



Security Training for ●●●●●●

Developers

Part 2: Section 4: Dependency Management



About This Training

- For Developers
- Designed in three parts
 - **Foundations and Systems Security** (for all Contributors)
 - **Vulnerability Management** (for all Contributors)
 - **Advanced Subjects** (for Committers and future Committers)

- Each part has a number of sections
 - Typically 15 minutes of study
 - Choose what matches your needs
- Content
 - Videos (live or pre-recorded)
 - Readings
 - Exercises





Evaluating risk of dependencies



Dependency Risks

- **Unfixed vulnerabilities**
- **Maintenance**
- **Code integrity**

Examples of past issues with dependencies



- XZ - compression library
 - Malicious co-maintainer included a backdoor
 - Single-maintainer project, social engineering attack
- Log4j - logging library
 - A bug in a popular library, that was not considered security-related



Common questions




- Is the project active?
- How big is the committer/maintainer team?
- Who can push to the repository?
- How do they protect code integrity?
- Do they provide stable APIs (Application Programming Interface) and ABI (Application Binary Interface)?

Evaluating dependencies: Best Practices Badge

- Human-filled
- <https://www.bestpractices.dev/en/projects>

 OpenSSF Best Practices 100% [Projects](#) [Sign Up](#) [Login](#) 



Eclipse Steady

Projects that follow the best practices below can voluntarily self-certify and show that they've achieved an Open Source Security Foundation (OpenSSF) best practices badge. [Show details](#)

If this is your project, please show your badge status on your project page! The badge status looks like this: `openssf best practices` passing Here is how to embed it: [Show details](#)

These are the passing level criteria. You can also view the silver or gold level criteria.

[Expand panels](#) [Show all details](#) [Show only incomplete criteria](#)

Basics 13/13

Identification

What is the human-readable name of the project? [Show details](#)

Eclipse Steady

What is a brief description of the project?

Analyses your Java and Python applications for open-source dependencies with known vulnerabilities, using both static analysis and testing to determine code context and usage for greater accuracy.

What is the URL for the project (as a whole)?

<https://eclipse.github.io/steady>

What is the URL for the version control repository (It may be the same as the project URL)?

<https://github.com/eclipse/steady>

What programming language(s) are used to implement the project? [Show details](#)

Java, JavaScript, Python, ANTLR, Smarty, TSQL, Go, Shell, Groovy, Dockerfile, Makefile, PLpgSQL

Tools

Dependabot



- Github-native tool to help developers automatically **manage** and **secure** their **project dependencies**
 - Reduces risk from vulnerable dependencies
 - Ensures up-to-date software
 - Minimizes overhead of manual dependency management
- Main features
 1. Security Alerts
 2. Security Updates
 3. Version Updates

1. Security Alerts



- Dependabot creates alerts for vulnerabilities affecting dependencies
- Triage can be manual or automated
- Resolution can be dismissed or fixed
- Notification preference can be customised

To enable Dependabot alerts for projects via Otterdog, set [Repository Settings](#) `dependabot_alerts_enabled` in the repository config file from `.eclipsefdn`.

To enable Dependabot alerts by default for new repositories at organization level, set [Organization Settings](#) `dependabot_alerts_enabled_for_new_repositories`.

What is a Security Alert?



- Package Name
- Vulnerability ID
 - Github Advisory (GHSA-xxxx)
 - CVE Identifier (CVE-2024-xxxx)
- Severity
 - Low/Moderate/High/Critical
- Affected version range
- Patched version range
- Dependency type
- Manifest file path
- Reference links

ReDoS in py library when used with subversion #65



Opened 3 months ago on `py` (pip) · poetry/poetry.lock

Dismiss alert ▾

Package	Affected versions	Patched version
<code>py</code> (pip)	<code><= 1.11.0</code>	None

The `py` library through 1.11.0 for Python allows remote attackers to conduct a ReDoS (Regular expression Denial of Service) attack via a Subversion repository with crafted info data, because the `InfoSvnCommand` argument is mishandled.

The particular codepath in question is the regular expression at `py._path.svnur1.InfoSvnCommand.lspattern` and is only relevant when dealing with subversion (svn) projects. Notably the codepath is not used in the popular `pytest` project. The developers of the `pytest` package have released version `7.2.0` which removes their dependency on `py`. Users of `pytest` seeing alerts relating to this advisory may update to version `7.2.0` of `pytest` to resolve this issue. See [pytest-dev/py#287](#) (comment) for additional context.



dependabot [bot] opened this from `c9bd99` 3 months ago

Severity

Moderate 5.3 / 10

CVSS base metrics	
Attack vector	Network
Attack complexity	Low
Privileges required	None
User interaction	None
Scope	Unchanged
Confidentiality	None
Integrity	None
Availability	Low

CVSS3.1/AV:N/AC:L/PR:N/UI:N/S:U/CN:I/N:A/L

Tags

Development

Source: <https://docs.github.com/en/code-security/dependabot/dependabot-alerts>

How to triage a Security Alert?



- Assess severity
 - Focus first on High or Critical alerts
 - Check CVSS vector and exploitability
- Check usage
 - Is the vulnerability affecting my project?
- Test patched version
 - Create a test branch with patched version
 - Run existing tests & CI workflows
- PR update and merge

octo-org / octo-repo Private

<> Code Issues Pull requests Actions Projects Security Insights Settings

Overview

Reporting

Policy

Advisories

Vulnerability alerts

Dependabot 92

Code scanning

Secret scanning

47 Open 5 Closed

Package Ecosystem Manifest Severity Sort

flat vulnerable to Prototype Pollution Critical

#54 opened 2 months ago • Detected in flat (npm) • yarn.lock

Source: <https://docs.github.com/en/code-security/dependabot/dependabot-alerts>

How to triage a Security Alert?



- **Asses severity**
 - Focus first on High or Critical alerts
 - Checks CVSS vector and exploitability
- **Check usage**
 - Is this vulnerability affecting my project?
- **Test patched version**
 - Create a test branch with patched version
 - Run existing tests & CI workflows
- **PR update and merge**

Regular Expression Denial of Service in Addressable templates #2

The screenshot shows a GitHub Dependabot alert interface. At the top, it says 'Regular Expression Denial of Service in Addressable templates #2'. Below this, there's a green 'Open' button and text indicating the alert was opened last month on 'addressable (RubyGems) - Gemfile.lock'. A 'Dismiss alert' button is in the top right. The main content area shows a checkbox for 'Upgrade addressable to fix 1 Dependabot alert in Gemfile.lock' with a link to the alert details. Below this, it says 'Upgrade addressable to version 2.8.0 or later. For example:' followed by a code block: `gem "addressable", ">= 2.8.0"`. A green button with a gem icon and the text 'Create Dependabot security update' is highlighted with a red box. On the right side, there's a 'Severity' section showing 'High 7.5 / 10' and a 'CVSS base metrics' table.

CVSS base metrics	
Attack vector	Network
Attack complexity	Low
Privileges required	None
User interaction	None

Source: <https://docs.github.com/en/code-security/dependabot/dependabot-alerts>

OR use **Dependabot Security Updates**

2. Security Updates



- **Automated PRs** that help you update vulnerable dependencies
- **Dependabot:**
 - Checks if an update does not disrupt the dependency graph
 - PR to update to the minimum version with a patch
 - Links the PR to the Security Alert

To enable Dependabot alerts for projects via Otterdog, set [Repository Settings](#) `dependabot_security_updates_enabled` in the repository config file from `.eclipsefdn`.

To enable Dependabot alerts by default for new repositories at organization level, set [Organization Settings](#) `dependabot_security_updates_enabled_for_new_repositories`.

3. Version Updates



- Not directly related to security issues
- Automatically checks for new versions of project dependencies
- Creates a PR to bump the version when a new one is released
- Can be customized
 - Frequency
 - Target specific ecosystems
 - Update strategy

Readings



- Securing Development Branches Best Practices:
<https://eclipse-csi.github.io/security-handbook/developer/branches.html>
- Dependabot explained:
<https://eclipse-csi.github.io/security-handbook/developer/branches.html>
- Dependabot documentation:
<https://docs.github.com/en/code-security/dependabot>
- Mend Renovate explained:
<https://eclipse-csi.github.io/security-handbook/developer/branches.html#mend-renovate>
- Mend Renovate documentation: <https://docs.renovatebot.com/>



Thank you!