MITRE CWE and CERT Secure Coding Standards

Robert C. Seacord
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213
rcs@cert.org

Robert Martin
MITRE Corporation
Bedford, MA 01730
ramartin@mitre.org

February 8, 2010

Common Weakness Enumeration

The Common Weakness Enumeration (CWE) is a unified, measurable set of software weaknesses that enables the effective discussion, description, selection, and use of software security tools and services that can find these weaknesses in source code and operational systems. The CWE also enables better understanding and management of software weaknesses related to architecture and design. It enumerates design and architectural weaknesses, as well as low-level coding and design errors.

CERT Secure Coding Standards

CERT is developing secure coding standards for commonly used programming languages such as C, C++, and Java through a broad-based community effort that includes members of the software development and software security communities. Well-documented and enforceable coding standards are essential to secure software development. Coding standards encourage programmers to follow a uniform set of rules and guidelines determined by the requirements of the project and organization, rather than by the programmer's familiarity or preference. Once established, these standards can be used as a metric to evaluate source code (using manual or automated processes) to determine compliance with the standard.

CERT secure coding standards include guidelines for avoiding coding and implementation errors, as well as low-level design errors.

CERT-CWE Relationship

The CWE and the CERT secure coding standards perform separate but mutually supportive roles. Simply stated, the CWE provides a comprehensive repository of know weaknesses, while CERT secure coding standards identify insecure coding constructs that, if present in code, could expose a weakness or vulnerability in the software. Not all weaknesses enumerated in the CWE are present in any particular secure coding standard, because not all weaknesses are present in each language and because CWE also includes high-level design issues. Not all CERT secure coding guidelines are mapped directly to weaknesses in the CWE, because some coding errors can manifest themselves in various ways that do not directly correlate to any given weakness. Both tools are necessary in evaluating the security and safety of software systems.

Figure 1 illustrates the software development aspect of the software assurance ecosystem supported by the CWE and the CERT secure coding standards. Related views could be constructed to show how these tools support acquisition, education, and research.

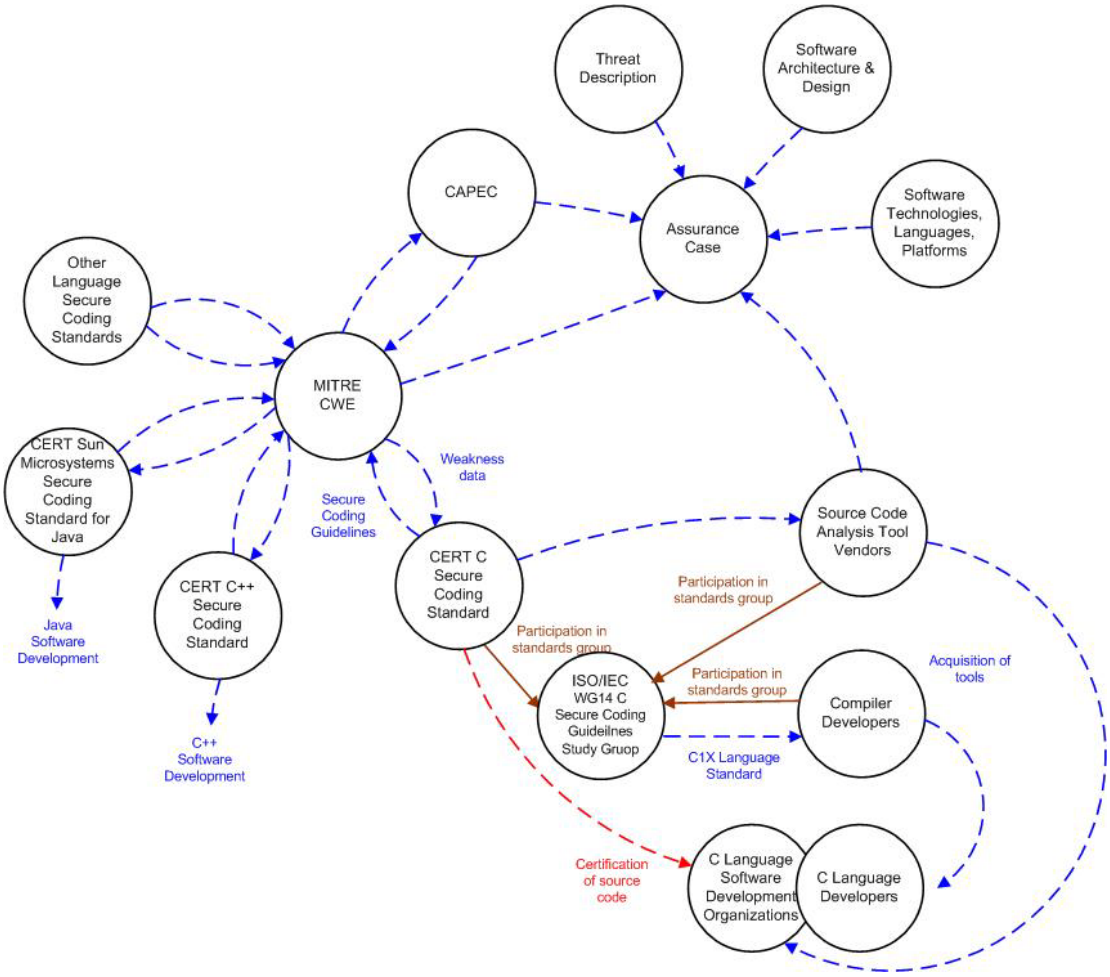

Figure 1: The CERT-CWE Landscape

CERT-CWE Mappings

The CWE contains multiple views, the most important of which are the full dictionary view, the development view, and the research view. The CWE-734 view enumerates weaknesses addressed by the *CERT C Secure Coding Standard* and includes 103 out of the 799 total CWEs. Developers can fully or partially prevent the weaknesses identified in CWE-734 if they adhere to the CERT coding standard. In addition, developers can use a CWE coverage graph to determine which weaknesses are not directly addressed by the standard. Making that determination can help identify and resolve remaining gaps in training, tool acquisition, and other approaches for reducing software weaknesses.

Guidelines in the *CERT C Secure Coding Standard* are cross-referenced with CWE entries. These cross-references are only created for guidelines which, if violated, directly contribute to the referenced weakness. Similar mappings will be created for other CERT coding standards once they are completed.