Software Assurance

Public/Private Collaboration Efforts
Mitigating Exploitable Software Risks throughout the Lifecycle

Sept 21, 2010

Joe Jarzombek, PMP, CSSLP
Director for Software Assurance
National Cyber Security Division
Office of the Assistant Secretary for Cybersecurity and Communications
Interdependencies Between Physical and Cyber Infrastructures -- Need for secure software applications
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 to facilitate the adoption of Malware Attribute Enumeration Classification (MAEC), Common Weakness Enumeration (CWE), and Common Attack Pattern Enumeration and Classification (CAPEC).
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 and the processes for acquiring and developing software represent a material weakness.
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)
IT/software security risk landscape is a convergence between “defense in depth” and “defense in breadth”

Enterprise Risk Management and Governance are security motivators

Acquisition could be considered the beginning of the lifecycle; not 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 and
-- Software Supply Chain Risk Management
“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
- Open Source
- US Dev. Center A
- Developed In-house
- 3rd Party Libraries
- US Dev. Center B
- Offshore
- Foreign Contractors
- Foreign Sub-Contractors

Procurement Process

- Purchased
- ISV (COTS)
- License 3rd Party Libraries
- Open Source
- Outsourcer Partners A
- Outsourcer Partners B
- 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
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.
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.
Need for Rating Schemes

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

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
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".
In Education and Training, Software Assurance could be addressed as:

• A “knowledge area” extension within each of the contributing disciplines;
• A stand-alone CBK drawing upon contributing disciplines;
• A set of functional roles, drawing upon a common body of knowledge; allowing more in-depth coverage dependent upon the specific roles.

Intent is to provide framework for curriculum development and evolution of contributing BOKs.

* See ‘Notes Page’ view for contributing BOK URLs and relevant links

The intent is not to create a new profession of Software Assurance; rather, to provide a common body of knowledge: (1) from which to provide input for developing curriculum in related fields of study and (2) for evolving the contributing disciplines to better address the needs of software security, safety, dependability, reliability and integrity.
Software Assurance Addresses Exploitable Software: Outcomes of non-secure practices and/or malicious intent

Exploitation potential of vulnerability is independent of “intent”

- Defects
- Intentional Vulnerabilities
- Unintentional Vulnerabilities
- Malware

*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
As part of the DHS risk mitigation effort, the SwA Program seeks to reduce software vulnerabilities, minimize exploitation, and address ways to improve the routine development of trustworthy software products and tools to analyze systems for hidden vulnerabilities.

The SwA framework encourages the production, evaluation and acquisition of better quality and more secure software; leverages resources to target the following four areas:

- **People** – education and training for developers and users
- **Processes** – sound practices, standards, and practical guidelines for the development of secure software
- **Technology** – diagnostic tools, cyber security R&D and measurement
- **Acquisition** – due-diligence questionnaires, contract templates and guidelines for acquisition management and outsourcing

* July 28, 2006 statement of George Foresman, DHS UnderSecretary for Preparedness, before the U.S. Senate Committee on Homeland Security and Governmental Affairs, Subcommittee on Federal Financial Management, Government Information, and International Security
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**

| SwA Common Body of Knowledge (CBK) & Glossary Organization of SwSys Security Principles/Guidelines SwA Developers' Guide on Security-Enhancing SDLC | SwA Metrics & Tool Evaluation (with NIST) |
| 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.*
DHS Software Assurance (SwA) Outreach

- Co-sponsor quarterly SwA WG sessions and semi-annual Software Assurance Forum for government, academia, and industry to facilitate ongoing public-private collaboration

- Co-sponsor SwA issues of CROSSTALK to “spread the word”
  - March 2007 issue on “Software Security”
  - May 2007 issue on “Software Acquisition”
  - Sep 2007 issue on “Service Oriented Architecture”
  - June 2008 issue on “Software Quality”
  - Sep 2008 issue on “Application Security”
  - Mar/Apr 2009 issue on “Reinforcing Good Practices”
  - Sep/Oct 2009 issue on “Resilient Software”
  - Mar/Apr 2010 issue on “Systems Assurance”
  - Sep/Oct 2010 issue on “Game Changing Tools & Practices”

- Provide outreach via DHS Speakers Bureau

- Collaborate with standards organizations, consortiums and professional societies in promoting SwA and participate in on-line communities, such as LinkedIn SwA mega-community

- Provide free SwA resources via “BuildSecurityIn” website to promote secure development methodologies (since Oct 05)

- Host Software Assurance Community Resources & Information Clearinghouse for SwA mega-community via https://buildsecurityin.us-cert.gov/SwA (since Dec 07)
Build Security In
Setting a higher standard for software assurance
Sponsored by DHS National Cyber Security Division

BSI https://buildsecurityin.us-cert.gov focuses on making Software Security a normal part of Software Engineering.

Software Assurance
Community Resources and Information Clearinghouse
Sponsored by DHS National Cyber Security Division

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) is the level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at any time during its life cycle, and that the software functions in the intended manner (from CNSI 4009 IA Glossary - see Wikipedia for definitions and descriptions).

As part of DHS risk mitigation efforts to enable greater resilience of cyber assets, the Software Assurance Program seeks to reduce software vulnerabilities, minimize exploitation, and address ways to routinely acquire, develop and deploy reliable and trustworthy software products with predictable execution, and to improve diagnostic capabilities to analyze systems for exploitable weaknesses.

The Software Assurance Forum and several working groups, composed of stakeholders in government, industry, and academia, are contributing to efforts focused on advancing software assurance objectives. Anyone can participate in these public/private collaboration activities. Information about upcoming Swa Forums and Working Group sessions is posted on the SwA Forums page as it becomes available.

Focused efforts for advancing software assurance are addressed in the working groups listed below. Click on any working group’s name to see Recent Releases and Updates, current activities, and other information for that working group.

- Workforce Education & Training
- Processes & Practices
- Technology, Tools & Product Evaluation
- Acquisition & Outsourcing
- Measurement
- Business Case
- Malware

WHY IS SOFTWARE ASSURANCE CRITICAL?

The nation’s critical infrastructure (energy, transportation, telecommunications, etc.), businesses, and services are extensively and increasingly controlled and enabled by software. Vulnerabilities in that software put those resources at risk. The risk is
Build Security In

Setting a higher standard for software assurance

Sponsored by DHS National Cyber Security Division

Search BSI:

What is Build Security In?

Build Security In is a collaborative effort that provides practices, tools, guidelines, rules, principles, and other resources that software developers, architects, and security practitioners can use to build security into software in every phase of its development.

Introduction to Software Security

Call for Authors and Reviewers

Submit an article for publication on BSI or volunteer to review new articles. See the Call for Authors and Reviewers for details.

Community Collaboration

To access other software assurance materials or to join the collaboration efforts of a related working group, visit the DHS Software Assurance Program's Community Resources and Information Clearinghouse.

Sponsor and Contributors

Build Security In is a Software Assurance strategic initiative of the National Cyber Security Division (NCSD) of the U.S. Department of Homeland Security. Peer-reviewed articles have been added.

Improve Security and Software Assurance: Tackle the CWE Top 25 – The Most Dangerous Programming Errors

The Top 25 CWEs represent the most significant exploitable software constructs that have made software so vulnerable. Addressing these will go a long way in securing software, both in development and in operation. Read more and see the list of Top 25 CWE Programming Errors on the Software Assurance Community Resources and Information Clearinghouse website.

Consistent with this list is the Top 10 for 2010 released by the Open Web Application Security Project (OWASP). OWASP's report captures the top ten risks associated with the use of web applications in an enterprise. Download the report, which contains examples and details that explain these risks to software developers, managers, and anyone interested in the future of web security, for free here.

What's New

Calls for papers have been posted for the Second IEEE International Conference on Information Privacy, Security, Risk, and Trust (PASSAT), the 1st International Workshop on Measurability of Security in Software Architectures (MeSSA), and the Software Assurance Minitrack of the 44th Hawaii International Conference on System Sciences (HICSS-44).

A new article, Improving Software Assurance, has been added.

A new article, Supply-Chain Risk Management: Incorporating Security, has been added.
Process Agnostic Lifecycle

Architecture & Design
- Architectural risk analysis
- Threat modeling
- Principles
- Guidelines
- Historical risks
- Modeling tools
- Resources

Code
- Code analysis
- Assembly, integration & evolution
- Coding practices
- Coding rules
- Code analysis
- Resources

Test
- Security testing
- White box testing
- Attack patterns
- Historical risks
- Resources

Requirements
- Requirements engineering
- Attack patterns
- Resources

System
- Penetration testing
- Incident management
- Deployment & operations
- Black box testing
- Resources

Touch Points & Artifacts

Fundamentals
- Risk management
- Project management
- Training & awareness
- Measurement
- SDLC process
- Business relevance
- Resources

Key
- Best (sound) practices
- Foundational knowledge
- Tools
- Resources
Software Security Engineering:
A Guide for Project Managers

Organized for Project Managers

- Derives material from DHS SwA “Build Security In” web site
  - https://buildsecurityin.us-cert.gov
- Provides a process focus for projects delivering software-intensive products and systems

Published in May 2008
July 2007 FREE publicly available resource provides a comprehensive look at efforts to improve the state of Software Security Assurance:

- describes the threats and common vulnerabilities to which software is subject;
- presents the many ways in which the S/W Security Assurance problem is being framed and understood across government, industry, and academia;
- describes numerous methodologies, best practices, technologies, and tools currently being used to specify, design, and implement software that will be less vulnerable to attack, and to verify its attack-resistance, attack-tolerance, and attack-resilience;
- offers a large number of available resources from which to learn more about principles and practices that constitute Software Security Assurance;
- provides observations about potentials for success, remaining shortcomings, and emerging trends across the S/W Security Assurance landscape.


The SOAR reflects output of efforts in the DoD-DHS Software Assurance Forum and Working Groups that provide collaborative venues for stakeholders to share and advance techniques and technologies relevant to software security.
• Describes how to integrate security principles and practices in software development life cycle
• Addresses security requirements, secure design principles, secure coding, risk-based software security testing, and secure sustainment
• Provides guidance for selecting secure development methodologies, practices, and technologies
  – Collaboratively developed/updated via SwA Forum working groups
  – Released Oct 2008 by DACS
  – Free, available for download via DACS & DHS SwA Community Resources & Information Clearinghouse

https://www.thedacs.com/techs/enhanced_life_cycles/
Common security-related elements of software development methodologies
- Security requirements help drive design, code handling, programming, and testing activities

Secure Programming practices:
- Minimize unsafe function use
- Use the latest compiler toolset
- Use static and dynamic analysis tools
- Use manual code review on high-risk code
- Validate input and output
- Use anti-cross site scripting libraries
- Use canonical data formats
- Avoid string concatenation for dynamic SQL
- Eliminate weak cryptography
- Use logging and tracing

Test to validate robustness and security
- Fuzz testing
- Penetration testing & third party assessment
- Automated test tools (in all development stages)

Code Integrity and Handling
- Least privilege access, Separation of duties,
- Persistent protection, Compliance management; Chain of custody & supply chain integrity.

Documentation (about software security posture & secure configurations)


Both collaboratively developed through the Software Assurance Working Group on Workforce Education and Training

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)
Organizations that provide security engineering & risk-based analysis throughout the lifecycle will have more resilient software products / systems.

“Build Security In” throughout the lifecycle

- **Plan**
  - Requirements and Use Cases
  - Architecture and Detailed Design

- **Design**
  - Security Design Reviews
  - Code and Testing

- **Build**
  - Application Security Testing

- **Deploy**
  - S/W Support Scanning & Remediation
  - Field Deployment and Feedback

Organizational Process Assets cover: governance, policies, standards, training, tailoring guidelines

- Leverage Software Assurance resources (freely available) to incorporate in training & awareness
- Modify SDLC to incorporate security processes and tools (should be done in phases by practitioners to determine best integration points)
- Avoid drastic changes to existing development environment and allow for time to change culture and processes
- Make the business case and balance the benefits
- Retain upper management sponsorship and commitment to producing secure software.

* Adopted in part from “Software Assurance: Mitigating Supply Chain Risks” (DHS NCSD SwA); “What to Test from a Security Perspective for the QA Professional” (Cigital) and “Neutralizing the Threat: A Case Study in Enterprise-wide Application Security Deployments” (Fortify Software & Accenture Security Technology Consulting)
Build Security In the SDLC

- Adding security practices throughout the SDLC establishes a software life cycle process that codifies both caution and intention.

- Key elements of a secure software life cycle process are:
  1. Security criteria in all software life cycle checkpoints (at entry & exit of a life cycle phase)
  2. Adherence to secure software principles and practices
  3. Adequate requirements, architecture, and design to address software security
  4. Secure coding practices with secure software integration/assembly practices
  5. Security testing practices that focus on verifying S/W dependability, trustworthiness, & resiliency
  6. Secure distribution and deployment practices and mechanisms
  7. Secure sustainment practices
  8. Supportive security tools (providing static & dynamic analysis) for developers and testers
  9. Secure software configuration management systems and processes
  10. Security risk analysis throughout the lifecycle

- Key people for producing secure software are:
  1. Security-knowledgeable software professionals
  2. Security-aware project management
  3. Upper management commitment to production of secure software

Adopted from Build Security In web site “Introduction to Software Security” which adapted or excerpted from Enhancing the Development Life Cycle to Produce Secure Software: A Reference Guidebook on Software Assurance [DHS/DACS 08].
“Software Assurance in Acquisition: Mitigating Risks to the Enterprise“
Version 1.0, Oct 2008, available for community use
Executive Summary

1. Introduction
   1.1 Background
   1.2 Purpose and Scope
   1.3 Audience—Acquisition Official Defined
   1.4 Document Structure
   1.5 Risk-Managed Software Acquisition Process

2. Planning Phase
   2.1 Needs Determination, Risk Categorization, & Solution Alternatives
   2.2 SwA Requirements
   2.3 Acquisition Plan and/or Acquisition Strategy
   2.4 Evaluation Plan and Criteria
   2.5 SwA Due Diligence Questionnaires

3. Contracting Phase
   3.1 Request for Proposals
      3.1.1 Work Statement
      3.1.2 Terms and Conditions
      3.1.3 Instructions to Suppliers
      3.1.4 Certifications
      3.1.5 Prequalification
   3.2 Proposal Evaluation
   3.3 Contract Negotiation
   3.4 Contract Award

4. Implementation and Acceptance Phase
   4.1 Contract Work Schedule
   4.2 Change Control
   4.3 Risk Management Plan
   4.4 Assurance Case Management
   4.5 Independent Software Testing
   4.6 Software Acceptance

5. Follow-on Phase
   5.1 Support and Maintenance
      5.1.1 Risk Management
      5.1.2 Assurance Case Management—Transition to Ops
      5.1.3 Other Change Management Considerations
   5.2 Disposal or Decommissioning

Appendix A/B—Acronyms/Glossary

Appendix C—An Imperative for SwA in Acquisition

Appendix D—Software Due Diligence Questionnaires
   Table D-1. COTS Proprietary Software Questionnaire
   Table D-2. COTS Open-Source Software Questionnaire
   Table D-3. Custom Software Questionnaire
   Table D-4. GOTS Software Questionnaire
   Table D-5. Software Services

Appendix E—Other Examples of Due Diligence Questionnaires

Appendix F—Sample Language for the RFP and/or Contract
   F.1 Security Controls and Standards
   F.2 Securely Configuring Commercial Software
   F.3 Acceptance Criteria
   F.4 Certifications
   F.5 Sample Instructions to Offerors Sections
   F.6 Sample Work Statement Sections
   F.7 Open Web Application Security Project
   F.8 Certification of Originality

Appendix H—References
<table>
<thead>
<tr>
<th>SwA Concern Categories</th>
<th>Risks</th>
<th>Purpose for Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software History and Licensing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Development Process Management</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software Security Training and Awareness</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Planning and Requirements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Architecture and Design</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software Development</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Built-in Software Defenses</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Component Assembly</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software Manufacture and Packaging</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Installation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assurance Claims and Evidence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software Change Management</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timeliness of Vulnerability Mitigation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Individual Malicious Behavior</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security “Track Record”</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Financial History and Status</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organizational History</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Foreign Interests and Influences</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service Confidentiality Policies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating Environment for Services</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security Services and Monitoring</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SwA Concern Categories</td>
<td>Risks</td>
<td>Purpose for Questions</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>Software History and Licensing</td>
<td>The software supplier’s development practice in using code of unknown origin may be unable to produce trustworthy software.</td>
<td>To address supply chain concerns and identify risks pertaining to history/pedigree of software during any and all phases of its life cycle that should have been considered by the supplier.</td>
</tr>
<tr>
<td>Development Process Management</td>
<td>If supplier project management does not perceive the value of SwA and enforce best practices, they will not be consistently implemented.</td>
<td>To determine whether project management enforces software assurance–related best practices.</td>
</tr>
<tr>
<td>Software Security Training and Awareness</td>
<td>Developers unaware of software assurance best practices are likely to implement software with security flaws (making it more susceptible to attack).</td>
<td>To determine whether training of developers in SwA best practices is a supplier policy and practice.</td>
</tr>
<tr>
<td>Planning and Requirements</td>
<td>If nonfunctional requirements (security, quality, safety) are not specified, developers will not implement them.</td>
<td>To determine whether the supplier’s requirements analysis process explicitly addresses SwA requirements.</td>
</tr>
<tr>
<td>Architecture and Design</td>
<td>The software may be designed without considering security or minimization of exploitable defects.</td>
<td>To determine how security is considered during the design phase.</td>
</tr>
<tr>
<td>Software Development</td>
<td>If developers lack qualified tools or if personnel are allowed to inappropriately access or change configuration items in the development environment, then delivered software might have unspecified features. The supplier might lack sufficient process capability to deliver secure products, systems or services.</td>
<td>To ascertain that the supplier has and enforces policies and SwA practices in the development of software that use secure software development environments to minimize risk exposures.</td>
</tr>
<tr>
<td>Built-in Software Defenses</td>
<td>The software may lack preventive measures to help it resist attack effectively and proactively.</td>
<td>To ensure that capabilities are designed to minimize the exposure of the software’s vulnerabilities to external threats and to keep the software in a secure state regardless of the input and parameters it receives from its users or environment.</td>
</tr>
<tr>
<td>SwA Concern Categories</td>
<td>Risks</td>
<td>Purpose for Questions</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Component Assembly</td>
<td>Insufficient analysis of software components used to assemble larger software packages may introduce vulnerabilities to the overall package.</td>
<td>To ensure that the software components are thoroughly vetted for their security properties, secure behaviors, and known types of weaknesses that can lead to exploitable vulnerabilities.</td>
</tr>
<tr>
<td>Testing</td>
<td>Software released with insufficient testing may contain an unacceptable number of exploitable defects.</td>
<td>To determine whether the appropriate set of analyses, reviews, and tests are performed on the software throughout the life cycle which evaluate security criteria.</td>
</tr>
<tr>
<td>Software Manufacture and Packaging</td>
<td>Vulnerabilities or malicious code could be introduced in the manufacturing or packaging process.</td>
<td>To determine how the software goes through the manufacturing process, how it is packaged, and how it remains secure.</td>
</tr>
<tr>
<td>Installation</td>
<td>The software may not install as advertised and the acquirer may not get the software to function as expected.</td>
<td>To ensure the supplier provides an acceptable level of support during the installation process.</td>
</tr>
<tr>
<td>Assurance Claims and Evidence</td>
<td>Supplier assurance claims (with supporting evidence) may be non-existent or insufficiently verified.</td>
<td>To determine how suppliers communicate their claims of assurance; ascertain what the claims have been measured against, and identify at what levels they will be verified.</td>
</tr>
<tr>
<td>Support</td>
<td>Supplier ceases to supply patches and new releases prior to the acquirer ending use of software. Vulnerabilities may go unmitigated.</td>
<td>To ensure understanding of supplier policy for security fixes and when products are no longer supported.</td>
</tr>
<tr>
<td>Software Change Management</td>
<td>Weak change control procedures can corrupt software and introduce new security vulnerabilities.</td>
<td>To determine whether software changes are adequately assessed and verified by supplier management.</td>
</tr>
<tr>
<td>Timeliness of Vulnerability Mitigation</td>
<td>Sometimes it can be extremely difficult to make a software supplier take notice and repair software to mitigate reported vulnerabilities.</td>
<td>To ensure security defects and configuration errors are fixed properly and in a timely fashion.</td>
</tr>
</tbody>
</table>
### Table 1 – SwA Concern Categories

<table>
<thead>
<tr>
<th>SwA Concern Categories</th>
<th>Risks</th>
<th>Purpose for Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individual Malicious Behavior</td>
<td>A developer purposely inserts malicious code, and supplier lacks procedures to mitigate risks from insider threats within the supply chain.</td>
<td>To determine whether the supplier has and enforces policies to minimize individual malicious behavior.</td>
</tr>
<tr>
<td>Security ‘Track Record’</td>
<td>A software supplier that is unresponsive to known software vulnerabilities may not mitigate/patch vulnerabilities in a timely manner.</td>
<td>To establish insight into whether the supplier places a high priority on security issues and will be responsive to vulnerabilities they will need to mitigate.</td>
</tr>
<tr>
<td>Financial History and Status</td>
<td>A software supplier that goes out of business will be unable to provide support or mitigate product defects and vulnerabilities.</td>
<td>To identify documented financial conditions or actions of the supplier that may impact its viability and stability, such as mergers, sell-offs, lawsuits, and financial losses.</td>
</tr>
<tr>
<td>Organizational History</td>
<td>There may be conflicting circumstances or competing interests within the organization that may lead to increased risk in the software development.</td>
<td>To understand the supplier’s organizational background, roles, and relationships that might have an impact on supporting the software.</td>
</tr>
<tr>
<td>Foreign Interests and Influences</td>
<td>There may be controlling foreign interests (among organization officers or from countries) with malicious intent to the users’ country or organization planning to use the software.</td>
<td>To help identify supplier companies that may have individuals with competing interests or malicious intent to a domestic buyer/user.</td>
</tr>
<tr>
<td>Service Confidentiality Policies</td>
<td>Without policies to enforce client data confidentiality/privacy, acquirer’s data could be at risk without service supplier liability.</td>
<td>To determine the service provider’s confidentiality and privacy policies and ensure their enforcement.</td>
</tr>
<tr>
<td>Operating Environment for Services</td>
<td>Operating environment for the services may not be hardened or otherwise secure.</td>
<td>To understand the controls the supplier has established to operate the software securely.</td>
</tr>
<tr>
<td>Security Services and Monitoring</td>
<td>Insufficient security monitoring may allow attacks to impact services.</td>
<td>To ensure software and its operating environment are regularly reviewed for adherence to SwA requirements through periodic testing and evaluation.</td>
</tr>
<tr>
<td>No</td>
<td>Question</td>
<td>COTS Proprietary</td>
</tr>
<tr>
<td>----</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>1</td>
<td>Can the pedigree of the software be established? Briefly explain what is known of the people and processes that created the software.</td>
<td>✓</td>
</tr>
<tr>
<td>2</td>
<td>Explain the change management procedure that identifies the type and extent of changes conducted on the software throughout its life cycle.</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>What type of license(s) are available for the open source software? Is it compatible with other software components in use? Is indemnification provided, and will the supplier indemnify the purchasing organization from any issues in the license agreement? Explain.</td>
<td>✓</td>
</tr>
<tr>
<td>4</td>
<td>Is there a clear chain of licensing from original author to latest modifier? Describe the chain of licensing.</td>
<td>✓</td>
</tr>
<tr>
<td>5</td>
<td>What assurances are provided that the licensed software does not infringe upon any copyright or patent? Explain.</td>
<td>✓</td>
</tr>
<tr>
<td>6</td>
<td>Does the company have corporate policies and management controls in place to ensure that only corporate-approved (licensed and vetted) software components are used during the development process? Explain.</td>
<td>✓</td>
</tr>
<tr>
<td>7</td>
<td>Are licensed software components still valid for the intended use?</td>
<td>✓</td>
</tr>
<tr>
<td>8</td>
<td>Is the software in question original source or a modified version?</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Has the software been reviewed to confirm that it does not infringe upon any copyright or patent?</td>
<td>✓</td>
</tr>
<tr>
<td>10</td>
<td>How long has the software source been available? Is there an active user community providing peer review and actively evolving the software?</td>
<td>✓</td>
</tr>
<tr>
<td>No.</td>
<td>Question</td>
<td>COTS Proprietary</td>
</tr>
<tr>
<td>-----</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>11</td>
<td>Does the license/contract restrict the licensee from discovering flaws or disclosing details about software defects or weaknesses with others (e.g., is there a “gag rule” or limits on sharing information about discovered flaws)?</td>
<td>✓</td>
</tr>
<tr>
<td>12</td>
<td>Does the license/contract restrict communications or limit the licensee in any potential communication with third-party advisors about provisions for support (e.g., is there a “gag rule” or limits placed on the licensee that affect ability to discuss contractual terms or breaches) regarding the licensed or contracted product or service?</td>
<td>✓</td>
</tr>
<tr>
<td>13</td>
<td>Does software have a positive reputation? Does software have a positive reputation relative to security? Are there reviews that recommend it?</td>
<td>✓</td>
</tr>
<tr>
<td>14</td>
<td>Is the level of security where the software was developed the same as where the software will operate?</td>
<td></td>
</tr>
</tbody>
</table>

**Development Process Management**

<table>
<thead>
<tr>
<th>No.</th>
<th>Question</th>
<th>COTS Proprietary</th>
<th>COTS Open-Source</th>
<th>GOTS</th>
<th>Custom</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>What are the processes (e.g., ISO 9000, CMMI, etc.), methods, tools (e.g., IDEs, compilers), techniques, etc. used to produce and transform the software (brief summary response)?</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>What security measurement practices and data does the company use to assist product planning?</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Is software assurance considered in all phases of development? Explain.</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>How is software risk managed? Are anticipated threats identified, assessed, and prioritized?</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Table 1 – SwA Concern Categories -- (with interests relevant to security and privacy)**

<table>
<thead>
<tr>
<th>SwA Concern Categories</th>
<th>Risks</th>
<th>Purpose for Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Service Confidentiality Policies</strong></td>
<td>Without policies to enforce client data confidentiality/privacy, acquirer’s data could be at risk without service supplier liability.</td>
<td>To determine the service provider’s confidentiality and privacy policies and ensure their enforcement.</td>
</tr>
</tbody>
</table>

**Table 3 - Questions for Hosted Applications**

<table>
<thead>
<tr>
<th>No.</th>
<th>Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>What are the customer confidentiality policies? How are they enforced?</td>
</tr>
<tr>
<td>2</td>
<td>What are the customer privacy policies? How are they enforced?</td>
</tr>
<tr>
<td>3</td>
<td>What are the policies and procedures used to protect sensitive information from unauthorized access? How are the policies enforced?</td>
</tr>
<tr>
<td>4</td>
<td>What are the set of controls to ensure separation of data and security information between different customers that are physically located in the same data center? On the same host server?</td>
</tr>
<tr>
<td>5</td>
<td>Who configures and deploys the servers? Are the configuration procedures available for review, including documentation for all registry settings?</td>
</tr>
<tr>
<td>7</td>
<td>What are the data backup policies and procedures? How frequently are the backup procedures verified?</td>
</tr>
<tr>
<td>11</td>
<td>What are the agents or scripts executing on servers of hosted applications? Are there procedures for reviewing the security of these scripts or agents?</td>
</tr>
<tr>
<td>12</td>
<td>What are the procedures and policies used to approve, grant, monitor and revoke access to the servers? Are audit logs maintained?</td>
</tr>
<tr>
<td>13</td>
<td>What are the procedures and policies for handling and destroying sensitive data on electronic and printed media?</td>
</tr>
<tr>
<td>15</td>
<td>What are the procedures used to approve, grant, monitor, and revoke file permissions for production data and executable code?</td>
</tr>
</tbody>
</table>
We are engaged with many parts of the Community for Software Assurance-related standardization.
ISO/IEC/IEEE 15026, System and Software Assurance

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
Many suppliers use maturity models to guide process improvement & assess capabilities; yet many models do not explicitly address safety and security.

Project leadership and team members need to know where and how to contribute.

Focus Topic: Assurance for Capability Maturity Model Integration (CMMI)® defines the Assurance Thread for Implementation and Improvement of Assurance Practices


Other Assurance Maturity Models have been released in 2009:
The Building Security In Maturity Model (BSIMM) helps organizations plan software security initiatives http://www.bsi-mm.com/
The Software Assurance Maturity Model (SAMM) which is an open framework to help organizations formulate and implement a strategy for software security that is tailored to specific risks facing the organization http://www.opensamm.org/
Project leadership and team members need to know where and how to contribute:

- Assurance PRM defines the goals and practices needed to achieve SwA
- Assurance for CMMI ® defines the Assurance Thread for Implementation and Improvement of Assurance Practices that are assumed when using the CMMI-DEV

Understanding gaps helps suppliers and acquirers prioritize organizational efforts and funding to implement improvement actions.

https://buildsecurityin.us-cert.gov/swa/procrsrc.html
• Capture and discuss community of practices software assurance issues
• Share best practices
• Provide community input to and comments on:
  – DHS and DoD Guidebooks relating to Software Assurance
  – National and International Software Assurance Standards
  – DHS and DoD Policy Guidance on System and Software Assurance
Processes & Practices Expected Outcomes

- In support of acquisition, management, and engineering and practices for software and systems assurance:
  - Community consensus standards for addressing assurance concerns throughout the system and software life cycles
  - Process benchmarking tools for assessing organizational capability with respect to assurance
  - Practice guidebooks providing compendiums of best practices and lessons learned
  - Community input to acquisition policy and guidance
Understand Assurance - Related Process Capability Expectations

Look to Standards for Assurance Process Detail

Build or Refine and Execute Your Assurance Processes

Mission/Business Process
Understand Your Business Requirements for Assurance

Measure Your Results

Information System

Organization Support

Adapted from: Paul Croll, Computer Sciences Corporation, August 2007
• Analyzed freely available models to determine how various models address similar goals and practices
• Identified the intersections of the common practices amongst the models regardless of the intended audience and levels of granularity
• Intended to support “Getting Started” by increasing awareness of improving software assurance by:
  – Learning how multiple models address similar assurance goals
  – Selecting practices from these models
• Provides a means for selecting models and practices that are best suited for the individual needs of various organizations
## Common SwA References Recommendations for Training

<table>
<thead>
<tr>
<th>Assurance PRM</th>
<th>SAFEcode</th>
<th>MS SDL</th>
<th>Open SAMM</th>
<th>BSIMM</th>
</tr>
</thead>
</table>
| 1. Foundational (everyone)  
2. Advanced (secure coding and testing practices)  
3. Specialized (role-based) | 1. Basic Concepts  
2. Common Baseline  
3. Custom Training | 1. Technical Security Awareness training  
2. Role specific guidance  
3. Comprehensive security training and certifications | 1. Create the software security satellite  
2. Make customized, role-based training available on demand  
3. Provide recognition for skills and career path progression |

Source: SwA Benchmarking and Implementation, Moss, SSTC 2010
Objectives for Creating A (Self) Assessment Tool

- Organizations must be able to understand and become aware of risk throughout the supply chain.
  - What assurance goals are being met?
  - What practices are being implemented?
  - Who are the suppliers and how are they managing risk?
- Organizations need to be able to quantify and baseline assurance and risk management activities to ensure rugged software and software services are being developed and acquired.
- Supply chain partners must achieve increased awareness and communication to effectively understand risk throughout the software supply chain.
## SwA Self-Assessment (High Level)

<table>
<thead>
<tr>
<th>Rule</th>
<th>Goal</th>
<th>Expected Practice</th>
<th>Activities</th>
<th>Source</th>
<th>CMMI-ACQ</th>
<th>OSEM</th>
<th>RMM</th>
<th>NSDOL</th>
<th>Developer Considerations</th>
<th>Acquirer Considerations</th>
<th>Practice Implementation Level</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRV</td>
<td></td>
<td></td>
<td>Planning the system assurance context.</td>
<td>AF SP 1.1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Planning the system vulnerabilities with each operating environment defined for the system.</td>
<td>AF SP 1.2</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Planning applicable assurance laws, policies, and constraints.</td>
<td>AF SP 2.1</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related threats and vulnerabilities.</td>
<td>AF SP 2.2</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related control objectives.</td>
<td>AF SP 2.3</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related risk assessments.</td>
<td>AF SP 2.4</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related testing procedures.</td>
<td>AF SP 2.5</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related training procedures.</td>
<td>AF SP 2.6</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related audit procedures.</td>
<td>AF SP 2.7</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident response procedures.</td>
<td>AF SP 2.8</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related contingency planning procedures.</td>
<td>AF SP 2.9</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related communication procedures.</td>
<td>AF SP 3.1</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related reporting procedures.</td>
<td>AF SP 3.2</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related review procedures.</td>
<td>AF SP 3.3</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related enforcement procedures.</td>
<td>AF SP 3.4</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related accountability procedures.</td>
<td>AF SP 3.5</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related governance procedures.</td>
<td>AF SP 3.6</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related compliance procedures.</td>
<td>AF SP 3.7</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related monitoring procedures.</td>
<td>AF SP 3.8</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident management procedures.</td>
<td>AF SP 3.9</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident response procedures.</td>
<td>AF SP 4.1</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident reporting procedures.</td>
<td>AF SP 4.2</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident review procedures.</td>
<td>AF SP 4.3</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident enforcement procedures.</td>
<td>AF SP 4.4</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident accountability procedures.</td>
<td>AF SP 4.5</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident governance procedures.</td>
<td>AF SP 4.6</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident monitoring procedures.</td>
<td>AF SP 4.7</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident management procedures.</td>
<td>AF SP 4.8</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident response procedures.</td>
<td>AF SP 4.9</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident reporting procedures.</td>
<td>AF SP 4.10</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident review procedures.</td>
<td>AF SP 4.11</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident enforcement procedures.</td>
<td>AF SP 4.12</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident accountability procedures.</td>
<td>AF SP 4.13</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident governance procedures.</td>
<td>AF SP 4.14</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident monitoring procedures.</td>
<td>AF SP 4.15</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident management procedures.</td>
<td>AF SP 4.16</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident response procedures.</td>
<td>AF SP 4.17</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident reporting procedures.</td>
<td>AF SP 4.18</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident review procedures.</td>
<td>AF SP 4.19</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident enforcement procedures.</td>
<td>AF SP 4.20</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident accountability procedures.</td>
<td>AF SP 4.21</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident governance procedures.</td>
<td>AF SP 4.22</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident monitoring procedures.</td>
<td>AF SP 4.23</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Identify and document all security related incident management procedures.</td>
<td>AF SP 4.24</td>
<td>ORG</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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
Practical Measurement Framework for Software Assurance and Information Security

Oct 2008
Measurement Guidance: Purpose

- To provide a practical framework for measuring software assurance achievement of SwA goals and objectives within the context of individual projects, programs, or enterprises.
  - Making informed decisions in the software development lifecycle related to information security compliance, performance, and functional requirements/controls
  - Facilitate adoption of secure software design practices
  - Mitigate risks throughout the System Development Lifecycle (SDLC) and ultimately reduce the numbers of vulnerabilities introduced into software code during development
  - Determining if security/performance/trade-offs have been defined and accepted
  - Assessing the trustworthiness of a system.
- Can be applied beyond SwA to a variety of security-related measurement efforts to help facilitate risk-based decision making through providing quantitative information on a variety of aspects of organization’s security related performance.
Measurement Guidance: Scope & Resources

- Common measurement framework and measurement process leverage established measurement methodologies or emerging measurement methodologies that enjoy broad industry support:
  - Capability Maturity Model Integration (CMMI) Measurement & Analysis
  - CMMI Goal Question Indicator Measure (GQ(I)M)

- A listing of resources has been published on the SwA web site targeting primary stakeholder groups: Executive, Developer/Vendor/Supplier, Buyer/Acquirer
  - Sample SwA goals and questions lists to be used to define measures
  - Sources of measurable requirements, such as NIST documents
  - Articles on related subjects, including SwA measurement, security measurement, and software security measurement
  - Useful links
  - Measures library
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

**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

**Reports**
- Risk Analysis, etc.

**IA Controls**
- CWE

**Hardware Environment**

**Software System Artifacts**
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 software configurations
- Common Vulnerabilities and Exposures (CVE), a nomenclature and dictionary of software flaws
- Common Vulnerability Scoring System (CVSS) 2.0, an open specification for scoring the severity of software flaw vulnerabilities [MEL07]
5. SCAP Use Case Requirements

5.1 SCAP Data Streams

5.2 SCAP Configuration Verification

5.3 SCAP Vulnerability Assessment

5.3.1 SCAP Vulnerability Assessment Using XCCDF and OVAL

5.3.2 SCAP Vulnerability Assessment Using Standalone OVAL

5.3.3 OVAL Definitions and Vulnerability Assessment

5.4 Patch Validation

5.4.1 Using OVAL Definitions for Patch Validation

5.4.2 Referencing an OVAL Patch Data Stream

5.5 SCAP Inventory Collection
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 Metammodel (OMG SAEM),
OMG Argumentation Metammodel (OMG ARG),
OMG Structured Metrics Metammodel (OMG SMM),
OMG Knowledge Discovery Metammodel (OMG KDM),
OMG Abstract Syntax Tree Metammodel (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?
“Other” Automation Protocols (“O”AP)

- **Event Management Automation Protocol (EMAP)**
  - For reporting of security events.
  - Uses Common Event Expression (CEE), Malware Attribute Enumeration & Characterization (MAEC), CAPEC, etc.

- **Enterprise Remediation Automation Protocol (ERAP)**
  - For automated remediation of mis-configuration & missing patches.
  - Uses Common Remediation Enumeration (CRE) and Extended Remediation Information (ERI).

- **Enterprise Compliance Automation Protocol (ECAP)**
  - For reporting configuration compliance.
  - Uses Asset Reporting Format (ARF), Open Checklist Reporting Language (OCRL), etc.

- **Enterprise System Information Protocol (ESIP)**
  - For reporting of asset inventory information.
  - Uses ..... 

- **Threat Analysis Automation Protocol (TAAP)**
  - For analyzing threats and security risks.
  - Uses ..... 

- **Incident Management Automation Protocol (IMAP)**
  - For supporting incident management and response.
  - Uses IODEF, etc
Enterprise IT Asset Management

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

Operational Enterprise Networks
Centralized Reporting

Change Management
Security Management Processes
Assessment of System Development, Integration, & Sustainment Activities and Certification & Accreditation

SwAAP
Development & Sustainment Security Management Processes

SCAP
Operations Security Management Processes
CVE/CWE/CVSS/CCE/CCSS/OVAL/XCCDF/CPE/CAPEC/MAEC

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

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

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

ESIP
SwAAP
ERAP
ECAP

Centralized Reporting
Enterprise IT Asset Management

Operations Security Management Processes
CVE/CWE/CVSS/CCE/CCSS/OVAL/XCCDF/CPE/CAPEC/MAEC/CEE

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

CWE/CAPEC/SBVR/CWSS/MAEC/OVAL/XCCDF/CPE/ARF

SCAP
Operations Security Management Processes
CVE/CWE/CVSS/CCE/CCSS/OVAL/XCCDF/CPE/CAPEC/MAEC

ESIP
SwAAP
ERAP
ECAP

Centralized Reporting
Enterprise IT Asset Management

Operations Security Management Processes
CVE/CWE/CVSS/CCE/CCSS/OVAL/XCCDF/CPE/CAPEC/MAEC/CEE

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

CWE/CAPEC/SBVR/CWSS/MAEC/OVAL/XCCDF/CPE/ARF
“A fortress mentality will not work in cyber. We cannot retreat behind a Maginot Line of firewalls...If we stand still for a minute, our adversaries will overtake us.”

-William Lynn, U.S. Deputy Secretary of Defense
January 2010
**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.

---

**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.

Twitter: @RuggedSoftware

http://ruggedsoftware.org
The Journey

Aware  Informed  Selective  Mature
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 Forum 27 Sep – 1 Oct 2010 at NIST, Gaithersburg, MD