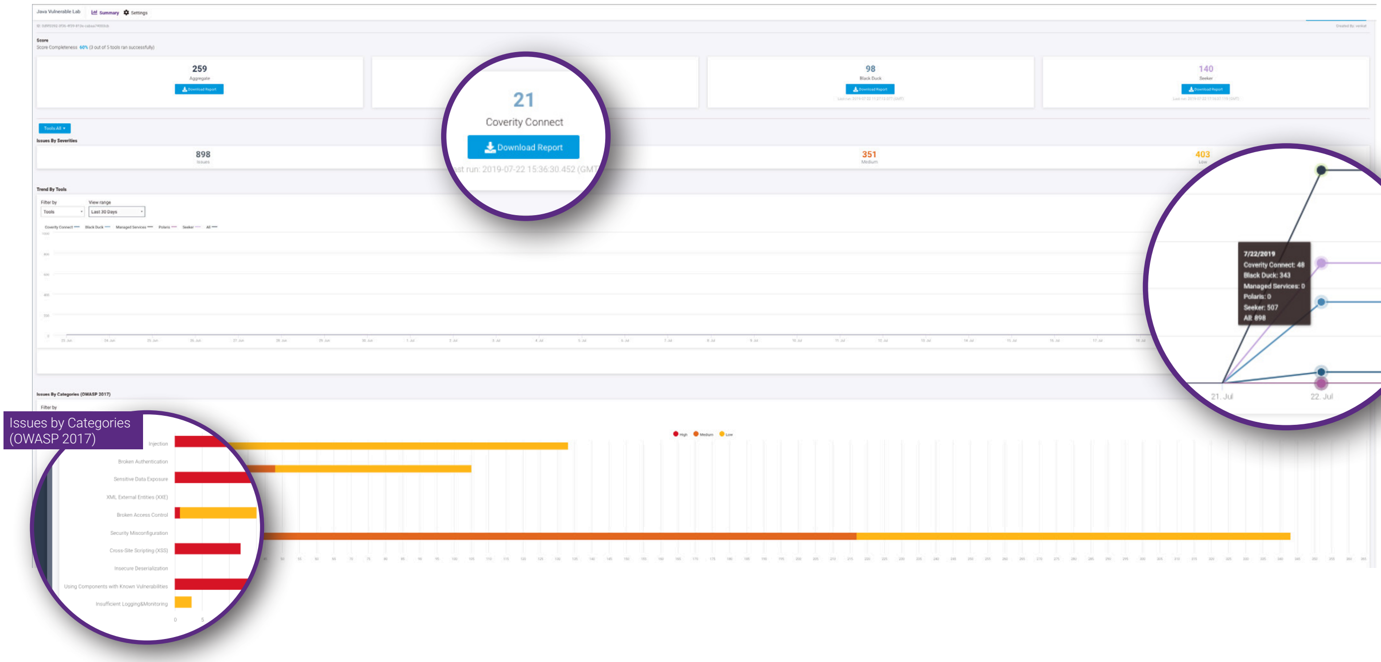
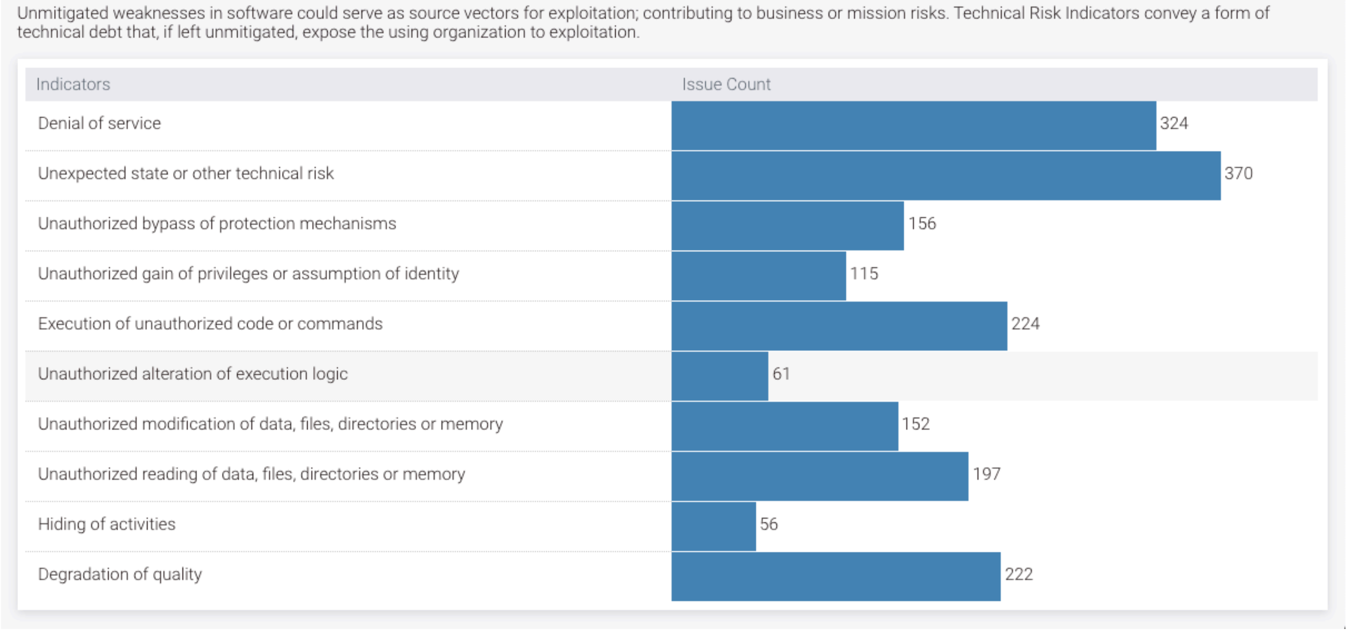


Figure 5 shows the Polaris reporting option to download either individual analysis reports or an aggregate report of analysis results for a project. For the project shown, analyses from Coverity SAST, Black Duck SCA, and Seeker IAST are available.

Polaris also provides reporting dashboards of industry-recognized security standards (e.g., OWASP Top 10, CWE/SANS Top 25). Users can create priority lists (e.g., top issues by technical risk indicators, as shown in Figure 6) so they can focus on the issues that matter most to the organization. Predefined filters allow users to filter and group issues by CWE, standards taxonomy, priority list, risk indicator, path, and individual developer owners for quick remediation. Polaris provides central policy management features, an aggregated risk profile of issues, trend information for categories and issues over time, and the ability to generate PDF reports for audit and management reporting.

Figure 6. Polaris Technical Risk Indicators view



Unmitigated weaknesses in software could serve as source vectors for exploitation; contributing to business or mission risks. Technical Risk Indicators convey a form of technical debt that, if left unmitigated, expose the using organization to exploitation.

Indicators	Issue Count
Denial of service	324
Unexpected state or other technical risk	370
Unauthorized bypass of protection mechanisms	156
Unauthorized gain of privileges or assumption of identity	115
Execution of unauthorized code or commands	224
Unauthorized alteration of execution logic	61
Unauthorized modification of data, files, directories or memory	152
Unauthorized reading of data, files, directories or memory	197
Hiding of activities	56
Degradation of quality	222

Best-in-class AppSec tools and services

Black Duck application security solutions have been recognized as industry leaders in industry analyst reports, such as the Gartner Magic Quadrant for Application Security Testing, The Forrester Wave™: Static Application Security Testing, Q3 2023, and The Forrester Wave™: Software Composition Analysis, Q2 2023. Our products and services help development and security teams build secure, high-quality software faster.

Coverity SAST. Coverity SAST helps developers find and fix security defects early in the SDLC, with support for 20 languages and over 70 frameworks and template engines, as well as security checkers to help ensure compliance with OWASP Top 10, CWE/SANS Top 25, PCI DSS, and other standards. Coverity SAST gives teams the flexibility to analyze code in the IDE and on the build server, on-premises and in the cloud.

Black Duck SCA. Black Duck SCA enables teams to secure and manage open source across their software supply chain. Black Duck SCA's unique multifactor open source discovery technology accurately detects open source in source code, binaries, and container images, giving development, security, and legal teams complete visibility into their open source security and license compliance risks. In addition, integrated policy management allows teams to automate open source governance, so they can build fast while staying secure and compliant.

Seeker IAST. Seeker helps development, QA, and security teams automate application security testing with CI and test automation tools. Seeker is the only IAST solution that actively verifies that identified vulnerabilities are exploitable, using patented technology, reducing false positives to near zero. Its unique sensitive-data tracking feature automatically detects when user-designated sensitive data is exposed in logs, databases, or files.

Managed Security Testing. Black Duck Managed Security Testing Services deliver on-demand security testing performed by a team of security experts, helping organizations cost-effectively address complex test scenarios. Black Duck's Managed Penetration Testing combines testing tools and in-depth manual tests focusing on business logic to find vulnerabilities outside common standards, including authentication checks, access control testing, logging and monitoring, workflow bypass, and manual review to identify false positives.

Polaris. Polaris brings Black Duck's tools together to provide a comprehensive, automated application security solution that enables teams to build secure software faster. The Code Sight IDE plugin integrates security analysis into the developer's desktop, while the Polaris central server gives security and development teams a single-pane-of-glass view of project vulnerability trends and helps them manage compliance with the security standards and regulations that are most important to their organization.

About the authors

Utsav Sanghani, *former Senior Product Manager, Software Integrity Group, Synopsys (now Black Duck)*
Utsav has spent a good part of the last decade with various enterprise software products, including productivity tools, software composition analysis, and static analysis. He drove thought leadership, strategic initiatives, and tangible solutions with a specific focus on making developers and DevOps solutions compatible with application security.

Anna Chiang, *former Senior Product Marketing Manager, Software Integrity Group, Synopsys (now Black Duck)*

Anna is a Certified Information Systems Security Professional who has worked in product marketing for SAST, IAST, and UEBA security products, platform product management, and developer programs management for enterprise and mobile application solutions.

References

1. Gene Kim, Jez Humble, John Willis, and Patrick Debois, The DevOps Handbook, IT Revolution, 2016.
2. Black Duck, 2019 Open Source Security and Risk Analysis, 2019.
3. Ibid.

About Black Duck

Black Duck® offers the most comprehensive, powerful, and trusted portfolio of application security solutions in the industry. We have an unmatched track record of helping organizations around the world secure their software quickly, integrate security efficiently in their development environments, and safely innovate with new technologies. As the recognized leaders, experts, and innovators in software security, Black Duck has everything you need to build trust in your software. Learn more at www.blackduck.com.