



Common Weakness Enumeration

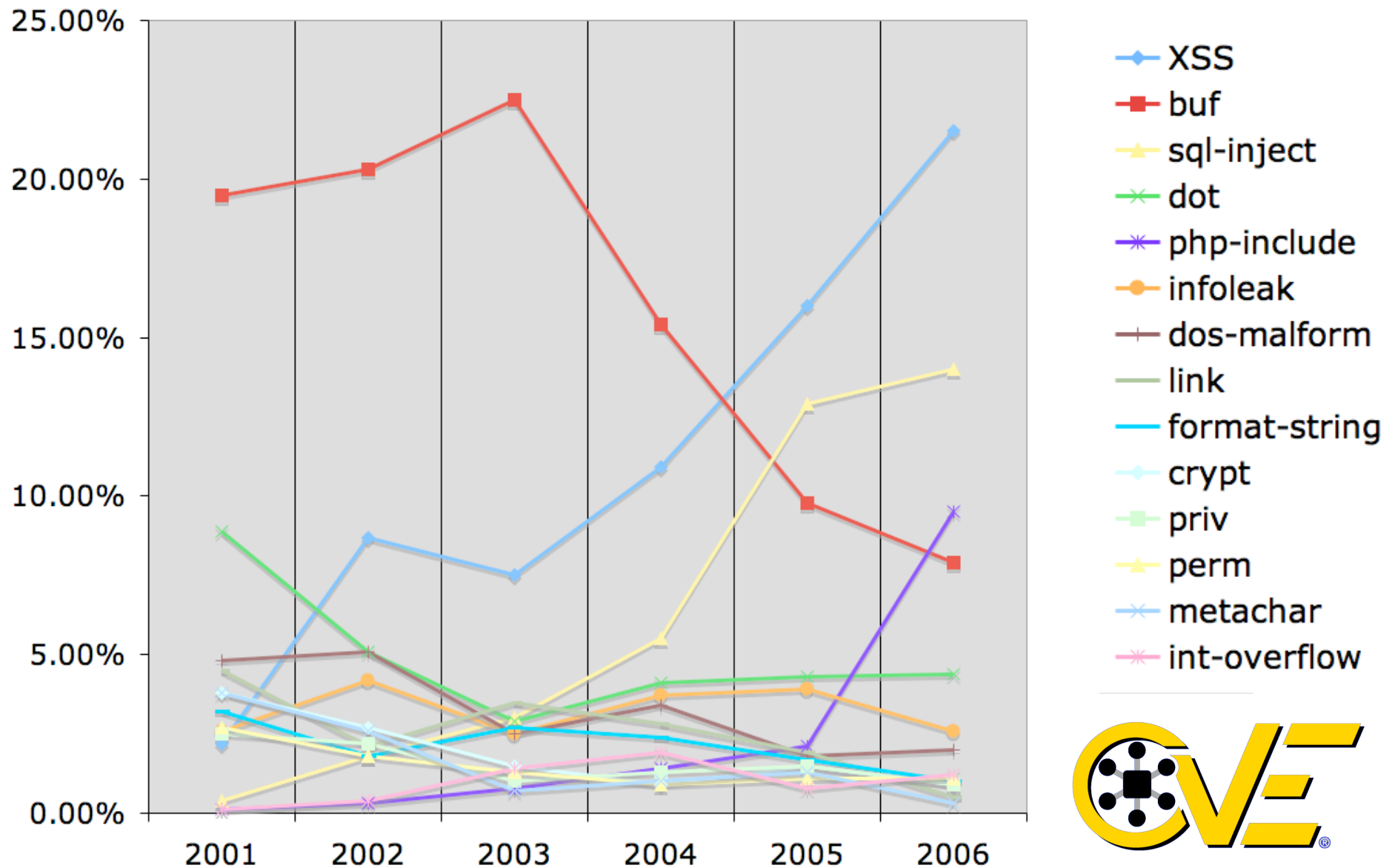
9 May 2007

Robert A. Martin



MITRE

Vulnerability Type Trends: A Look at the CVE List (2001 - 2006)



Removing and Preventing the Vulnerabilities Requires More Specific Definitions...CWEs

◆ XSS
■ buf
★ sql-inject
✕ dot
✱ php-include
● infoleak
— dos-malform
— link
— format-string
— crypt
— priv
★ perm
— metachar
✱ int-overflow

Cross-site scripting (XSS) (79)

- Basic XSS (80)
- XSS in error pages (81)
- Script in IMG tags (82)
- XSS using Script in Attributes (83)
- XSS using Script Via Encoded URI Schemes (84)
- Doubled character XSS manipulations, e.g. '<<script' (85)
- Invalid Characters in Identifiers (86)
- Alternate XSS syntax (87)
- Mobile Code: Invoking untrusted mobile code (494)

Buffer Errors (119)

- Unbounded Transfer (classic overflow) (120)
- Write-what-where condition (123)
- Boundary beginning violation ('buffer underwrite') (124)
- Out-of-bounds Read (125)
- Wrap-around error (128)
- Unchecked array indexing (129)
- Length Parameter Inconsistency (130)
- Other length calculation error (131)
- Miscalculated null termination (132)
- String Errors (133)
- Often Misused: Path Manipulation (249)

Relative Path Traversal (22)

- Path Issue - dot dot slash - '../filedir' (24)
- Path Issue - leading dot dot slash - '/../filedir' (25)
- Path Issue - leading directory dot dot slash - '/directory/../filename' (26)
- Path Issue - directory doubled dot dot slash - 'directory/../../filename' (27)
- Path Issue - dot dot backslash - '..\filename' (28)
- Path Issue - leading dot dot backslash - '\..\filename' (29)
- Path Issue - leading directory dot dot backslash - 'directory..\filename' (30)
- Path Issue - directory doubled dot dot backslash - 'directory\..\filename' (31)
- Path Issue - triple dot - '...' (32)
- Path Issue - multiple dot - '....' (33)
- Path Issue - doubled dot dot slash - '.../' (34)
- Path Issue - doubled triple dot slash - '.../...' (35)

Current Community Contributing to the Common Weakness Enumeration

- AppSIC
- Booz Allen Hamilton Inc.
- Cenxic
- CERIAs/Purdue University
- CERT/CC
- Cigital
- CodescanLabs
- Core Security
- Coverity
- Fortify
- Gramma Tech
- IBM Interoperability Clearing House
- JHU/APL
- JMU
- Kestrel Technology
- KDM Analytics
- Klocwork
- McAfee/Foundstone
- Microsoft
- MIT Lincoln Labs
- MITRE
- North Carolina State University
- NIST
- NSA
- Oracle
- Ounce Labs
- OWASP
- Palamida
- Parasoft
- PolySpace Technologies
- proServices Corporation
- SANS Institute
- SecurityInnovation
- Secure Software
- Security University
- Semantic Designs
- SofCheck
- SPI Dynamics
- SureLogic, Inc.
- UNISYS
- VERACODE
- Watchfire
- WASC
- Whitehat Security, Inc.
- Tim Newsham



To join send e-mail to cwe@mitre.org

Using A Unilateral NDA with MITRE to Bring in Info

Purpose:

- Sharing the proprietary/company confidential information contained in the underlying Knowledge Repository of the Knowledge Owner's Capability for the sole purpose of establishing a public Common Weakness Enumeration (CWE) dictionary that can be used by vendors, customers, and researchers to describe software, design, and architecture related weaknesses that have security ramifications.
- The individual contributions from numerous organizations, based on their proprietary/company-confidential information, will be combined into a consolidated collection of weakness descriptions and definitions with the resultant collection being shared publicly.
- The consolidated collection of knowledge about weaknesses in software, design, and architecture will make no reference to the source of the information used to describe, define, and explain the individual weaknesses.



CWE-79 Cross-site scripting (XSS)

[cwe.mitre.org/data/definition/79.html]

Individual CWE Dictionary Definition (Draft 6)

Search by ID

Cross-site scripting (XSS)

CWE ID	79
Description	Cross-site scripting weakness occurs when dynamically generated web pages display input, such as login information, that is not properly validated, allowing an attacker to embed malicious scripts into the generated page and then execute the script on the machine of any user that views the site. If successful, Cross-site scripting vulnerabilities can be exploited to manipulate or steal cookies, create requests that can be mistaken for those of a valid user, compromise confidential information, or execute malicious code on the end user systems for a variety of nefarious purposes.
Alternate Terms	"CSS" was once used as the acronym for this problem, but this can cause confusion with the "Cascading Style Sheets," so its use has declined significantly, and its use is discouraged by the author.
Likelihood of Exploit	High to Very High
Weakness Ordinality	Resultant (Weakness is typically related to the presence of some other Weaknesses)
Causal Nature	Explicit (This is an explicit weakness resulting from behavior of the developer)
Common Consequences	Confidentiality: The most common attack payload is the theft of confidential information stored in user cookies. Access control: In some circumstances it may be possible to gain access to resources when cross-site scripting is combined with other weaknesses.
Potential Mitigations	Carefully check each input parameter against a list of specific characters and format allowed. All input should be validated, but all data in the request should be displayed as-is, so forth. A common mistake that leads to XSS is to escape characters expected to be redisplayed by the site. We do not recommend escaping application server or the application that the user is currently reflected may be used by a future attacker. This involves "HTML Entity Encoding" all non-whitespace characters that the user and is now being written to the request. With Struts, you should write all data from the request to the response. Additionally, to help mitigate XSS attacks as well as SQL injection, HttpOnly. In browsers that support the HttpOnly flag, this prevents the user's session cookie from being accessed by a XSS attack.
Demonstrative Examples	The following JSP code segment reads an error message from the user. JSP Example: <pre><% String eid = request.getParameter("eid")</pre>

References

M. Howard and D. LeBlanc. "Writing Secure Code". 2nd Edition. Microsoft. 2003.

Node Relationships

ChildOf - [Injection](#) (74)
ResultsIn - [Mobile Code: Invoking untrusted mobile code](#) (494)
ParentOf - [Basic XSS](#) (80)
ParentOf - [XSS in error pages](#) (81)
ParentOf - [Script in IMG tags](#) (82)
ParentOf - [XSS using Script in Attributes](#) (83)
ParentOf - [XSS using Script Via Encoded URI Schemes](#) (84)
ParentOf - [Doubled character XSS manipulations, e.g. '<<script'](#) (85)
ParentOf - [Invalid Characters in Identifiers](#) (86)
ParentOf - [Alternate XSS syntax](#) (87)

Source Taxonomies

PLOVER - Cross-site scripting (XSS)
7 Pernicious Kingdoms - Cross-site Scripting
CLASP - Cross-site scripting

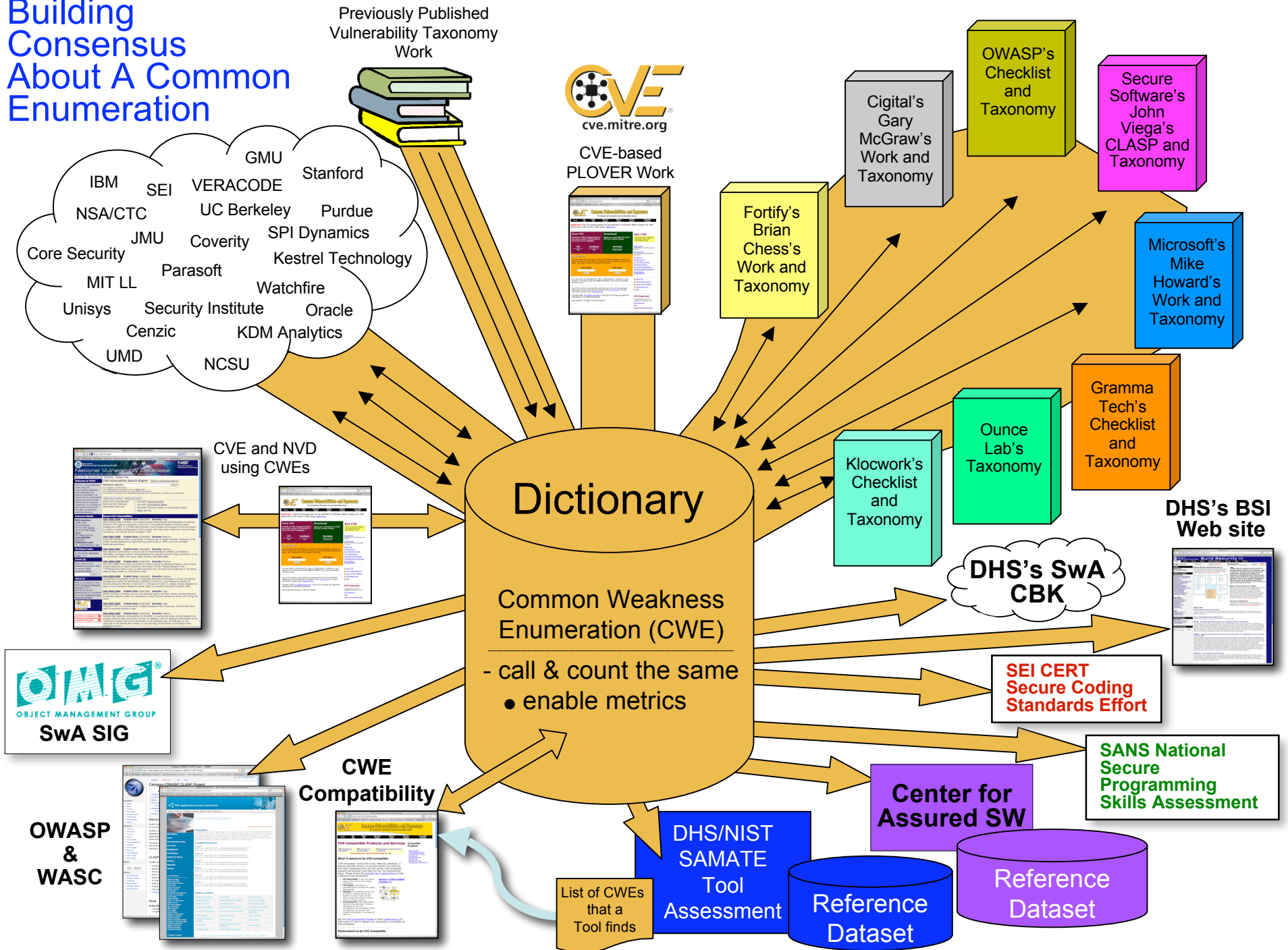
Applicable Platforms

C
C++
Java
.NET
SOAP

[BACK TO TOP](#)

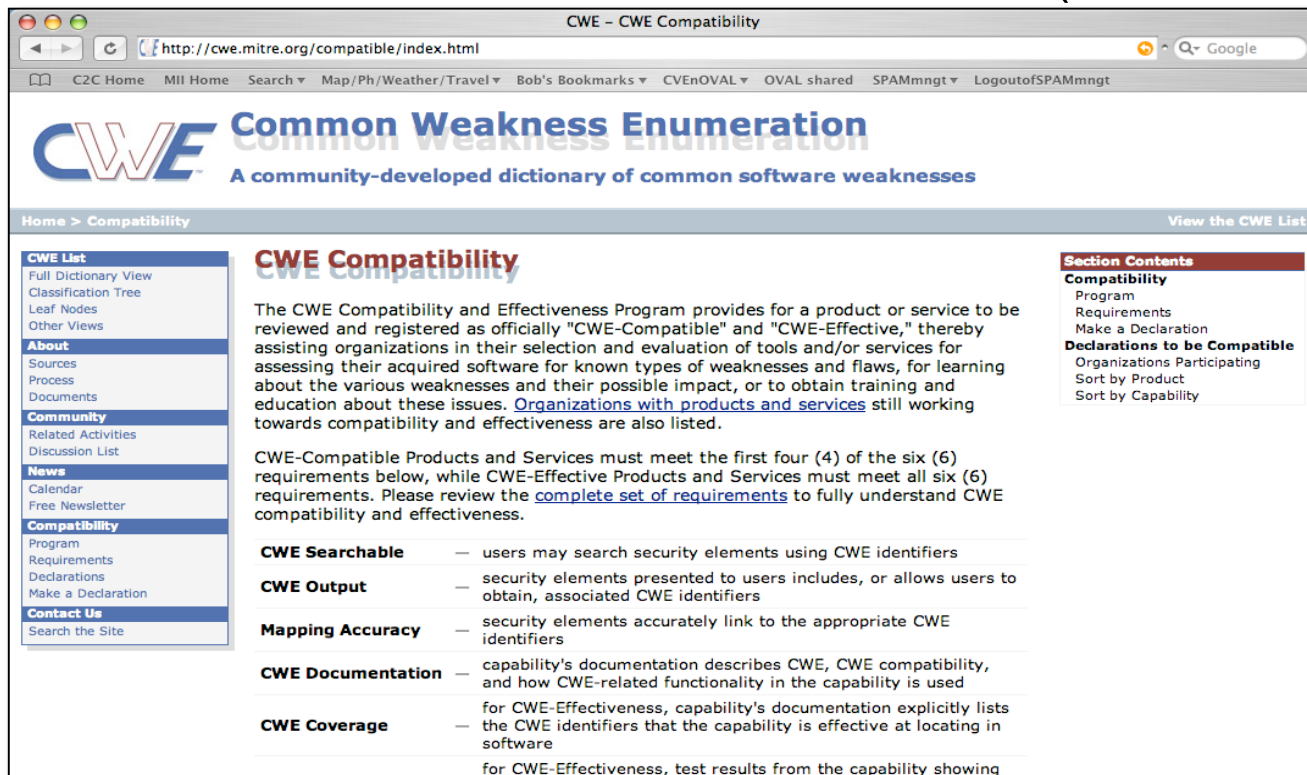
Making Security Measurable™

Building Consensus About A Common Enumeration



CWE Compatibility & Effectiveness Program

(launched Feb 2007)



Organizations Participating

All organizations participating in the CWE Compatibility and Effectiveness Program are listed below, including those with CWE-Compatible Products and Services and those with Declarations to Be CWE-Compatible.

Products are listed alphabetically by organization name:

cwe.mitre.org/compatible/

TOTALS

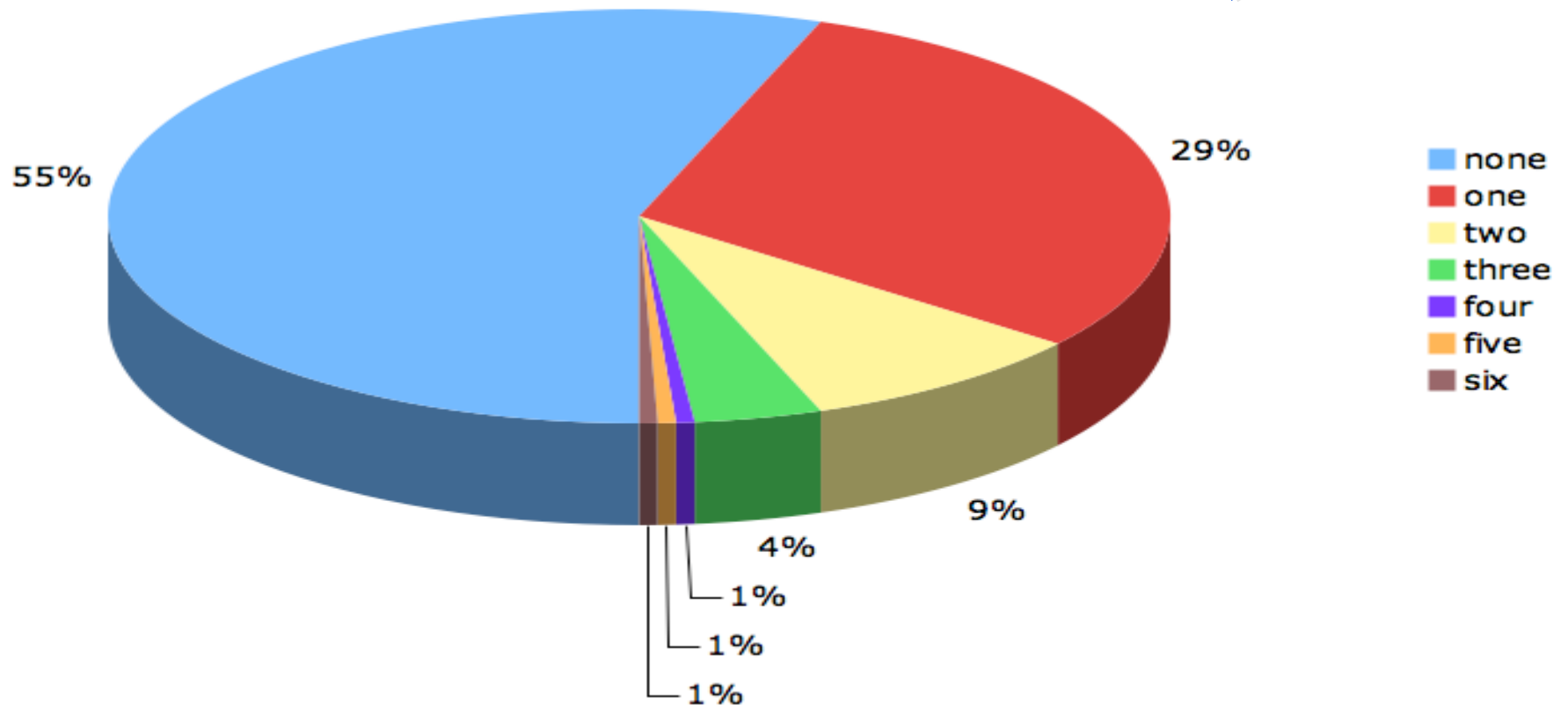
Organizations Participating: 11
Products & Services: 21

Updated: February 05, 2007

Coverage of CWE

CWE

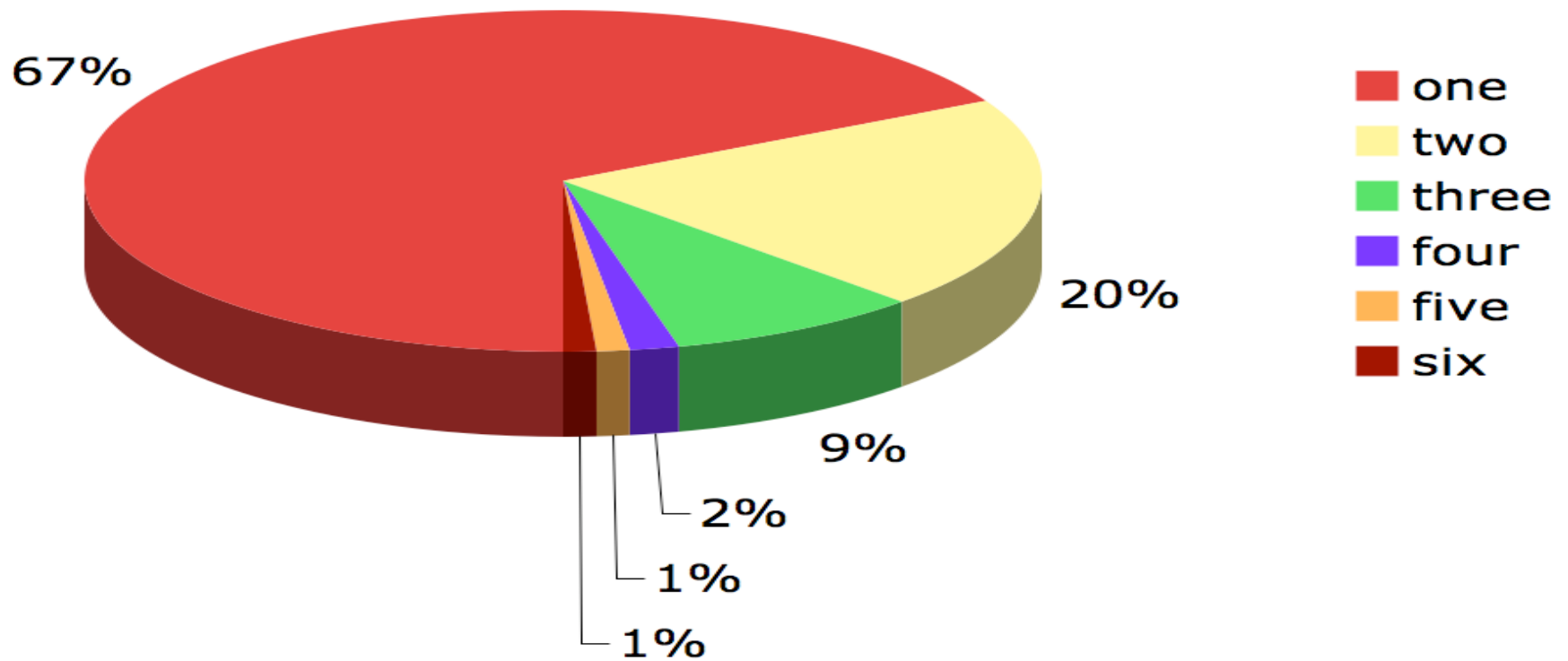
Draft



Covered CWEs - By Number of Tools

Covered CWEs

Draft



What are the Building Blocks of a “Security Architecture”

- Standard ways for enumerating “things we care about”
- Languages for encoding high fidelity information about how to find the “things we care about”
- Repositories of content in languages for use in communities or individual organizations
- Adoption/branding and vetting programs to encourage adoption by tools and services

The Building Blocks Are:

- Enumerations
 - Catalog the fundamental entities in IA, Cyber Security, and Software Assurance
 - Vulnerabilities (CVE), misconfigurations (CCE), software packages (CPE), malware (CME), attack patterns (CAPEC), weaknesses in code/design/architecture (CWE)
- Languages/Formats
 - Support the creation of machine-readable state assertions, assessment results, and messages
 - Configuration/vulnerability/patch/asset patterns (XCCDF & OVAL), software security patterns (SBVR), event patterns (CEE), malware patterns (MAEC), risk of a vulnerability (CVSS), information messages (CAIF & *DEF)
- Knowledge Repositories
 - Packages of assertions supporting a specific application
 - Vulnerability advisories & alerts, (US-CERT Advisories/IAVAs), configuration assessment (NIST Checklists, CIS Benchmarks, NSA Configuration Guides, DISA STIGS), asset inventory (NIST/DHS NVD), code assessment & certification (NIST SAMATE, DoD DIACAP & eMASS)

Tools

- Interpret IA, Cyber Security, and SwA content in context of enterprise network
- Methods for assessing compliance to languages, formats, and enumerations





Making Security Measurable™

A Collection of Information Security Community Standardization Activities and Initiatives

[Home](#) | [Current Collection](#) | [Feedback Requested](#)

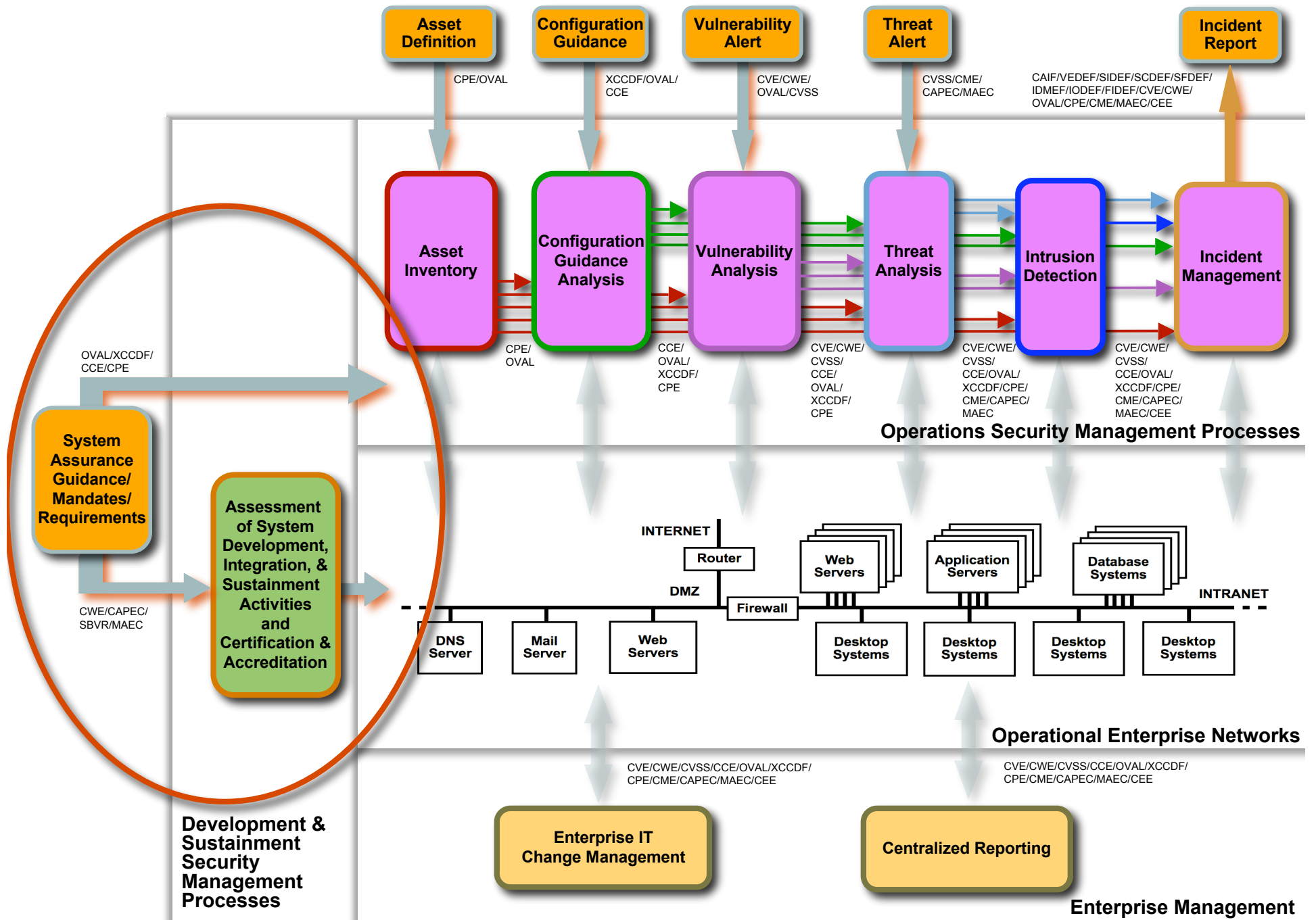
Measurable security pertains at a minimum to the following areas:

- Vulnerability Management
- System Assessment
- Configuration Management
- Malware Management
- System Management
- Intrusion Detection
- Patch Management
- Incident Management
- Asset Management

Enumerations	Languages	Repositories
 Common Vulnerabilities and Exposures (CVE®) - common vulnerability identifiers	 Open Vulnerability and Assessment Language (OVAL™) - standard for determining vulnerability and configuration issues	 OVAL Repository - community-developed OVAL Vulnerability, Compliance, Inventory, and Patch Definitions
 Common Weakness Enumeration (CWE™) - list of software weakness types	Extensible Configuration Checklist Description Format (XCCDF) - specification language for uniform expression of security checklists, benchmarks, and other configuration guidance	National Vulnerability Database (NVD) - U.S. vulnerability database based on CVE that integrates all publicly available vulnerability resources and references
 Common Malware Enumeration (CME™) - common virus identifiers	Common Vulnerability Scoring System (CVSS) - open standard that conveys vulnerability severity and helps determine urgency and priority of response	NIST Security Content Automation Program (SCAP) - security content for automating technical control compliance activities, vulnerability checking, and security measurement
 Common Configuration Enumeration (CCE™) - common security configuration identifiers	Common Announcement Interchange Format (CAIF) - XML-based format created to store and exchange security announcements in a normalized way	Red Hat Repository - OVAL Patch Definitions corresponding to Red Hat Errata security advisories
 Common Platform Enumeration (CPE™) - common platform identifiers	OMG Semantics of Business Vocabulary and Business Rules (SBVR) - language for interchange of business vocabularies and rules among organizations and software tools	Center for Internet Security (CIS) Benchmarks - best-practice security configurations accepted for compliance with FISMA, the ISO standard, GLB, SOx, HIPAA, and FIRPA, and other regulatory requirements for information security
SANS Top Twenty - SANS/FBI consensus list of the Twenty Most Critical Internet Security Vulnerabilities that uses CVE-IDs to identify the issues		DISA Security Technical Implementation Guides (STIGS) - U.S. Defense Information Systems Agency's (DISA) STIGS are configuration standards for DOD information assurance and information assurance-enabled devices and systems
OWASP Top Ten - ten most critical Web application security flaws		
WASC Web Security Threat Classification - list of Web security threats		

View the current collection of organizations, activities, and initiatives.

Knowledge Repositories

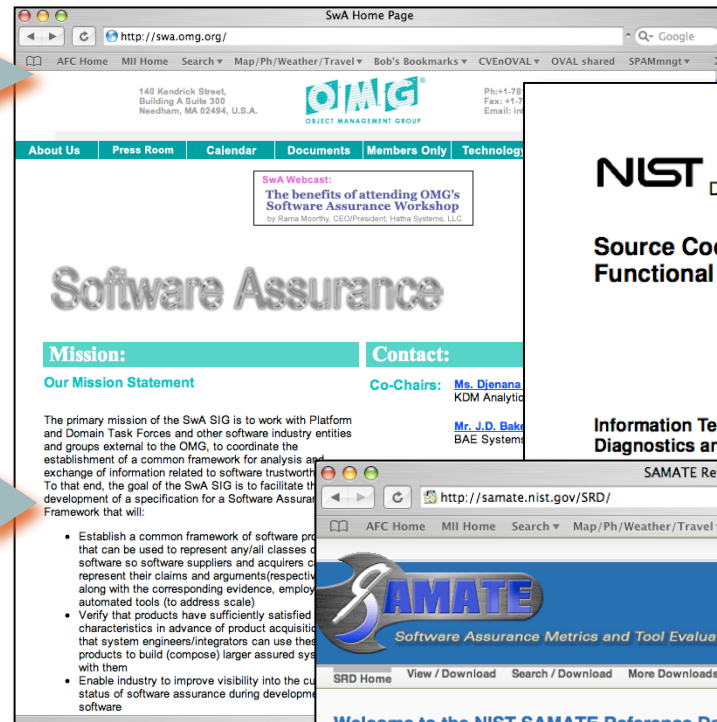


OVAL/XCCDF/
CCE/CPE

**System
Assurance
Guidance/
Mandates/
Requirements**

CWE/CAPEC/
SBVR

**Certification &
Assessment
of System
Development,
Integration,
&
Sustainment
Activities**



NIST Draft Special Publication 500-268

**Source Code Security Analysis Tool
Functional Specification Version 1.0**

**Information Technology Laboratory (ITL), Software
Diagnostics and Conformance Testing Division**

