



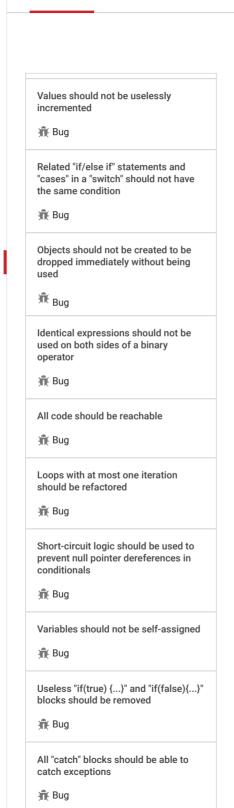


PHP static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your PHP code



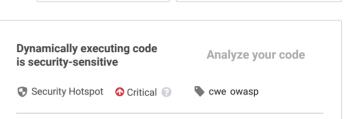
Tags



Constructing arguments of system

commands from user input is

security-sensitive



Search by name...

Executing code dynamically is security-sensitive. It has led in the past to the following vulnerabilities:

- CVE-2017-9807
- CVE-2017-9802

Some APIs enable the execution of dynamic code by providing it as strings at runtime. These APIs might be useful in some very specific metaprogramming use-cases. However most of the time their use is frowned upon as they also increase the risk of Injected Code. Such attacks can either run on the server or in the client (exemple: XSS attack) and have a huge impact on an application's security.

This rule marks for review each occurrence of the eval function. This rule does not detect code injections. It only highlights the use of APIs which should be used sparingly and very carefully. The goal is to guide security code reviews.

Ask Yourself Whether

- the executed code may come from an untrusted source and hasn't been sanitized.
- you really need to run code dynamically.

There is a risk if you answered yes to any of those questions.

Recommended Secure Coding Practices

Regarding the execution of unknown code, the best solution is to not run code provided by an untrusted source. If you really need to do it, run the code in a sandboxed environment. Use jails, firewalls and whatever means your operating system and programming language provide (example: Security Managers in java, iframes and same-origin policy for javascript in a web browser).

Do not try to create a blacklist of dangerous code. It is impossible to cover all attacks that way.

Avoid using dynamic code APIs whenever possible. Hard-coded code is always safer.

Noncompliant Code Example

eval(\$code_to_be_dynamically_executed)

See

- OWASP Top 10 2021 Category A3 Injection
- OWASP Top 10 2017 Category A1 Injection

Allowing unfiltered HTML content in WordPress is security-sensitive

Security Hotspot

Allowing unauthenticated database repair in WordPress is security-sensitive

Security Hotspot

Allowing all external requests from a WordPress server is security-sensitive

Security Hotspot

Security Hotspot

Disabling automatic updates is

security-sensitive

 <u>MITRE, CWE-95</u> - Improper Neutralization of Directives in Dynamically Evaluated Code ('Eval Injection')

Available In:

sonarcloud 👌 sonarqube

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.

Privacy Policy