snyk

Snyk security and compliance

How does Snyk handle your data?

Last update: March 12, 2021

Snyk is a cloud native application security platform and as such, we place the utmost importance on data security. **Fully understanding your privacy and security needs, the goal of this document is to provide you with transparency as to how and what data is accessed, transferred, and stored by Snyk.**

The data handled by Snyk will vary depending on the product you are using, how you are integrating with Snyk, and your Snyk deployment. Because Snyk is a fast moving product, the types of data accessed and stored might change with the introduction of a new capability or changes to an existing capability.

Flexible deployment options

Snyk customer DevSecOps programs thrive off the fast time-to-value and ease of use from our SaaS model. In case you have further data sovereignty requirements we provide additional deployment options.

SaaS - A multi-tenant, fully managed and provisioned by Snyk service. The most common deployment option used by Snyk customers.

SaaS with Broker - A client service that is installed on your private infrastructure, acting as a proxy between Snyk and your on-premise codebase. The Broker securely handles inbound and outbound connections, encrypting data during transit, and deliberately controlling the access Snyk has to your data. Sensitive credentials stay behind your firewall.

Cloud Native Appliance (CNA) - For organizations that have stringent data sovereignty requirements, Snyk is also available in a self-hosted deployment model with a cloud native appliance. Available initially on AWS (hosted within a customer's AWS account), customers achieve fully isolated data and isolated computing cycles with the simplicity of rapid provisioning by Snyk.

Be sure to reach out to your Snyk contact to find out more details on how these different options can be leveraged to meet your needs.

snyk

Customer dataflows across Snyk

Snyk provides a wide range of development tools and integration points, requiring different types of data, and entailing different data interactions. The sections below provide an overview of both the common types of data Snyk accesses and stores as well as product and integration-specific types. The information is reviewed twice annually or when a significant change occurs within the product operations.

Common data types

- **Vulnerability data** - Snyk stores information on the vulnerabilities identified in customers applications and related remediation context.
- **Integration-related data** - Snyk stores information required to set up an integration with Snyk. Examples: tokens and configurations.
- **User list** - For the purposes of an accurate contributor counting, Snyk accesses commits from the last 90 days for repositories monitored and stores a hashed version of user emails.
- **Billing data** - Snyk stores information required for billing your Snyk account.
- **Vulnerability source** - Snyk stores information on where the vulnerability was identified. Examples: source code repository/registry, file name and location, dependency tree, vulnerability path.
- **User behavior analytics** - Snyk stores various types of information pertaining to usage patterns. Examples: Website visits, executed CLI commands.
- **User data** - Snyk stores basic user information. Examples: user name, ID, email address.

Product-specific data types

We know how important it is for you to protect your data. Our products only access and store the information needed to ensure accurate analysis.

Snyk Open Source

- Snyk accesses manifest and build configuration files in order to identify your open source dependencies.
- Snyk accesses and stores the names and version numbers of your dependencies.
- Snyk stores the names of associated licenses, including copyright and attribution information.
- Snyk accesses and stores repository-specific information.
- Snyk accesses and stores Git provider push & pull specific information. Examples: contributor name, filenames, timestamps.

OPTIONAL ADD-ONS (opt-in)

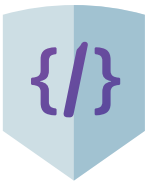
- For Reachable Vulnerabilities computation - Snyk stores your source code to facilitate the building of a call graph. Once the analysis completes, your code is removed from the Snyk system. Only the call graph and function names are maintained.
- For Lambda integration only - Snyk pulls a short term copy then destroys it as part of analysis.

Snyk Container

- Snyk accesses and stores package versions, executable hashes/versions, operating system, container image metadata (e.g. rootfs hashes, history), image ID.
- Snyk accesses and stores information pertaining to the parent image - name/version/tag.
- Snyk accesses and stores RUN instructions from Dockerfile.
- Kubernetes configurations - Snyk accesses workload security settings (e.g. 'run as root'). This is only accessed if you use Snyk's Kubernetes integration.
- Container registry integrations - Snyk accesses and then stores a short term copy of the container image (unless a broker is used). This copy is removed from the Snyk network after analysis.

Snyk Infrastructure as Code

- Snyk accesses Infrastructure as Code files, stores them for a short period of time for analysis, and subsequently completely deletes them from our system. Outside of scans, IaC files are not stored by Snyk.
- In the Snyk UI for Infrastructure as Code there is a side-by-side code review interface. Snyk generates this dynamically using the provided source code repository integration - your source code is not stored by Snyk.



Snyk Code

- Snyk accesses your repository code for a one-time analysis, caching it for a period of up to 24 hours. After this period, only the location (file path, line, and column) to the issues found, the issue id and explanations are maintained. Your code is removed and is not stored in the Snyk network or logs.
- Results are stored in a database and used for analytic and monitoring purposes by Snyk.
- Snyk Code does not use any customer code (1) for engine training purposes, or (2) to extract examples to show possible fixes.
- The scan results do not contain original source code but rather pointers to positions (e.g., files, line, and column numbers), plus identification meta-information so that results are displayed using the correct version of the source code.
- Snyk stores repository-specific information. Examples: Names of the Git repository, file names.
- The server infrastructure ensures separation between customers by using authentication and authorization. Snyk Code uses software controls to ensure customer data segregation. All communication is encrypted using high-grade industry-standard protocols.

Snyk certifications

Coming in 2021 - ISO27001 & ISO27017 compliance

Snyk policies:

Privacy

Data processing

Tracking & analytics