Common Weakness Enumeration

A Community-Developed List of Software Weakness Types

CWE and SANS Institute TOP 25 MOST DANGEROUS SOFTWARE ERRORS

CWE Overview

What Is CWE?

Targeted to developers and security practitioners, the Common Weakness Enumeration (CWE) is a formal list of software weakness types created to:

- Serve as a common language for describing software security weaknesses in architecture, design, or code.
- Serve as a standard measuring stick for software security tools targeting these weaknesses.
- Provide a common baseline standard for weakness identification, mitigation, and prevention efforts.

Introduction

Some Common Types of Software Weaknesses:

- Buffer Overflows, Format Strings, Etc.
- Structure and Validity Problems
- Common Special Element Manipulations
- Channel and Path Errors
- Handler Errors
- User Interface Errors
- Pathname Traversal and Equivalence Errors
- Authentication Errors
- Resource Management Errors
- Insufficient Verification of Data
- Code Evaluation and Injection
- Randomness and Predictability

Organizations want assurance that the software products they acquire and develop are free of known types of security flaws. Today, high-quality tools and services for finding security flaws and weaknesses in code are new and the question of which tool/service is appropriate/better for a particular job is hard to answer given the lack of structure and definition in the code assessment industry. CWE was created specifically to address these problems.

MITRE began working on the issue of categorizing software weaknesses

as early 1999 when it launched the CVE List. As part of the development of CVE MITRE's CVE Team developed a preliminary classification and categorization of vulnerabilities, attacks, faults, and other concepts to help define common software weaknesses. However, while sufficient for CVE those groupings are too rough to be used to identify and categorize the functionality offered within the offerings of the code security assessment industry. To support that type of usage additional fidelity and succinctness are needed, as are additional details and description for each of the different nodes and groupings such as the effects, behaviors, and implementation details, etc.

To do this MITRE took a first cut at revising the internal CVE category work for usage in the code assessment industry in 2005 as part of MITRE's participation in the U.S. Department of Homeland Security (DHS) sponsored National Institute of Technology (NIST) Software Assurance Metrics and Tool Evaluation (SAMATE) project. Our resulting document, entitled Preliminary List Of Vulnerability Examples for Researchers (PLOVER), was a working document that lists over 1,500 diverse, real-world examples of vulnerabilities, identified by their CVE name. The vulnerabilities in PLOVER are organized within a detailed conceptual framework that currently enumerates 290 individual types of software weaknesses, idiosyncrasies, faults, and flaws, with a large number of real-world vulnerability examples for each. PLOVER represented the first attempt at a truly bottom-up effort to take real-world observed faults and flaws that do exist in code, abstract them and group them into common classes representing more general potential vulnerabilities that could exist in code, and then finally to organize them in an appropriate relative structure so as to make them accessible and useful to a diverse set of audiences for a diverse set of purposes.

The next step after PLOVER was to establish acceptable definitions and descriptions of these common weaknesses by the community under the NIST SAMATE project, which led to the creation of the "Common Weakness Enumeration" list and associated classification taxonomy. The CWE List now serves as a mechanism for describing code vulnerability assessment capabilities in terms of their coverage of the different CWEs.

Why CWE

The Challenge

Software acquirers want assurance that the software products they are obtaining are reviewed for known types of security flaws, and the

acquisition groups in large government and private organizations are moving forward to use these types of reviews as part of future contracts. However, the tools and services that can be used for this type of review are new at best and there are no nomenclature, taxonomies, or standards to define the capabilities and coverage of these tools and services. This makes it difficult to comparatively decide which tool/service is best suited for a particular job. What is needed is a standard list and classification of software security weaknesses to serve as a unifying language of discourse and a measuring stick for tools and services.

The Solution

CWE is a community-developed formal list of common software weaknesses. It serves as a common language for describing software security weaknesses, a standard measuring stick for software security tools targeting these vulnerabilities, and as a baseline standard for weakness identification, mitigation, and prevention efforts. Leveraging the diverse thinking on this topic from academia, the commercial sector, and government, CWE unites the most valuable breadth and depth of content and structure to serve as a unified standard. Our objective is to help shape and mature the code security assessment industry and also dramatically accelerate the use and utility of software assurance capabilities for organizations in reviewing the software systems they acquire or develop.

As discussed above, we leveraged MITRE's PLOVER effort as a starting point for the creation of the formal Common Weakness Enumeration. Not only does CWE encompass a large portion of the CVE List's 15,000 CVE names, but it also includes detail, breadth and classification structure from a diverse set of other industry and academic sources and examples including the McGraw/Fortify "Kingdoms" taxonomy; Howard, LeBlanc & Viega's 19 Deadly Sins; and Secure Software's CLASP project; among others.

CWE's definitions and descriptions support the finding of these common types of software security flaws in code prior to fielding. This means both users and developers of software assurance tools and services now have a mechanism for describing each of the industry's software security flaw code assessment capabilities in terms of their coverage of the different CWEs. If necessary, CWE can also be scoped to specific languages, frameworks, platforms, and machine architectures.

CWE Compatibility

Beyond the creation of the CWE List and associated classification tree for the reasons described above, a further end-goal of this effort is to take the findings and results of this work and use them as the foundation of a CWE Compatibility Program that can be directly used by organizations in their selection and evaluation of tools and/or services for assessing their acquired software for known types of weaknesses.

Related Efforts

Several additional efforts are currently ongoing targeted at resolving some of the other shortcomings in software assurance, including NIST's SAMATE project as mentioned above; the U.S. Department of Defense (DOD)-sponsored Code Assessment Methodology Project (CAMP) that is part of the Protection of Vital Data (POVD) effort being conducted by Concurrent Technologies Corporation (CTC); the Object Management Group (OMG) Software Assurance (SwA) Special Interest Group (SIG); and the work of the International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) Joint Technical Committee 1/Subcommittee 22 (JTC 1/SC22) Other Working Group (OWG): Vulnerabilities working group tasked with ISO project 22.24772; among others.

While these efforts are well placed, timely in their objectives, and will surely yield high value in the end, they would all benefit from a common description of the underlying security weaknesses in software that they are targeted to resolve. Without such a common description—that is, CWE—these efforts cannot move forward in a meaningful fashion or be aligned and integrated with each other to provide strategic value. Most past efforts at developing such a capability have been limited by a very narrow technical domain focus or have largely focused on high-level theories, taxonomies, or schemes that do not reach the level of detail or variety of security issues that are found in today's software products.

CWE is intended to address all of these concerns and will only enhance the usefulness of SAMATE, CAMP, OMG, and other such efforts.

Contact Us

To discuss the CWE effort in general, the impacts and transition opportunities noted above, or any other questions or concerns, please email us at cwe@mitre.org.

Privacy policy
Terms of use
Contact us