보안 관련 룰

Security-related Rules

The SonarQube Quality Model has three different types of rules: Reliability (bug), Vulnerability (security), and Maintainability (code smell) rules. But divided another way, there are only two types: security rules, and all the rest. The distinction between these two groups is not so much in what they catch but in where they come from and in the standards imposed on them.

What to expect from security-related rules

To be clear, the standard for most rules implemented in SonarQube language plugins is very strict: no false positives. For normal rules, you should be able to be confident that whatever is reported to you as an issue really is an issue.

But for security-related rules, the story is a little different. For instance, a lot of security guidelines talk about how "sensitive" data should be handled (e.g. not logged, not stored un-encrypted, &etc.). But since it's not really possible in a rule to tell which data is sensitive and which isn't, the choice becomes: maintain the no-false-positives standard and don't implement security-related rules, or implement security-related rules with a different standard.

That's why security-related rules cast a wider net than you may be used to seeing. The idea is that the rule will flag anything suspicious, and leave it to the human security auditor to cull the false positives and sent the real issues for remediation.

Security Hotspots are a special type of issue that identify sensitive areas of code that should be reviewed by a Security Auditor to determine if they are truly Vulnerabilities. See Security Audits and Reports for detail on Security Hotspots and the audit process.

Where security-related rules come from

The vast majority of security-related rules originate from established standards: CWE, SANS Top 25, and OWASP Top 10. To find rules that relate to any of these standards, you can search rules either by tag or by text. The standards that a rule relates to will be listed in the **See**section at the bottom of the rule description.

CWE

CWE stands for Common Weakness Enumeration. According to the CWE FAQ:

> *Common Weakness Enumeration (CWE™) is a formal list or dictionary of common software weaknesses that can occur in software's architecture, design, code or implementation that can lead to exploitable security vulnerabilities. CWE was created to serve as a common language for describing software security weaknesses; serve as a standard measuring stick for software security tools targeting these weaknesses; and to provide a common baseline standard for weakness identification, mitigation, and prevention efforts.*

The CWE is a hierarchy of weakness descriptions. The lowest level in the hierarchy is a "Weakness Base", which describes a granular weakness. Above Weakness Bases, are Weakness Classes and Categories. In general, rules are linked to Weakness Bases or Classes.

Tools which meet certain requirements can be certified as CWE Compatible. Those requirements are:

- You must be able to search for CWE-related rules using a CWE identifier. To do so in the SonarQube platform, simply drop the CWE identifier (e.g. CWE-595) in the search text input on the rules page and run the search.
- Rules must be accurately linked to their related CWE items. To see the CWE mapping for a SonarQube rule, consult the rule's See section at the bottom of the rule description.
- You must be able to identify the relevant CWE from an Issue. To do so in the SonarQube platform, consult the related rule.
- The product documentation must include a description of CWE and CWE Compatibility.
- The version of CWE supported must be listed. The SonarQube langauge plugins support version 2.8.
- In addition to searching rules by CWE id's, you can also search by the "cwe" rule tag.

To see which CWE items are covered for a language, consult the links below.

- C/C++
- Java
- Objective-C

SANS Top 25

The SANS Top 25 list is a collection of the 25-most dangerous errors listed in the CWE, as compiled by the SANS organization. The current SANS list is divided into three categories: Insecure Interaction Between Components, Risky Resource Management, and Porous Defenses.

The tags used for SANS correspond to its categories: sans-top25-insecure, sans-top25-risky, sans-top25-porous.

To find rules relating to SANS Top 25, you can perform a text search for the category, or the relevant CWE item, or perform a rule tag search.

OWASP Top 10

OWASP stands for Open Web Application Security Project. According to its site, it is:

> *A 501(c)(3) worldwide not-for-profit charitable organization focused on improving the security of software. Our mission is to make software security visible, so that individuals and organizations worldwide can make informed decisions about true software security risks.*

The OWASP Top 10 is a list of broad categories of weaknesses, each of which can map to many individual rules.

The tags used for OWASP correspond to the weakness categories: owasp-a1, owasp-a2, owasp-a3, owasp-a4, owasp-a5, owasp-a6, owasp-a7, owasp-a8, owasp-a9, owasp-a10.

To find rules relating to OWASP Top 10, you can perform a text search for the category, or perform a rule tag search.