Public/Private Collaboration Efforts for Software Supply Chain Risk Management

Next SwA Working Group Sessions 14-16 Dec 2010 at MITRE, McLean, VA
Software Assurance: Enabling Software Resilience and Mitigating Supply Chain Risk

Dec 6, 2010

Joe Jarzombek, PMP, CSSLP
Director for Software Assurance
National Cyber Security Division
Office of the Assistant Secretary for Cybersecurity and Communications
Today Everything’s Connected

Your System is attackable…

When this Other System gets subverted through an un-patched vulnerability, a mis-configuration, or an application weakness…
Cyber Infrastructure represents the convergence of information technology and communications systems, is inherent to nearly every aspect of modern life.
Interdependencies Between Physical & Cyber Infrastructures: Requires Convergence of Safety, Security and Dependability

-- Need for secure software applications
Security is a Requisite Quality Attribute: Vulnerable Software Enables Exploitation

- Rather than attempt to break or defeat network or system security, hackers are opting to target application software to circumvent security controls.
  - 75% of hacks occurred at application level
    - “90% of software attacks were aimed at application layer” (Gartner & Symantec, June 2006)
  - most exploitable software vulnerabilities are attributable to non-secure coding practices (and not identified in testing).
- Functional correctness must be exhibited even when software is subjected to abnormal and hostile conditions.

In an era riddled with asymmetric cyber attacks, claims about system reliability, integrity & safety must include provisions for built-in security of the enabling software.
Critical Considerations

- Software is the core constituent of modern products and services – it enables functionality and business operations.

- Dramatic increase in mission risk due to increasing:
  - Software dependence and system interdependence (weakest link syndrome)
  - Software Size & Complexity (obscures intent and precludes exhaustive test)
  - Outsourcing and use of un-vetted software supply chain (COTS & custom)
  - Attack sophistication (easing exploitation)
  - Reuse (unintended consequences increasing number of vulnerable targets)
  - Number of vulnerabilities & incidents with threats targeting software
  - Risk of Asymmetric Attack and Threats

- Increasing awareness and concern
**Software Assurance Addresses Exploitable Software:** Outcomes of non-secure practices and/or malicious intent

Exploitation potential of vulnerability is independent of “intent”

*Intentional vulnerabilities: spyware & malicious logic deliberately imbedded (might not be considered defects)*

‘High quality’ can reduce security flaws attributable to defects; yet traditional S/W quality assurance does not address intentional malicious behavior in software

Note: Chart is not to scale – notional representation -- for discussions
In the digital age, sovereignty is demarcated not by territorial frontiers but by supply chains.

– Dan Geer, CISO In-Q-Tel

Enterprise Risk Management and Governance are security motivators.

Acquisition could be considered the beginning of the lifecycle; more than development.

“In the digital age, sovereignty is demarcated not by territorial frontiers but by supply chains.”

– Dan Geer, CISO In-Q-Tel

Software Assurance provides a focus for:
-- Secure Software Components,
-- Security in the Software Life Cycle,
-- Software Security in Services, and
-- Software Supply Chain Risk Management.
Security-Enhanced Capabilities: Mitigating Risks to the Enterprise

- With today’s global software supply chain, Software Engineering, Quality Assurance, Testing and Project Management must explicitly address security risks posed by exploitable software.
  - Traditional processes do not explicitly address software-related security risks that can be passed from projects to using organizations.

- Mitigating Supply Chain Risks requires an understanding and management of Suppliers’ Capabilities, Products and Services
  - Enterprise risks stemming from supply chain are influenced by suppliers and acquisition projects (including procurement, SwEng, QA, & testing).
  - IT/Software Assurance processes/practices span development/acquisition.
  - Derived (non-explicit) security requirements should be elicited/considered.

- More comprehensive diagnostic capabilities and standards are needed to support processes and provide transparency for more informed decision-making for mitigating risks to the enterprise

Free resources are available to assist personnel in security-enhancing contracting, outsourcing and development activities (see https://buildsecurityin.us-cert.gov)
Context for Enterprise IT Security and Layered Assurance

The environment consists of a changing set of conditions, Policies, and other factors unknown at the time of implementation but realized during use or consumption.

The system is an arrangement of products fulfilling a need, Constrains the environment of each product.

The product is the unit of purchase, And frequently has multiple uses.

Implementation of an IA algorithm in a product

“feature function”

“product”

“system”

“environment”

Domain of FIPS

Domain of Common Criteria evaluated products

Domain of Certification and Accreditation (all products, interfaces, configuration and other Issues)
Assurance Challenges in Mitigating Software Supply Chain Risks

- Complexity hampers our ability to determine and predict code behavior; so any “assurance” claims for security/safety-critical applications are limited.

- Without adequate diagnostic capabilities and commonly recognized standards from which to benchmark process capabilities and assert claims about the assurance of products, systems and services, the “providence and pedigree of supply chain actors” become a more dominant consideration for security/safety-critical applications:
  - Enterprises and Consumers lack requisite transparency for more informed decision-making for mitigating risks;
  - Favoring domestic suppliers does not necessarily address ‘assurance’ in terms of capabilities to deliver secure/safe components, systems or software-reliant services.

- Several needs arise:
  - Need internationally recognized standards to support processes and provide transparency for more informed decision-making for mitigating enterprise risks.
  - Need ‘Assurance’ to be explicitly addressed in standards & capability benchmarking models for organizations involved with security/safety-critical applications.
  - Need more comprehensive diagnostic capabilities to provide sufficient evidence that “code behavior” can be well understood to not possess exploitable or malicious constructs.
  - Need rating schemes for software products and supplier capabilities
DHS Software Assurance Program Overview

Program established in response to the National Strategy to Secure Cyberspace - Action/Recommendation 2-14:

“DHS will facilitate a national public-private effort to promulgate best practices and methodologies that promote integrity, security, and reliability in software code development, including processes and procedures that diminish the possibilities of erroneous code, malicious code, or trap doors that could be introduced during development.”

- DHS Program goals promote the **security and resilience** of software across the development, acquisition, and operational life cycle

- DHS Software Assurance (SwA) program is scoped to address:
  - **Trustworthiness** - No exploitable vulnerabilities or malicious logic exist in the software, either intentionally or unintentionally inserted,
  - **Dependability (Correct and Predictable Execution)** - Justifiable confidence that software, when executed, functions as intended,
  - **Survivability** - If compromised, damage to the software will be minimized; it will recover quickly to an acceptable level of operating capacity; it’s ‘rugged’;
  - **Conformance** – Planned, systematic set of multi-disciplinary activities that ensure processes/products conform to requirements, standards/procedures.

See Wikipedia.org for “Software Assurance” - CNSS Instruction No. 4009, "National Information Assurance Glossary," Revised 2006, defines Software Assurance as: "the level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at anytime during its lifecycle, and that the software functions in the intended manner".
DHS NCSD Software Assurance (SwA) Program

Through public-private collaboration promotes security and resilience of software throughout the lifecycle; focused on reducing exploitable software weaknesses and addressing means to improve capabilities that routinely develop, acquire, and deploy resilient software products. Collaboratively advancing software-relevant rating schemes

• Serves as a focal point for interagency public-private collaboration to enhance development and acquisition processes and capability benchmarking to address software security needs.
  – Hosts interagency Software Assurance Forums, Working Groups and training to provide public-private collaboration in advancing software security and providing publicly available resources.
  – Provides collaboratively developed, peer-reviewed information resources on Software Assurance, via journals, guides & on-line resources suitable for use in education, training, and process improvement.
  – Provides input and criteria for leveraging international standards and maturity models used for process improvement and capability benchmarking of software suppliers and acquisition organizations.

• Enables software security automation and measurement capabilities through use of common indexing and reporting capabilities for malware, exploitable software weaknesses, and common attacks which target software.
  – Collaborates with the National Institute of Standards and Technology, international standards organizations, and tool vendors to create standards, metrics and certification mechanisms from which tools can be qualified for software security verification.
  – Manages programs for Malware Attribute Enumeration Classification (MAEC), Common Weakness Enumeration (CWE), and Common Attack Pattern Enumeration and Classification (CAPEC).
  – Manages programs for Common Vulnerabilities & Exposures (CVE) and Open Vulnerability & Assessment Language (OVAL) that provide information feeds for Security Content Automation Protocol (SCAP), vulnerability databases, and security/threat alerts from many organizations.
Software Assurance “End State” Objectives…

- Government, in collaboration with industry / academia, raised expectations for product assurance with requisite levels of integrity and security:
  - Helped advance more comprehensive software assurance diagnostic capabilities to mitigate risks stemming from exploitable vulnerabilities and weaknesses;
  - Collaboratively advanced use of software security measurement & benchmarking schemes;
  - Promoted use of methodologies and tools that enabled security to be part of normal business.

- Acquisition managers & users factored risks posed by the software supply chain as part of the trade-space in risk mitigation efforts:
  - Information on suppliers’ process capabilities (business practices) would be used to determine security risks posed by the suppliers’ products and services to the acquisition project and to the operations enabled by the software.
  - Information about evaluated products would be available, along with responsive provisions for discovering exploitable vulnerabilities, and products would be securely configured in use.

- Suppliers delivered quality products with requisite integrity and made assurance claims about the IT/software safety, security and dependability:
  - Relevant standards would be used from which to base business practices & make claims;
  - Qualified tools used in software lifecycle enabled developers/testers to mitigate security risks;
  - Standards and qualified tools would be used to certify software by independent third parties;
  - IT/software workforce had requisite knowledge/skills for developing secure, quality products.

...Enabling Software Supply Chain Transparency
Software Assurance Forum & Working Groups*

... encourage the production, evaluation and acquisition of better quality and more secure software through targeting

<table>
<thead>
<tr>
<th>People</th>
<th>Processes</th>
<th>Technology</th>
<th>Acquisition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developers and users education &amp; training</td>
<td>Sound practices, standards, &amp; practical guidelines for secure software development</td>
<td>Security test criteria, diagnostic tools, common enumerations, SwA R&amp;D, and SwA measurement</td>
<td>Software security improvements through due-diligence questions, specs and guidelines for acquisitions/outsourcing</td>
</tr>
</tbody>
</table>

### Products and Contributions

**Build Security In** - https://buildsecurityin.us-cert.gov and SwA community resources & info clearinghouse

SwA Common Body of Knowledge (CBK) & Glossary Organization of SwSys Security Principles/Guidelines

SwA Developers' Guide on Security-Enhancing SDLC

Software Security Assurance State of the Art Report

Systems Assurance Guide (via DoD and NDIA)


Practical Measurement Framework for SwA/InfoSec

Making the Business Case for Software Assurance

SwA Metrics & Tool Evaluation (with NIST)

SwA Ecosystem w/ DoD, NSA, NIST, OMG & TOG

NIST Special Pub 500 Series on SwA Tools

Common Weakness Enumeration (CWE) dictionary

Common Attack Pattern Enumeration (CAPEC)

SwA in Acquisition: Mitigating Risks to Enterprise

Software Project Management for SwA SOAR

* SwA Forum is part of Cross-Sector Cyber Security Working Group (CSCSWG) established under auspices of the Critical Infrastructure Partnership Advisory Council (CIPAC) that provides legal framework for participation.
“Supply chain introduces risks to American society that relies on Federal Government for essential information and services.”

30 Sep 2005 changes to Federal Acquisition Regulation (FAR) focus on IT Security

Focuses on the role of contractors in security as Federal agencies outsource various IT functions.

Enterprise Processes for deploying capabilities: Increasingly Distributed and Complex

New Considerations for Quality & Security

Development Process

Company Employees

Contractors

Enterprise Employees

Foreign Contractors

Foreign Sub-Contractors

US Dev. Center A

Open Source

3rd Party Libraries

Offshore

US Dev. Center B

Procurement Process

Agency/Enterprise

ISV Employees

Foreign Contractor

License 3rd Party Libraries

Open Source

Purchased

Outsource Partner A

Outsource Partners

Outsourcer Employees

Indian Contractor

Chinese Contractor

License 3rd Party Libraries

Source: SwA WG Panel presentations, 2008
Risk Management (Enterprise <=> Project): Shared Processes & Practices // Different Focuses

- **Enterprise-Level:**
  - Regulatory compliance
  - Changing threat environment
  - Business Case

- **Program/Project-Level:**
  - Cost
  - Schedule
  - Performance

Software Supply Chain Risk Management traverses enterprise and program/project interests
The New Issue is Virtual Security

In addition to physical security, we now worry about cyber risks:

- Theft of intellectual property
- Fake or counterfeit products
- Import/export of strong encryption
- IT/software with deliberately embedded malicious functionality
  - Logic bombs and self-modifying code
  - Other “added features” like key loggers
  - Deliberately hidden back doors for unauthorized remote access
- Exploitable IT/software from suppliers with poor security practices
  - Failure to use manufacturing processes/capabilities to design and build secure products (no malicious intent) in delivering exploitable products
  - Resuppliers (VARs, integrators, and service providers) often lack incentives and capabilities to adequately check content of sub-contracted and outsourced IT/software products

IT/software security laws, policies, & standards are immature

One who knows the enemy and knows himself will not be endangered in a hundred engagements.

One who does not know the enemy but knows himself will sometimes be victorious; sometimes meet with defeat.

One who knows neither the enemy nor himself will invariably be defeated in every engagement.

- The Art of War, Sun Tzu

An appropriate defense can only be established if one knows its weaknesses and how it will be attacked; thus controlling attack surface/vectors

- Software Assurance Forum, Joe Jarzombek
We are engaged with many parts of the Community for Software Assurance-related standardization
ISO/IEC JTC1

  - This technical report was reviewed and approved by the project editor, then published in early October.
  - As published, the document includes language-independent summaries of nearly 70 classes of vulnerabilities.
  - The working group is already drafting the 2\textsuperscript{nd} Edition of the report which will add information specific to individual programming languages.

- **SC7**: ISO/IEC 15026-2, Software Assurance Case has entered Final Draft International Standard (FDIS) ballot; the final ISO/IEC ballot completed in December 2010.
  - Upon completion, it will be submitted for its final IEEE recirculation.
  - It is reasonable to anticipate publication of the standard, by both ISO/IEC and IEEE, in spring 2011.
ISO/IEC/IEEE 15026, System and Software Assurance

ISO/IEC 15288: Life cycle processes for systems
Common vocabulary, process architecture, and process description conventions

Other standards providing details of selected SW processes

ISO/IEC 12207: Life cycle processes for Software
ISO/IEC 15289: Document - ation
ISO/IEC 15288: Life cycle processes for systems

Interoperation

ISO/IEC 16326: Project Mgmt
ISO/IEC 15939: Measure - ment
ISO/IEC 16085: Risk Mgmt

Other standards providing details of selected system processes

ISO/IEC 15026: Additional practices for higher assurance systems

Other standards providing details of selected SW processes


"System and software assurance focuses on the management of risk and assurance of safety, security, and dependability within the context of system and software life cycle.

Terms of Reference changed: ISO/IEC JTC1/SC7 WG7, previously "System and Software Integrity" SC7 WG9"
ISO/IEC/IEEE 15026 Assurance Case

- Set of structured assurance claims, supported by evidence and reasoning (arguments), that demonstrates how assurance needs have been satisfied.
  - Shows compliance with assurance objectives
  - Provides an argument for the safety and security of the product or service.
  - Built, collected, and maintained throughout the life cycle
  - Derived from multiple sources

Sub-parts
- A high level summary
- Justification that product or service is acceptably safe, secure, or dependable
- Rationale for claiming a specified level of safety and security
- Conformance with relevant standards & regulatory requirements
- The configuration baseline
- Identified hazards and threats and residual risk of each hazard / threat
- Operational & support assumptions

Attributes
- Clear
- Consistent
- Complete
- Comprehensible
- Defensible
- Bounded
- Addresses all life cycle stages
Common Criteria v4 CCDB
• TOE to leverage CAPEC & CWE
• Also investigating how to leverage ISO/IEC 15026 NIAP Evaluation Scheme
• Above plus
• Also investigating how to leverage Security Content Automation Protocol (SCAP)
Need for Rating Schemes

► Rating of Suppliers providing software products and services
  ▪ Standards-based or model-based frameworks to support process improvement and enable benchmarking of organizational capabilities
  ▪ Credential programs for professionals involved in software lifecycle activities and decisions

► Rating of Software products:
  ▪ Supported by automation
  ▪ Standards-based
  ▪ Rules for aggregation and scaling
  ▪ Verifiable by independent third parties
  ▪ Labeling to support various needs (eg., security, dependability, etc)
  ▪ Meaningful and economical for consumers and suppliers

Collaborate with OWASP “Security Facts” labeling efforts
BSI https://buildsecurityin.us-cert.gov focuses on making Software Security a normal part of Software Engineering.

SwA Community Resources and Information Clearinghouse (CRIC) https://buildsecurityin.us-cert.gov/swa/ focuses on all contributing disciplines, practices and methodologies that advance risk mitigation efforts to enable greater resilience of software/cyber assets.

The SwA CRIC provides a primary resource for SwA Working Groups. Where applicable, SwA CRIC & BSI provide relevant links to each other.
Software Assurance (SwA) Pocket Guide Series

SwA in Acquisition & Outsourcing
• Software Assurance in Acquisition and Contract Language
• Software Supply Chain Risk Management and Due-Diligence

SwA in Development
• Integrating Security into the Software Development Life Cycle
• Key Practices for Mitigating the Most Egregious Exploitable Software Weaknesses
• Risk-based Software Security Testing
• Requirements and Analysis for Secure Software
• Architecture and Design Considerations for Secure Software
• Secure Coding and Software Construction
• Security Considerations for Technologies, Methodologies & Languages

SwA Life Cycle Support
• SwA in Education, Training and Certification
• Secure Software Distribution, Deployment, and Operations
• Code Transparency & Software Labels
• Assurance Case Management
• Secure Software Environment and Assurance EcoSystem

SwA Measurement and Information Needs
• Making Software Security Measurable
• Practical Measurement Framework for SwA and InfoSec
• SwA Business Case and Return on Investment

SwA Pocket Guides and SwA-related documents are collaboratively developed with peer review; they are subject to update and are freely available for download via the DHS Software Assurance Community Resources and Information Clearinghouse at https://buildsecurityin.us-cert.gov/swa (see SwA Resources)
Many SwA Resources Focus On Development
SOFTWARE ASSURANCE FORUM
BUILDING SECURITY IN

Process Improvement Lifecycle - A Process for Achieving Assurance

Understand Your Business Requirements for Assurance

Understand Assurance-Related Process Capability Expectations

Look to Standards for Assurance Process Detail

Build or Refine and Execute Your Assurance Processes

Measure Your Results

Mission/Business Process

Information System

Organization Support

Adapted from: Paul Croll, Computer Sciences Corporation, August 2007
The Assurance PRM Is A Holistic Framework

**Define Business Goals**

**Development Organization**
- DO 1 Establish the assurance resources to achieve key business objectives
- DO 2 Establish the environment to sustain the assurance program within the organization

**Acquisition and Supplier Management**
- AM 1 Select, manage, and use effective suppliers and third party applications based upon their assurance capabilities.

**Development Engineering**
- DE 1 Establish assurance requirements
- DE 2 Create IT solutions with integrated business objectives and assurance
- DE 3 Verify and Validate an implementation for assurance

**Development Project**
- DP 1 Identify and manage risks due to vulnerabilities throughout the product and system lifecycle
- DP 2 Establish and maintain assurance support from the project
- DP 3 Protect project and organizational assets

**Enterprise Assurance Support**
- ES 1 Establish and maintain organizational culture where assurance is an integral part of achieving the mission
- ES 2 Establish and maintain the ability to support continued delivery of assurance capabilities
- ES 3 Monitor and improve enterprise support to IT assets

**Enable Resilient Technology**
- Sustained environment to achieve business goals through technology

Created to facilitate Communication Across An Organization’s Multi-Disciplinary Stakeholders

Courtesy of Michele Moss, BAH, SwA Processes & Practices

https://buildsecurityin.us-cert.gov/swa/proself_assm.html
The DHS SwA Processes and Practices Working Group has synthesized the contributions of leading government and industry experts into a set of high-level goals and supporting practices (an evolution of the SwA community’s Assurance Process Reference Model).

The goals and practices are mapped to specific industry resources providing additional detail and real world implementation and supporting practices:

- Assurance Focus for CMMI
- Building Security In Maturity Model
- Open Software Assurance Maturity Model
- CERT® Resilience Management Model
- CMMI for Acquisition
- CMMI for Development
- CMMI for Services
- SwA Community’s Assurance Process Reference Model – Initial Mappings
- SwA Community’s Assurance Process Reference Model - Self Assessment
- SwA Community’s Assurance Process Reference Model – Mapping to Assurance Models

Other valuable resources that are in the process of being mapped include:

- NDIA System Assurance Guidebook
- Microsoft Security Development Lifecycle
- SAFECode
Process Reference Model for Assurance – Goals and Practices September 2010

In the following table, all references to “assurance” are intended to include system and software assurance, information assurance, and cyber security in support of the business/mission functions supported by systems and software.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Practice List</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DE 1 Establish assurance requirements</strong></td>
<td>Understand the operating environment and define the operating constraints for mission and information assurance within the environments of system development.</td>
</tr>
<tr>
<td></td>
<td>Develop customer mission and information assurance requirements</td>
</tr>
<tr>
<td></td>
<td>Define product and product component assurance requirements</td>
</tr>
<tr>
<td></td>
<td>Identify operational concepts and associated scenarios for intended and unintended use and associated assurance considerations</td>
</tr>
<tr>
<td></td>
<td>Identify appropriate controls for integrity and availability of the system to in support of organizational objectives</td>
</tr>
<tr>
<td></td>
<td>Analyze assurance requirements</td>
</tr>
<tr>
<td></td>
<td>Balance assurance needs against cost benefits</td>
</tr>
<tr>
<td></td>
<td>Obtain Agreement of risk for assurance level</td>
</tr>
</tbody>
</table>

https://buildsecurityin.us-cert.gov/swa/proself_assm.html
It can be used by acquirers, suppliers and integrators as a tool to discuss areas of strength and weakness.

- What assurance goals are being met?
- What practices are being implemented?
- Who are the suppliers and how are they managing risk?

<table>
<thead>
<tr>
<th>SwA Community Assurance Process Reference Model – Self Assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Goal</strong></td>
</tr>
<tr>
<td><strong>Development – Engineering</strong></td>
</tr>
<tr>
<td>DE 1 Establish assurance requirements</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

https://buildsecurityin.us-cert.gov/swa/proself_assm.html
It can be used as a navigation tool to guide SwA implementation efforts.

You have been asked to ensure that the OWASP Top Ten (an assurance coding Standard) are not in the Code.

You can look at the OSAMM for guidance on how to do it.

---

**SwA Community’s Assurance Process Reference Model - Initial Mappings**

In the following table, all references to “assurance” are intended to include system and software assurance, information assurance, and cybersecurity in support of the business/mission functions supported by systems and software.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Practice</th>
<th>AF CMMI</th>
<th>BSIMM</th>
<th>CMMI-ACQ</th>
<th>CMMI-DEV</th>
<th>CMMI-SVC</th>
<th>OSAMM</th>
<th>RMM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DE 2 Create IT solutions with integrated business objectives and assurance</strong></td>
<td><strong>Develop alternative solutions and selection criteria for mission and information assurance.</strong></td>
<td>AF TS SP 1.1.1</td>
<td>SF01.1</td>
<td>ATM SG2</td>
<td>TS SG1</td>
<td>SA1A</td>
<td>RTSE:SG 1 - SG2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SF01.2</td>
<td></td>
<td>AVAL SG2</td>
<td></td>
<td>SA1B</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Architect for mission and information assurance.</strong></td>
<td>AF TS SP 2.1.1</td>
<td>SF02.1</td>
<td>ATM SG2</td>
<td>TS SG2</td>
<td>SA2A</td>
<td>RTSE:SG 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SF02.3</td>
<td></td>
<td>AVAL SG2</td>
<td></td>
<td>SA2B</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Design for mission and information assurance.</strong></td>
<td>AF TS SP 2.1.2</td>
<td>SF02.1</td>
<td>ATM SG2</td>
<td>TS SG2</td>
<td>SA2A</td>
<td>RTSE:SG 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SF02.2</td>
<td></td>
<td>AVAL SG2</td>
<td></td>
<td>SA2B</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Identify deviations from mission and information assurance designs of the product components.</strong></td>
<td>AF TS SP 3.1.1</td>
<td>SF03.1</td>
<td>ATM SG2</td>
<td>TS SG3</td>
<td>SA1B</td>
<td>RTSE:SG 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SF03.2</td>
<td></td>
<td>AVAL SG2</td>
<td></td>
<td>SA1B</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Identify deviations from mission and information assurance coding standards. Implement appropriate mitigation to meet defined mission and information assurance objectives.**

**AF TS SP 3.1.2**

- **CR1.4**
  - AVER SG3
  - TS SG3

- **CR2.3**
  - **CR3.1**
  - **CR3.2**
  - **CR3.3**

Visit [https://buildsecurityin.us-cert.gov/swa/proself_assm.html](https://buildsecurityin.us-cert.gov/swa/proself_assm.html) for more information.
In the following table, all references to “assurance” are intended to include system and software assurance, and cyber security in support of the business/mission functions supported by systems and software.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Practice</th>
<th>CMMI-ACQ</th>
<th>CMMI-DEV</th>
<th>CMMI-SVC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DE 1 Establish assurance requirements</strong></td>
<td><strong>Development – Engineering</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Understand the operating environment and define the operating constraints for mission and information assurance within the environments of system development.</td>
<td>PP SG1</td>
<td>IPPD SG1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Develop customer mission and information assurance requirements</td>
<td>ARD SG1, SG3</td>
<td>RD SG1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>REQM SG1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Define product and product component assurance requirements</td>
<td>CM SG1</td>
<td>RD SG2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identify operational concepts and associated scenarios for intended and unintended use and associated assurance considerations</td>
<td>RSKM SG1 – SG2</td>
<td>RD SG3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identify appropriate controls for integrity and availability of the system to in support of organizational objectives</td>
<td>RSKM SG1</td>
<td>RSKM SG1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analyze assurance requirements</td>
<td>ARD SG3</td>
<td>RD SG3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Balance assurance needs against cost benefits</td>
<td>ARD SG3</td>
<td>RD SG3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Obtain Agreement of risk for assurance level</td>
<td>RSKM SG2</td>
<td>RSKM SG2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Efforts are underway to map to
- ISO/IEEE 15288
- ISO/IEEE 12207
<table>
<thead>
<tr>
<th>Assurance PRM</th>
<th>SAFEcode</th>
<th>MS SDL</th>
<th>Open SAMM</th>
<th>BSIMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Establish and maintain the strategic assurance training needs of the organization • Ensure resources have the training needed to do their job</td>
<td>1. Foundational (everyone) 2. Advanced (secure coding and testing practices) 3. Specialized (role-based)</td>
<td>1. Basic Concepts 2. Common Baseline 3. Custom Training</td>
<td>1. Technical Security Awareness training 2. Role specific guidance 3. Comprehensive security training and certifications</td>
<td>1. Create the software security satellite 2. Make customized, role-based training available on demand 3. Provide recognition for skills and career path progression</td>
</tr>
</tbody>
</table>

Source: SwA Benchmarking and Implementation, Moss, SSTC 2010
It can be used to begin the translation of SwA Activities across organizational leadership.
April 2009 SwA Report provides background, context and examples:

- Motivators
- Cost/Benefit Models Overview
- Measurement
- Risk
- Prioritization
- Process Improvement & Secure Software
- Globalization
- Organizational Development
- Case Studies and Examples
The Center for Internet Security

Practical Measurement Framework for Software Assurance and Information Security

October 2008

The CIS Security Metrics

February 9, 2009

Measuring Cyber Security and Information Assurance

Consensus Metric Definitions

© 2009 The Center for Internet Security
Software Assurance Ecosystem: The Formal Framework

The value of formalization extends beyond software systems to include related software system process, people and documentation.

**Process, People & Documentation**
- Evaluation Environment
  - Some point tools to assist evaluators but mainly manual work
  - Claims in Formal SBVR vocabulary
  - Evidence in Formal SBVR vocabulary
  - Large scope requires large effort

**Software System / Architecture Evaluation**
- Many integrated & highly automated tools to assist evaluators
- Claims and Evidence in Formal vocabulary
- Combination of tools and ISO/OMG standards
- Standardized SW System Representation In KDM
- Large scope capable (system of systems)
- Iterative extraction and analysis for rules

**Software System Artifacts**

**Process Docs & Artifacts**

**Requirements/Design Docs & Artifacts**

**Claims, Arguments and Evidence Repository**
- Formalized in SBVR vocabulary
- Automated verification of claims against evidence
- Highly automated and sophisticated risk assessments using transitive inter-evidence point relationships

**Hardware Environment**

**Reports Risk Analysis, etc)**

**Process, People, documentation Evidence**

**Formalized Specifications**

**Software system Technical Evidence**

**Executable Specifications**

**Protection Profiles**

**CWE**
2010 CWE/SANS Top 25 Most Dangerous Software Errors

Copyright © 2010
http://cwe.mitre.org/top25/

Document version: 1.06 (pdf)          Date: September 27, 2010

Project Coordinators:  
Bob Martin (MITRE)  
Mason Brown (SANS)  
Alan Paller (SANS)  
Dennis Kirby (SANS)

Document Editor:  
Steve Christey (MITRE)

Introduction

The 2010 CWE/SANS Top 25 Most Dangerous Software Errors is a list of the most widespread and critical programming errors that can lead to serious software vulnerabilities. They are often easy to find, and easy to exploit. They are dangerous because they will frequently allow attackers to completely take over the software, steal data, or prevent the software from working at all.

The Top 25 list is a tool for education and awareness to help programmers to prevent the kinds of vulnerabilities that plague the software industry, by identifying and avoiding all-too-common mistakes that occur before software is even shipped. Software customers can use the same list to help them to ask for more secure software. Researchers in software security can use the Top 25 to focus on a narrow but important subset of all known security weaknesses. Finally, software managers and CIOs can use the Top 25 list as a measuring stick of progress in their efforts to secure their software.

The list is the result of collaboration between the SANS Institute, MITRE, and many top software security experts in the US and Europe. It leverages experiences in the development of the SANS Top 20 attack vectors (http://www.sans.org/top20/) and MITRE's Common Weakness Enumeration (CWE) (http://cwe.mitre.org/). MITRE maintains the CWE web site, with the support of the US Department of Homeland Security's National Cyber Security Division, presenting detailed descriptions of the top 25 programming errors along with authoritative guidance for mitigating and avoiding them. The CWE site contains data on more than 800 programming errors, design errors, and architecture errors that can lead to exploitable...
Many DHS sponsored efforts are key to changing how software-based systems are developed, deployed and operated securely.
SCAP 1.1 uses the following specifications:

- Extensible Configuration Checklist Description Format (XCCDF) 1.1.4, a language for authoring security checklists/benchmarks and for reporting results of checklist evaluation [QUI08]
- Open Vulnerability and Assessment Language (OVAL) 5.6, a language for representing system configuration information, assessing machine state, and reporting assessment results
- Open Checklist Interactive Language (OCIL) 2.0, a language for representing security checks that requires human feedback
- Common Platform Enumeration (CPE) 2.2, a nomenclature and dictionary of hardware, operating systems, and applications [BUT09]
- Common Configuration Enumeration (CCE) 5, a nomenclature and dictionary of system configurations
- Common Vulnerabilities and Exposures (CVE), a nomenclature and dictionary for software flaws
- Common Vulnerability Scoring System (CVSS) 2.0, an open specification for quantifying the severity of software flaw vulnerabilities [MEL07].
Software Assurance Automation Protocol (SwAAP)

- For measuring & enumerating software weaknesses and the assurance cases.

Common Weakness Enumeration (CWE),
Common Attack Pattern Enumeration & Classification (CAPEC),
Malware Attribute Enumeration & Characterization (MAEC),
Common Weakness Scoring System (CWSS),
Software Assurance Findings Expression Schema (SAFES),
NIST SAMATE’s “Software Transparency Label”,
ISO/IEC 15026 “Assurance Case” (ISO 15026),
OMG Software Assurance Evidence Metamodel (OMG SAEM),
OMG Argumentation Metamodel (OMG ARG),
OMG Structured Metrics Metamodel (OMG SMM),
OMG Knowledge Discovery Metamodel (OMG KDM),
OMG Abstract Syntax Tree Metamodel (OMG ASTM)

• plus SCAP to capture “accredited” system CPEs and CCE settings?
• OVAL checks for capturing “finger print” of software applications to address supply-chain risk measurement?
Enterprise IT Asset Management

Asset Inventory
Configuration Guidance Analysis
Vulnerability Analysis
Threat Analysis
Intrusion Detection
Incident Management

Operational Enterprise Networks
Centralized Reporting

Change Management
Development & Sustainment
Security Management Processes

Assessment of System Development, Integration, & Sustainment Activities and Certification & Accreditation

CVE/CWE/CVSS/CCE/CCSS/OVAL/XCCDF/CPE/CAPEC/MAEC/SBVR/CWSS/CEE/ARF

ScAP

SwAAP

Development & Sustainment Security Management Processes

ERAP

Enterprise IT Change Management

ECAP

Enterprise IT Asset Management
SwA and Operational Resilience

Adapted from September 2010 SwA Forum, CERT RMM for Assurance, Lisa Young, SEI

Courtesy of Michele Moss
The Rugged Software MANIFESTO

Focus on Resilience and Survivability - If compromised, damage to the software will be minimized, and it will recover quickly to an acceptable level of operating capacity; it is ‘rugged’
I am rugged - and more importantly, my code is rugged.
I recognize that software has become a foundation of our modern world.
I recognize the awesome responsibility that comes with this foundational role.
I recognize that my code will be used in ways I cannot anticipate, in ways it was not designed, and for longer than it was ever intended.
I recognize that my code will be attacked by talented and persistent adversaries who threaten our physical, economic, and national security.
I recognize these things - and I choose to be rugged.
I am rugged because I refuse to be a source of vulnerability or weakness.
I am rugged because I assure my code will support its mission.
I am rugged because my code can face these challenges and persist in spite of them.
I am rugged, not because it is easy, but because it is necessary... and I am up for the challenge.
The Rugged Software Manifesto

I am rugged... and more importantly, my code is rugged.

I recognize that software has become a foundation of our modern world.

I recognize the awesome responsibility that comes with this foundational role.

I recognize that my code will be used in ways I cannot anticipate, in ways it was not designed, and for longer than it was ever intended.

I recognize that my code will be attacked by talented and persistent adversaries who threaten our physical, economic, and national security.

I recognize these things - and I choose to be rugged.

I am rugged because I refuse to be a source of vulnerability or weakness.

I am rugged because I assure my code will support its mission.

I am rugged because my code can face these challenges and persist in spite of them.

I am rugged, not because it is easy, but because it is necessary... and I am up for the challenge.
IT/Software Supply Chain Management is a National Security & Economic Issue

- Adversaries can gain “intimate access” to target systems, especially in a global supply chain that offers limited transparency

- Advances in science and technology will always outpace the ability of government and industry to react with new policies and standards
  - National security policies must conform with international laws and agreements while preserving a nation’s rights and freedoms, and protecting a nation’s self interests and economic goals
  - Forward-looking policies can adapt to the new world of global supply chains
  - International standards must mature to better address supply chain risk management, IT security, systems & software assurance
  - Assurance Rating Schemes for software products and organizations are needed

- IT/software suppliers and buyers can take more deliberate actions to security-enhance their processes and practices to mitigate risks
  - Government & Industry have significant leadership roles in solving this
  - Individuals can influence the way their organizations adopt security practices

Globalization will not be reversed; this is how we conduct business – To remain relevant, standards and capability benchmarking measures must address “assurance” mechanisms needed to manage IT/Software Supply Chain risks.
SOFTWARE ASSURANCE FORUM

“Building Security In”

https://buildsecurityin.us-cert.gov/swa
SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

Homeland Security

Commerce

National Defense

Next SwA Working Group Sessions 14-16 Dec 2010 at MITRE, McLean, VA