Security Policy

How to Report Potential Security Vulnerabilities

Any potential security vulnerabilities in the entire Spring portfolio should be reported through the Security Advisories page.

The Spring team needs to receive reports of potential security vulnerabilities through GitHub's ability to privately report a security vulnerability. To simplify the process, the spring-projects/security-advisories repository is used to report potential vulnerabilities for any project within the Spring portfolio (this includes projects in other GitHub organizations, such as Spring Cloud).

Viewing Security Vulnerabilities

All security vulnerabilities are posted to https://spring.io/security/.

Guidelines for Reporting a Vulnerability

If you believe you have found a security vulnerability, please report it as described in How to Report Potential Security Vulnerabilities. Below, you can see examples of vulnerabilities and examples of non-vulnerabilities.

Examples of Vulnerabilities

For examples of vulnerabilities, refer to https://spring.io/security/.

Examples of Non-vulnerabilities

Unsafe Deserialization

The Spring team's stance is that, in order for deserialization to be considered a vulnerability in Spring, Spring must pass data from an untrusted source (that is, HTTP parameters) into a method that performs deserialization in a way that produces a CVE. The reason for this stance is that deserialization of arbitrary types is necessary but cannot be made secure.

Spring provides necessary tools for developers, but it is the responsibility of the developer to use them securely. This is no different than any other library or the JDK itself. If a developer passes untrusted data from an HTTP request to `ProcessBuilder`, then it is not a CVE in the JDK but in the application that used `ProcessBuilder` incorrectly. If a developer uses untrusted data to create an SQL query by using String concatenation, it is not a CVE in the SQL driver but in the application for not using prepared SQL statements. Similarly, if a developer passes untrusted data into a deserialization method, it is the developer that needs to ensure it is safe to do so.

Vulnerabilities in Dependencies

Vulnerabilities in Spring's dependencies should be reported to the respective project and not to the Spring team.

Vulnerable Dependency Versions

The Spring team does its best to keep its dependencies up to date regardless of whether a dependency contains a vulnerability. However, we do not consider it a vulnerability in Spring when Spring defines a vulnerable dependency version, because developers can override these versions and because releasing for any transitive dependency would become unmanageable for the Spring portfolio.

It is up to the developer of the dependency to release a compatible version with the security fix. If this is made available, the Spring project will be updated to that dependency version prior to releasing the next version of the Spring project.

Typically, there is not a special release for updating dependency versions. Instead, the Spring team encourages developers to override the version until the next Spring release.