

Black Duck

Software Composition Analysis

Secure and manage
open source throughout
the software supply
chain

Overview

Black Duck is a comprehensive solution for managing security, license compliance, and code quality risks that come from the use of open source in applications, containers, and infrastructure-as-code (IaC). Named a leader in software composition analysis (SCA) by Forrester, Black Duck gives you unmatched visibility into third-party code, enabling you to control it across your software supply chain and throughout the application life cycle.

An integrated solution for source and binaries

Only Black Duck combines versatile open source risk management with deep binary inspection to provide a best-in-class SCA solution that helps you minimize risks associated with open source and other third-party software. In a time when [open source composes 70% of the average codebase](#), Black Duck empowers your development, operations, procurement, and security teams to:

- **Find and fix security vulnerabilities** at each stage in the SDLC, with detailed, vulnerability-specific remediation guidance and technical insight.
- **Eliminate risk of open source license noncompliance** and safeguard your intellectual property by using the industry's largest open source knowledge base to identify which of 2,750 licenses are relevant to the open source in your applications (including code snippets from larger components).
- **Avoid development cost overruns and combat code decay** with operational risk metrics associated with poor open source code quality.
- **Build a complete software Bill of Materials (SBOM)** by scanning virtually any software, firmware, IaC, and source code. Export SBOMs in standardized formats, such as SPDX and CycloneDX.
- **Automatically monitor for new vulnerabilities** that affect your BOM, with custom policies and workflow triggers to accelerate remediation and reduce your risk exposure.

Discover

- **Identify** open source in code, binaries, and containers.
- **Detect** partial and modified components.
- **Automate** scanning with DevOps integrations.

Protect

- **Map** components to known vulnerabilities.
- **Identify** license and component quality risks.
- **Monitor** for new vulnerabilities in development and production.

Manage

- **Set and enforce** open source use and security policies.
- **Automate** policy enforcement with DevOps integrations.
- **Prioritize and track** remediation activities.

Key benefits

Get deeper, more streamlined analysis

Black Duck identifies more open source, with greater accuracy, using a unique multifactor detection technology to generate and validate a complete BOM to track declared components, unique file hash signatures, dependencies resolved during a build, and open source code snippets. Black Duck applies these scanning methods in an efficient manner, bringing security and compliance to every stage and role in the SDLC. This includes Rapid Scan in the IDE, and integrations with CI/CD and binary repository tools for deeper analysis during build and post-build stages.

Find and fix vulnerabilities quickly

Black Duck's open source security risk insight combines curated data from public sources (e.g., NVD) and detailed, proprietary analysis from the Synopsys Cybersecurity Research Center (CyRC). Get notified of new vulnerabilities weeks before they are published in the NVD (reducing your window of exposure), and benefit from our exclusive enhanced vulnerability data and Black Duck Security Advisories (BDSAs), including:

- Critical risk metrics, vulnerability-specific technical insight, exploit details, and impact analysis
- CVSS 2 and CVSS 3 scoring and CWE classification data
- Common Attack Pattern Enumeration and Classification (CAPEC)
- Temporal scoring not provided by the NVD
- Component-level upgrade and remediation guidance, mitigating factors, and compensating controls
- Vulnerability impact analysis to determine if the vulnerable code is being called by the application
- Custom vulnerability risk scoring to match your company risk profile
- Vulnerabilities are prioritized for remediation across multiple critical data points, including severity, solution availability, exploitability, CWE, and reachability

Automatically enforce security and use policies

Configure your open source security and use policies based on a comprehensive array of criteria, including license type, vulnerability severity, open source component version, and more. Enforce policies with automatic workflow triggers, notifications, and bidirectional Jira integration for accelerated remediation initiation and reporting.

Identify open source risks, even without source code

With Black Duck in your toolkit, you can quickly and easily analyze vendor-supplied binaries to identify weak links in your software supply chain without access to the source code. Get deep, actionable risk metrics to make informed decisions about your use and procurement of technologies before they put you at risk. Black Duck's intelligent scan client automatically determines if the target software is source or a compiled binary, then identifies and catalogs all third-party software components, associated licenses, and known vulnerabilities affecting your applications.

Scanning

Languages

- C
- C++
- C#
- Clojure
- Erlang ■
- Golang
- Groovy
- Java
- JavaScript
- Kotlin
- Node.js ■
- Objective-C
- Perl ■
- Python
- PHP ■
- R ■
- Ruby
- Scala ■
- Swift ■
- .NET Cloud technologies

Package Managers

- NuGet ■
- Hex ■
- Vndr ■
- Godep ■
- Dep ■
- Maven ■
- Gradle ■
- Npm
- CocoaPods
- Cpanm ■
- Conda ■
- Pear ■
- Composer ■
- Pip
- Packrat ■
- RubyGems
- SBT ■
- Bazel ■
- Cargo ■
- C/C++ (Clang) ■

- GoLang
- Erlang/Hex
- Rebar
- Python
- Yarn
- Yocto

BDBA Package Manager Support

- Distro-package-manager: Leverages information from a Linux distribution package manager database to extract component information.
- The remaining four methods are only applicable to Java bytecode:
 - pom: Extracts the Java package, group name, and version from the pom.xml or pom.properties files in a JAR file.
 - manifest: extracts the Java package name and version from the entries in the MANIFEST.MF file in a JAR file.
 - jar-filename: Extracts the Java package name and version from the jar-filename.
 - hashsum: Uses the sha1 checksum of the JAR file to look it up from known Maven Central registered Java projects.

Binary formats

- Native binaries
- Java binaries
- .NET binaries
- Go binaries

Compression formats

- Gzip (.gz)
- bzip2 (.bz2)
- LZMA (.lz)
- LZ4 (.lz4) ✖
- Compress (.Z)
- XZ (.xz)
- Pack200 (.jar)
- UPX (.exe)
- Snappy
- DEFLATE
- zStandard (.zst) ✖

Archive formats

- ZIP (.zip, .jar, .apk, and other derivatives)
- XAR (.xar) ✖
- 7-Zip (.7z)
- ARJ (.arj)
- TAR (.tar)
- VM TAR (.tar) ✖
- cpio (.cpio)
- RAR (.rar)
- LZH (.lzh) ✖
- Electron archive (.asar) ✖
- DUMP

Installation formats

- Red Hat RPM (.rpm)
- Debian package (.deb)
- Mac installers (.dmg, .pkg)
- Unix shell file installers (.sh, .bin)
- Windows installers (.exe, .msi, .cab)
- vSphere Installation Bundle (.vib) ✖
- Bitrock Installer ✖
- Installer generator formats that are supported:
 - 7z, zip, rar self extracting .exe ✖
 - MSI Installer ✖
 - CAB Installer ✖
 - InstallAnywhere ✖
 - Install4J ✖
 - InstallShield ✖
 - InnoSetup ✖
 - Wise Installer ✖
 - Nullsoft Scriptable Install System (NSIS) ✖
 - WiX Installer ✖

Firmware formats

- Intel HEX ✖
- SREC ✖
- U-Boot ✖
- Arris firmware ✖
- Juniper firmware ✖
- Kosmos firmware ✖
- Android sparse file system ✖
- Cisco firmware ✖

File systems / disk images

- ISO 9660 / UDF (.iso) ✕
- Windows Imaging ✕
- ext2/3/4 ✕
- JFFS2 ✕
- UBIFS ✕
- RomFS ✕
- Microsoft Disk Image ✕
- Macintosh HFS ✕

- VMware VMDK (.vmdk, .ova) ✕
- QEMU Copy-On-Write (.qcow2) ✕
- VirtualBox VDI (.vdi) ✕
- QNX—EFS, IFS ✕
- NetBoot image (.nbi) ✕
- FreeBSD UFS ✕

Container Formats

- Docker

Supported IaC file formats

- HCL (Terraform)
- HTML
- JSON
- JSX
- TOML
- TSX
- Vue
- XML
- YAML

Black Duck | Integrations

Cloud technologies

Cloud platforms

- Amazon Web Services
- Google Cloud Platform
- Microsoft Azure
- Pivotal Cloud Foundry

Container platforms

- Docker
- OpenShift
- Pivotal Cloud Foundry
- Kubernetes Package managers

Databases

- PostgreSQL

DevOps tools

IDEs

- Eclipse
- Visual Studio IDE
- IntelliJ IDEA

- WebStorm
- PyCharm
- RubyMine
- PhpStorm
- VS Code
- Android Studio

Continuous integration

- Jenkins
- TeamCity
- Bamboo
- Team Foundation Server
- Travis CI
- CircleCI
- GitLab CI
- Visual Studio Team Services
- Concourse CI
- AWS CodeBuild
- Codeship
- Azure DevOps
- GitHub Actions
- OpenShift

Workflow and notifications

- Jira
- Slack
- Email
- SPDX
- Azure Boards
- Microsoft Teams

Binary and source repositories

- Artifactory
- Nexus

Application security suites

- IBM AppScan
- Micro Focus Fortify
- SonarQube
- ThreadFix
- Cybric
- Code Dx
- Fortify
- ZeroNorth

The Synopsys difference

Synopsys provides integrated solutions that transform the way you build and deliver software, accelerating innovation while addressing business risk. With Synopsys, your developers can secure code as fast as they write it. Your development and DevSecOps teams can automate testing within development pipelines without compromising velocity. And your security teams can proactively manage risk and focus remediation efforts on what matters most to your organization. Our unmatched expertise helps you plan and execute any security initiative. Only Synopsys offers everything you need to build trust in your software.

For more information about the Synopsys Software Integrity Group, visit us online at www.synopsys.com/software.

©2023 Synopsys, Inc. All rights reserved. Synopsys is a trademark of Synopsys, Inc. in the United States and other countries. A list of Synopsys trademarks is available at www.synopsys.com/copyright.html. All other names mentioned herein are trademarks or registered trademarks of their respective owners. April 2023.