Advisory database sources and matching service

- Advisory database sources
- Severity and scoring priority
- Vulnerability matching
- Supported package ecosystems

Reliable information sources are key for Docker Scout's ability to surface relevant and accurate assessments of your software artifacts. Given the diversity of sources and methodologies in the industry, discrepancies in vulnerability assessment results can and do happen. This page describes how the Docker Scout advisory database and its CVE-to-package matching approach works to deal with these discrepancies.

Advisory database sources

Docker Scout aggregates vulnerability data from multiple sources. The data is continuously updated to ensure that your security posture is represented using the latest available information, in real-time.

Docker Scout uses the following package repositories and security trackers:

- AlmaLinux Security Advisory
- Alpine secdb
- Amazon Linux Security Center
- Bitnami Vulnerability Database
- CISA Known Exploited Vulnerability Catalog
- CISA Vulnrichment
- Chainguard Security Feed
- Debian Security Bug Tracker

- Exploit Prediction Scoring System (EPSS)
- GitHub Advisory Database
- GitLab Advisory Database
- Golang VulnDB
- National Vulnerability Database
- Oracle Linux Security
- Photon OS 3.0 Security Advisories
- Python Packaging Advisory Database
- RedHat Security Data
- Rocky Linux Security Advisory
- RustSec Advisory Database
- SUSE Security CVRF
- Ubuntu CVE Tracker
- Wolfi Security Feed
- inTheWild, a community-driven open database of vulnerability exploitation

When you enable Docker Scout for your Docker organization, a new database instance is provisioned on the Docker Scout platform. The database stores the Software Bill of Materials (SBOM) and other metadata about your images. When a security advisory has new information about a vulnerability, your SBOM is cross-referenced with the CVE information to detect how it affects you.

For more details on how image analysis works, see the image analysis page.

Severity and scoring priority

Docker Scout uses two main principles when determining severity and scoring for CVEs:

- Source priority
- CVSS version preference

For source priority, Docker Scout follows this order:

1. Vendor advisories: Scout always uses the severity and scoring data from the source that matches the package and version. For example, Debian data for Debian packages.
2. NIST scoring data: If the vendor doesn't provide scoring data for a CVE, Scout falls back to NIST scoring data.

For CVSS version preference, once Scout has selected a source, it prefers CVSS v4 over v3 when both are available, as v4 is the more modern and precise scoring model.

Vulnerability matching

Traditional tools often rely on broad Common Product Enumeration (CPE) matching, which can lead to many false-positive results.

Docker Scout uses Package URLs (PURLs) to match packages against CVEs, which yields more precise identification of vulnerabilities. PURLs significantly reduce the chances of false positives, focusing only on genuinely affected packages.

Supported package ecosystems

Docker Scout supports the following package ecosystems:

- .NET
- GitHub packages
- Go
- Java
- JavaScript
- PHP
- Python
- RPM
- Ruby
- `alpm` (Arch Linux)
- `apk` (Alpine Linux)
- `deb` (Debian Linux and derivatives)

□□□ #2
□ □□□ □□□ 2026-10-31 19:35:55 UTC
□ □□□ □□□ 2026-10-31 19:35:56 UTC