

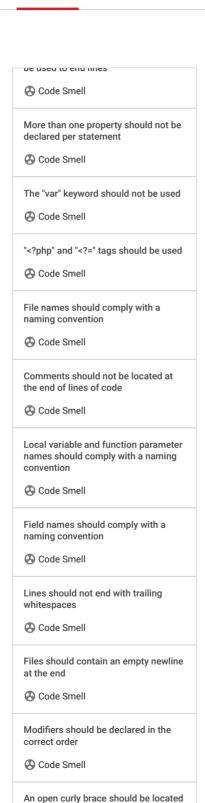


PHP static code analysis

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

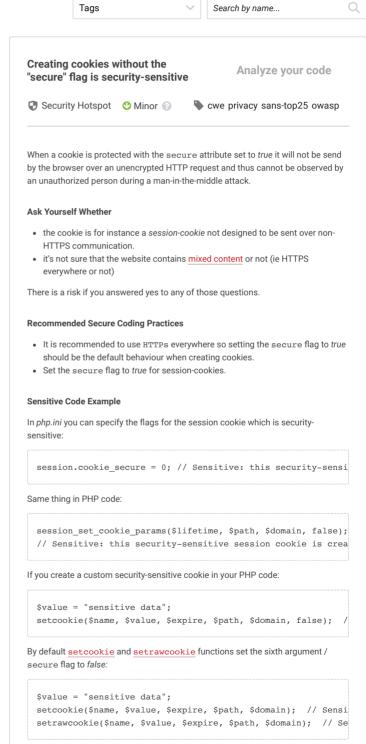
All rules (268)	6 Vulnerability 40	n Bug 51	Security Hotspot 33	Code Smell 144

Compliant Solution



at the beginning of a line

Code Smell



session.cookie_secure = 1; // Compliant: the sensitive cooki

An open curly brace should be located at the end of a line

Code Smell

Tabulation characters should not be used

Code Smell

Method and function names should comply with a naming convention

Code Smell

Creating cookies with broadly defined "domain" flags is security-sensitive

Security Hotspot

session_set_cookie_params(\$lifetime, \$path, \$domain, true);

\$value = "sensitive data";
setcookie(\$name, \$value, \$expire, \$path, \$domain, true); //
setrawcookie(\$name, \$value, \$expire, \$path, \$domain, true);/

See

- OWASP Top 10 2021 Category A4 Insecure Design
- OWASP Top 10 2021 Category A5 Security Misconfiguration
- OWASP Top 10 2017 Category A3 Sensitive Data Exposure
- MITRE, CWE-311 Missing Encryption of Sensitive Data
- MITRE, CWE-315 Cleartext Storage of Sensitive Information in a Cookie
- MITRE, CWE-614 Sensitive Cookie in HTTPS Session Without 'Secure' Attribute
- SANS Top 25 Porous Defenses

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