Software Assurance: Enabling Security and Resilience throughout the Software Lifecycle

Joe Jarzombek, PMP, CSSLP
Director for Software Assurance
Cyber Security & Communications
US Department of Homeland Security
Public/Private Collaboration Efforts for Security Automation and Software Supply Chain Risk Management

Next SwA Working Groups sessions: 27-29 Nov 2012 at MITRE, McLean, VA
Software Assurance Addresses Exploitable Software:
Outcomes of non-secure practices and/or malicious intent

Exploitation potential of vulnerability is independent of “intent”

Software Assurance (SwA) is the level of confidence that software functions as intended and is free of vulnerabilities, either intentionally or unintentionally designed or inserted as part of the software throughout the life cycle.*

*From CNSS Instruction 4009 “National Information Assurance Glossary” (26APR2010)
Challenges in Mitigating Risks Attributable to Exploitable Software and Supply Chains

• Complexity hampers 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:
  – discern product assurance;
  – benchmark process capabilities, and
  – assert claims about the assurance of products, systems and services,

• “provenance and pedigree of supply chain actors” become a more dominant consideration for security/safety-critical applications:
  – Enterprises and Users 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.
Challenges in Mitigating Risks Attributable to Exploitable Software and Supply Chains

• Several needs arise:
  – Need internationally recognized standards to support security automation and processes to 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.
Enterprises seek comprehensive capabilities to:

- Avoid installing software with **MALWARE** pre-installed.
- Determine that no publicly reported **VULNERABILITIES** remain in code prior to operational acceptance, and that future discoveries of common vulnerabilities and exposures can be quickly patched.
- Determine that exploitable software **WEAKNESSES** that put the users most at risk are mitigated prior to operational acceptance.
Challenges in Preventing and Responding to Cyber Incidents

• “Silos” of operation
• Proprietary reporting formats

• Needs arise:
  – Need standards to support security automation and processes to support exchange of information and cyber indicators relative to incident management and response.

Software Assurance Forum & Working Groups*
Exploitable Software Weaknesses (CWEs) are sources for future Zero-Day Attacks

- Cross-site Scripting (XSS) Attack (CAPEC-86)
- Improper Neutralization of Input During Web Page Generation (CWE-79)
- SQL Injection Attack (CAPEC-66)
- Improper Neutralization of Special Elements used in an SQL Command (CWE-89)
Software Security Assurance: Not just a good idea

- Many people responsible for protecting most critical infrastructure facilities have felt comfortable about security of their systems.
  - Facilities rely on industrial control systems (ICS) -- custom-built suites of systems that control essential mechanical functions of power grids, processing plants, etc -- usually not connected to the Internet, also known as "air-gapped."
  - Many industry owners, operators and regulators believed that this security model provided an infallible, invulnerable barrier to malicious cyber attacks from criminals and advanced persistent threat (APT) adversaries.
- National Defense Authorization Act (NDAA) -- which included a focus on software security (in Section 932, Strategy on Computer Software Assurance) -- serves as first cybersecurity law of 2011 and requires the U.S. Dept of Defense to develop a strategy for ensuring the security of software applications.
- Software Security Assurance, a set of practices for ensuring proactive application security, is key to making applications compliant with this new law.

“How Stuxnet Demonstrates That Software Assurance Equals Mission Assurance:
The rules of the game have changed,” by Rob Roy, Federal CTO of Fortify, an HP Company
Software Security Assurance: Not just a good idea

Steps organizations can take now to support software security assurance.

Tips from white paper on “7 Practical Steps to Delivering More Secure Software”:
1. Quickly evaluate current state of software security and create a plan for dealing with it throughout the life cycle.
2. Specify the risks and threats to the software so they can be eliminated before they are deployed.
3. Review the code for security vulnerabilities introduced during development.
4. Test and verify the code for vulnerabilities.
5. Build a gate to prevent applications with vulnerabilities from going into production.
6. Measure the success of the security plan so that the process can be continually improved.
7. Educate stakeholders about security so they can implement the security plan.

Any development organization can implement this security plan immediately and begin to receive a return on their efforts within a minimal period of time. The key is to start now.

To complement the software strategy, there are several other areas of good security practices to observe and implement if they are not already part of the organizational security approach:
1. Implement software configurations such as the U.S. Government Configuration Baseline (formerly the Federal Desktop Core Configuration), strong authentication, and strict, documented internal policies and procedures.
2. Ask vendors to provide guarantees of software security as required by HR 6523.
3. Insert and enforce software assurance requirements in contracts.
4. Review IT security policies to ensure that all users of organizational networks and data comply with the strictest security policies possible with respect to the mission.
5. Determine how much risk the organization can afford and who is accountable for that risk. Constructing a new building in parts of California without accounting for earthquakes is unacceptable.

“How Stuxnet Demonstrates That Software Assurance Equals Mission Assurance:
The rules of the game have changed,” by Rob Roy, Federal CTO of Fortify, an HP Company
http://email.tailorednews.com/r/jm892fwx7ega4ZTy4QI.htm
What’s in the DoD Policy Memo?

- “Every acquisition program shall submit a PPP for Milestone Decision Authority review and approval at Milestone A and shall update the PPP at each subsequent milestone and the Full-Rate Production decision.”

- Expected business practice, effective immediately, and reflected in upcoming DoDI 5000.02 and DAG updates

The PPP is the Single Focal Point for All Security Activities on the Program

http://www.acq.osd.mil/se/pg/index.html#PPP
Software Assurance Methods

**Development Process**
Apply assurance activities to the procedures and structure imposed on software development

**Operational System**
Implement countermeasures to the design and acquisition of end-item software products and their interfaces

**Development Environment**
Apply assurance activities to the environment and tools for developing, testing, and integrating software code and interfaces

---

**Additional Guidance in PPP Outline and Guidance**

---

**Table 5.3-5-5: Application of Software Assurance Countermeasures (sample)**

<table>
<thead>
<tr>
<th>Software (CPI, critical function components, other software)</th>
<th>Development Process</th>
<th>Code Inspect p/a</th>
<th>CVE p/a</th>
<th>CAPEC p/a</th>
<th>CWE p/a</th>
<th>Pen Test</th>
<th>Test Coverage p/a</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developmental CPI SW</td>
<td>100/80%</td>
<td>Two Levels</td>
<td>100/80</td>
<td>100/60</td>
<td>100/60</td>
<td>Yes</td>
<td>75/50%</td>
</tr>
<tr>
<td>Developmental Critical Function SW</td>
<td>100/80%</td>
<td>Two Levels</td>
<td>100/80</td>
<td>100/70</td>
<td>100/70</td>
<td>Yes</td>
<td>75/50%</td>
</tr>
<tr>
<td>Other Developmental SW</td>
<td>none</td>
<td>One level</td>
<td>100/65</td>
<td>10/0</td>
<td>10/0</td>
<td>No</td>
<td>50/25%</td>
</tr>
<tr>
<td>COTS CPI and Critical Function SW</td>
<td>Vendor SwA</td>
<td>Vendor SwA</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Yes</td>
<td>UNK</td>
</tr>
<tr>
<td>COTS (other than CPI and Critical Function) and NDI SW</td>
<td>No</td>
<td>No</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>No</td>
<td>UNK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operational System</th>
<th>Failover</th>
<th>Multiple Supplier Redundancy</th>
<th>Fault Isolation</th>
<th>Least Privilege</th>
<th>System Element Isolation</th>
<th>Input checking / validation</th>
<th>SW load key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developmental CPI SW</td>
<td>30%</td>
<td>All</td>
<td>all</td>
<td>yes</td>
<td>All</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>Developmental Critical Function SW</td>
<td>50%</td>
<td>All</td>
<td>all</td>
<td>yes</td>
<td>All</td>
<td>All</td>
<td>all</td>
</tr>
<tr>
<td>Other Developmental SW</td>
<td>none</td>
<td>Partial</td>
<td>none</td>
<td>None</td>
<td>all</td>
<td>all</td>
<td>all</td>
</tr>
<tr>
<td>COTS (CPI and CF) and NDI SW</td>
<td>none</td>
<td>Partial</td>
<td>All</td>
<td>None</td>
<td>Wrappers/ all</td>
<td>all</td>
<td>all</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SW Product</th>
<th>Source</th>
<th>Release testing</th>
<th>Generated code inspection p/a</th>
</tr>
</thead>
<tbody>
<tr>
<td>C Compiler</td>
<td>No</td>
<td>Yes</td>
<td>50/20</td>
</tr>
<tr>
<td>Runtime libraries</td>
<td>Yes</td>
<td>Yes</td>
<td>70/none</td>
</tr>
<tr>
<td>Automated test system</td>
<td>No</td>
<td>Yes</td>
<td>50/none</td>
</tr>
<tr>
<td>Configuration management system</td>
<td>No</td>
<td>Yes</td>
<td>NA</td>
</tr>
<tr>
<td>Database</td>
<td>No</td>
<td>Yes</td>
<td>50/none</td>
</tr>
<tr>
<td>Development Environment Access</td>
<td>Controlled access; Cleared personnel only</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
FY 2012 FISMA Reporting Criteria

**AP Performance Areas:**
- Continuous Monitoring
- Automated Asset Management
- Automated Configuration Management
- Automated Vulnerability Management
- HSP0-12
- TiC v1.0 Capabilities
- TiC v2.0 Capabilities
- TiC Traffic Consolidation

**KFM Performance Areas:**
- Privileged User Training
- User Training
- Remote Access Authentication
- Remote Access Encryption
- DNSSEC Implementation
- Controlled Incident Detection
- US CERT SAR Remediation

**Base Performance Areas:**
- EINSTEIN 3 Status
  - Baseline questions are being asked to establish current performance, against which future performance may be measured.
  - Some of these questions are also intended to determine whether such future performance measures are needed.

- Administration Priorities (AP)
- Key FISMA Metrics (KFM)
- Baseline Questions (Base)

**GENERAL INSTRUCTIONS**

1. SYSTEM INVENTORY
2. ASSET MANAGEMENT
3. CONFIGURATION MANAGEMENT
4. VULNERABILITY AND WEAKNESS MANAGEMENT
5. IDENTITY AND ACCESS MANAGEMENT
6. DATA PROTECTION
7. BOUNDARY PROTECTION
8. INCIDENT MANAGEMENT
9. TRAINING AND EDUCATION
10. REMOTE ACCESS
11. NETWORK SECURITY PROTOCOLS

Prepared by:
US Department of Homeland Security
National Cyber Security Division
Federal Network Security

February 14, 2012
The revised milestone schedule reflects the ongoing work with the Joint Task Force (JTF) Transformation Initiative and the priorities established by the participating partners representing the Defense, Intelligence, and Civil communities of interest. In certain situations, selected publications have been slightly delayed due to an adjustment in priorities.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FIPS 199</td>
<td>Final</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIPS 200</td>
<td>Final</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP 800-18, Rev 2</td>
<td>RVC</td>
<td>RVC</td>
<td>RVC</td>
<td>RVC</td>
<td>RVC</td>
<td>RVC</td>
<td>RVC</td>
<td>RVC</td>
<td>RVC</td>
<td>1PD</td>
<td>RVC</td>
<td>FPD</td>
<td>RVC</td>
<td>Final</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP 800-30, Rev 1</td>
<td>JTF</td>
<td>RVC</td>
<td>RVC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Final</td>
<td></td>
</tr>
<tr>
<td>SP 800-37, Rev 1</td>
<td>JTF</td>
<td>Final</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP 800-39</td>
<td>JTF</td>
<td>Final</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP 800-53, Rev 3</td>
<td>JTF</td>
<td>Final</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP 800-53, Rev 4</td>
<td>JTF</td>
<td>Final</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP 800-53A, Rev 1</td>
<td>JTF</td>
<td>Final</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP 800-53A, Rev 2</td>
<td>JTF</td>
<td>Final</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP 800-59</td>
<td>Final</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP 800-60, Rev 1</td>
<td>Final</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP 800-137</td>
<td>Final</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**
- **1PD:** Initial public draft
- **FPD:** Final public draft
- **DEV:** Development cycle
- **RVC:** Revision cycle
- **JTF:** Joint Task Force Transformation Initiative

**FIPS PUB 199:** Standards for Security Categorization of Federal Information and Information Systems

**FIPS PUB 200:** Minimum Security Requirements for Federal Information and Information Systems

**SP 800-18, Revision 2:** Guide for Developing Security Plans for Federal Information Systems and Organizations

**SP 800-30, Revision 1:** Guide for Conducting Risk Assessments


**SP 800-39:** Managing Information Security Risk: Organization, Mission, and Information Systems View

**SP 800-53, Revision 4:** Recommended Security and Privacy Controls for Federal Information Systems and Organizations

**SP 800-60, Revision 1:** Guide for Mapping Types of Information and Information Systems to Security Categories

**SP 800-137:** Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations

(1) Publication refocused to address only risk assessments.
(2) Publication developed as part of the Joint Task Force Transformation Initiative (DOD, ODNI, CNSS, and NIST).
(3) Publication priority changed due to request from JTF partners, releasing the publication three months earlier than originally scheduled.
(4) Publication priority changed due to request from JTF partners, delaying publication until after the release on SP 800-30, Revision 1.
(5) Publication may be finalized in November 2012 (eliminating FPD), pending final decision by JTF partners.
(6) Publication schedule will be adjusted if SP 800-53, Revision 4, is published (final) in November.
DHS CS&C Software Assurance (SwA) Program

Advances security and resilience of software throughout the lifecycle; focuses on reducing exploitable software weaknesses and addresses means to improve capabilities that routinely develop, acquire, and deploy resilient software.

- Serves as a focal point for interagency public-private collaboration to enhance development and acquisition processes, capability benchmarking and rating schemes 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, cyber indicators and attacks which target software.
  - Collaborates with national & international standards organizations and industry to create standards, metrics and certification mechanisms from which products and tools could be qualified for software security verification.
  - Manages programs for Malware Attribute Enumeration Classification (MAEC), Common Weakness Enumeration (CWE), Common Attack Pattern (CAPEC) & Cyber Observable eXpression (CybOX).
  - Manages programs for Common Vulnerabilities & Exposures (CVE) and Open Vulnerability & Assessment Language (OVAL) that provide information feeds for continuous monitoring, security content automation, vulnerability databases, and security/threat alerts from many organizations.
DHS Software Assurance (SwA) Outreach & Awareness

- Co-sponsor SwA working group sessions, semi-annual SwA Forum, for government, academia, and industry to facilitate ongoing public-private collaboration.

- Provide SwA presentations, workshops, and tracks at conferences

- Co-sponsor SwA issues of CROSSTALK to “spread the word”
  - 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 “System Assurance”
  - Sep/Oct 2010 issue on “Game Changing Tools & Practices”
  - Mar/Apr 2011 issue on “Rugged Software”
  - Sep/Oct 2011 issue on “Protecting against Predatory Practices”
  - Mar/Apr 2012 issue on “Securing a Mobile World”
  - Sep/Oct 2012 issue on “Resilient Cyber Ecosystem”
  - Mar/Apr 2013 issue on “Supply Chain Risk Management”

- Collaborate with standards organizations, consortiums, professional societies, education/training initiatives in promoting SwA

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

- Host SwA Community Resources & Information Clearinghouse via https://buildsecurityin.us-cert.gov/SwA (since Dec 07)
SwA Collaboration for Content & Peer Review

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.
Enhancing the Development Life Cycle to Produce Secure Software


Executive commitment → SDL a mandatory policy at Microsoft since 2004

Ongoing Process Improvements → 6 month cycle

http://www.microsoft.com/sdl

Assurance for CMMI ®
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](https://buildsecurityin.us-cert.gov/swa) (see SwA Resources)
Architecture and Design Considerations for Secure Software – SwA Pocket Guide*

The IEEE Guide to the Software Engineering Body of Knowledge (SWEBOK) defines the design phase as both “the process of defining the architecture, components, interfaces, and other characteristics of a system or component” and “the result of [that] process.” The software design phase:

- is the software engineering life cycle activity where software requirements are analyzed in order to produce a description of the software’s internal structure that will serve as the basis for its implementation.
- consists of the architectural design and detailed design activities that follow the software requirements analysis phase and precedes software implementation in the SDLC.

This pocket guide includes the following topics:

- Basic Concepts
- Design Principles for Secure Software
- Architecture and Threat Modeling
- Secure Design Patterns
  - Architectural-level Patterns
  - Design-level Patterns
- Secure Session Management
- Design and Architectural Considerations for Mobile Applications
- Formal Methods and Architectural Design
- Design Review and Verification
- Key Architecture and Design Practices for Mitigating Exploitable Software Weaknesses
- Questions to Ask Developers

*Download FREE SwA Pocket Guides at https://buildsecurityin.us-cert.gov/swa

Software security testing is not the same as testing the correctness and adequacy of security functions implemented by software, which are most often verified through requirements-based testing that:

- cannot fully demonstrate that software is free from exploitable weaknesses / vulnerabilities.
- is not the best approach to determining how software will behave under anomalous and hostile conditions.

Penetration Testing can enhance pre-deployment test outcomes and identify post-release exploit points.
Secure Coding

- Preparing to Write Secure Code
- Secure Coding Principles
- Secure Coding Practices
- Secure Memory and Cache Management
- Secure Error and Exception Handling
- What to Avoid
- Questions to Ask Developers

Are any compiler warnings disabled in code being delivered?
Key Practices for Mitigating the Most Egregious Exploitable Software Weaknesses

- Identifies mission/business risks attributable to the respective weaknesses; identifies common attacks that exploit those weaknesses, and provides recommended practices for preventing the weaknesses.

- CWE focuses on stopping vulnerabilities at the source by educating designers, programmers, and QA/testers on how to eliminate all too-common mistakes before software is even shipped.

- CWE Top-N lists serve as tools for education, training and awareness to help programmers prevent the kinds of vulnerabilities that plague the software industry.

- Software consumers could use the same list to help them to ask for more secure software.

- Software managers and CIOs can use the CWE list as a measuring stick of progress in their efforts to secure their software.

Understand Assurance - Related Process Capability Expectations

Look to Standards for Assurance Process Detail

Mission/Business Process
Understand Your Business Requirements for Assurance

Measure Your Results

Information System
Build or Refine and Execute Your Assurance Processes

Organization Support

Understand Assurance-Related Process Capability Expectations

Look to Standards for Assurance Process Detail

Adapted from: Paul Croll, Computer Sciences Corporation, August 2007
Assurance Process Reference Model

**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.

**Prioritize funds and manage risks**

- **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

- **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

**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](https://buildsecurityin.us-cert.gov/swa/proself_assm.html)
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

The Center for Internet Security

The CIS Security Metrics

February 2009

Organizations struggle to make cost-effective security investment decisions. Information security professionals lack widely accepted and unambiguous metrics for decision support. CIS established a consensus team of over one hundred (100) industry experts to address this need. The result is a set of standard metrics and data definitions that can be used among organizations to collect and analyze data on security program performance and outcomes.

This document contains twenty-one (21) metrics definitions for six (6) important business functions: Incident Management, Vulnerability Management, Patch Management, Application Security, Configuration Management and Financial Metrics. Additional consensus metrics are currently being defined for these and additional business functions.

© 2008 The Center for Internet Security
Software Assurance Curriculum Project

• **Vol I: Master of Software Assurance Reference Curriculum**
  
  In Dec 2010 the IEEE Computer Society and the ACM recognized the Master of Software Assurance (MSwA) Reference Curriculum as a certified master’s degree program in SwA—the first curriculum to focus on assuring the functionality, dependability, and security of software and systems.

• **Vol II: SwA Undergraduate Course Outlines**
  
  see [www.sei.cmu.edu/library/abstracts/reports/10tr019.cfm](http://www.sei.cmu.edu/library/abstracts/reports/10tr019.cfm) to download the PDF version of the report CMU/SEI-2010-TR-019

• **Vol III: Master of SwA Course Syllabi**

• **Vol IV: Community College Education**

  • Report on “Integrating the MSwA Reference Curriculum into Model Curriculum and Guidelines for Graduate Degree Programs in Information Systems” provides reference and guidance material.

  • To facilitate implementation, the MSwA project team is offering assistance, free of charge, to educational institutions looking to launch an MSwA degree program.

  • For more information, go to [https://buildsecurityin.us-cert.gov/bsi/1165-BSI.html](https://buildsecurityin.us-cert.gov/bsi/1165-BSI.html).
Software Assurance Professional Competency Model

Specialty Areas*

• Software Assurance & Security Engineering
• Information Assurance Compliance
• Vulnerability Assessment & Management
• Cyber Threat Analysis
• Systems Requirements Planning
• Systems Security Architecture
• Strategic Planning & Policy Development
• Technology Research and Development
• Education and Training
• Knowledge Management

* Specialty Areas aligned with Framework for National Initiative for Cybersecurity Education
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
“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.

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

1. Insert and enforce software assurance requirements in contracts.
2. Review IT security policies to ensure that all users of organizational networks and data comply with the strictest security policies possible with respect to the mission.
3. Determine how much risk the organization can afford and who is accountable for that risk.
Even after vulnerabilities are discovered and patches made available, many developers use the flawed, non-patched version of reused components.

Who makes risk decisions?

Who inherits the residual risk?

Who ‘owns’ the residual risk attributable to exploitable software?

Source: Maximizing Benefits and Mitigating Risks of Open Source Components in Application Development, by Sonatype
Security-Enhanced Process Improvements

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

“Build Security In” throughout the lifecycle

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” (HP Fortify Software & Accenture Security Technology Consulting)
Objectives for SCRM & SwA in Acquisition

- we need “systems-of-systems” or “enterprise systems” thinking for risk management (building on 800-39 and 800-64)
- IT Baselines for SCRM are different, but should build on 800-53

Secure It Or Don’t Procure It

1. Framing
2. Preconditions
3. Distributed Enhancements
4. Integrated Activities, 5. Measurement & Monitoring
SwA Acquisition & Outsourcing Handbook

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>
<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>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>
15 What are the procedures used to approve, grant, monitor, and revoke file permissions for production data and executable code?

11 What are the agents or scripts executing on servers of hosted applications? Are there procedures for reviewing the security of these scripts or agents?

12 What are the procedures and policies used to approve, grant, monitor and revoke access to the servers? Are audit logs maintained?

7 What are the data backup policies and procedures? How frequently are the backup procedures verified?

<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>
</tbody>
</table>

5 Who configures and deploys the servers? Are the configuration procedures available for review, including documentation for all registry settings?

11 What are the agents or scripts executing on servers of hosted applications? Are there procedures for reviewing the security of these scripts or agents?

12 What are the procedures and policies used to approve, grant, monitor and revoke access to the servers? Are audit logs maintained?

13 What are the procedures and policies for handling and destroying sensitive data on electronic and printed media?

15 What are the procedures used to approve, grant, monitor, and revoke file permissions for production data and executable code?

---

**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>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>
</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>
</tbody>
</table>

| 5   | Who configures and deploys the servers? Are the configuration procedures available for review, including documentation for all registry settings? |
| 7   | What are the data backup policies and procedures? How frequently are the backup procedures verified? |
| 11  | What are the agents or scripts executing on servers of hosted applications? Are there procedures for reviewing the security of these scripts or agents? |
| 12  | What are the procedures and policies used to approve, grant, monitor and revoke access to the servers? Are audit logs maintained? |
| 13  | What are the procedures and policies for handling and destroying sensitive data on electronic and printed media? |
| 15  | What are the procedures used to approve, grant, monitor, and revoke file permissions for production data and executable code? |
Software Assurance

Software Assurance (SwA) is the level of confidence that software functions as intended and is free from vulnerabilities, either intentionally or unintentionally designed or inserted as part of the software throughout the life cycle.*

*Derived From: CNSSI-4009

Automation

Languages, enumerations, registries, tools, and repositories throughout the Lifecycle

Including design, coding, testing, deployment, configuration and operation
Automation is one piece of the SwA puzzle.
Many DHS, DoD, and NIST sponsored efforts are key to changing how software-based systems are developed, deployed & operated securely. These are (or are becoming used in) international standards.
Making Security Measurable (MSM): You Are Here

Software Assurance  \rightarrow Design  \rightarrow Deploy  \rightarrow Test

Design  \rightarrow Build  \rightarrow Assess  \rightarrow Test  \rightarrow Deploy

Test  \rightarrow Design  \rightarrow Build  \rightarrow Assess  \rightarrow Test

Vulnerabilities  \rightarrow Exploits  \rightarrow Attacks  \rightarrow Malware

CWE, CAPEC, CWSS, CWRAF

CPE, CCE, OVAL, OCIL, XCCDF, AssetId, ARF

CVE, CWE, CAPEC, MAEC, CybOX, IODEF
Cyber Threats Emerged Over Time

- **1980’s**
  - Password guessing
  - Exploiting known vulnerabilities
  - Burglaries
  - Packet spoofing
  - Automated probes/scans
  - Network management diagnostics
  - Sniffers
  - GUI intruder tools

- **1990’s**
  - Executable code attacks (against browsers)
  - Automated widespread attacks
  - Automated probes/scans
  - Email propagation of malicious code
  - Widespread attacks on DNS infrastructure using NNTP to distribute attacks
  - Widespread attacks using FTP to distribute attacks

- **2000’s**
  - Binary encryption
  - Sophisticated command & control
  - Anti-forensic techniques
  - Diffuse spyware
  - Home users targeted
  - Distributed attack tools
  - Increase in wide-scale Trojan horse distribution
  - Increase in tailored worms
  - Widespread denial-of-service attacks
  - Techiques to analyze code for vulnerabilities without source code
  - Widespread denial-of-service attacks

- **2010’s**
  - Attack sophistication
  - Sophisticated techniques to analyze code for vulnerabilities without source code
  - Distributed attack tools
  - Increase in wide-scale Trojan horse distribution
  - Increase in tailored worms
  - Widespread denial-of-service attacks
  - Techiques to analyze code for vulnerabilities without source code
  - Attack sophistication
Solutions Also Emerged Over Time

- **1980’s**
  - Password guessing
  - Exploiting known vulnerabilities
  - Packet spoofing
  - Burglaries

- **1990’s**
  - Disabling audits
  - Internet social engineering attacks
  - GUI intruder tools
  - Executable code attacks (against browsers)
  - Automated widespread attacks
  - Automated probes/scans

- **2000’s**
  - Email propagation of malicious code
  - Widespread attacks on DNS infrastructure
  - Widespread attacks using NNTP to distribute attack
  - Automated widespread attacks
  - Automated probes/scans
  - Network mgmt. diagnostics

- **2010’s**
  - Sophisticated command & control techniques to analyze code for vulnerabilities without source code
  - Increase in widespread denial-of-service attacks
  - Increase in wide-scale Trojan horse distribution
  - Distributed attack tools
  - Home users targeted
  - Trixie

**Attack Sophistication**

- Diffuse spyware
- Anti-forensic techniques
- Windows-based remote controllable Trojans (Back Orifice)
- Email propagation of malicious code
- DDoS attacks
- Increase in tailored worms
Operational Enterprise Networks

Development & Sustainment Security Management Processes

Security Management Processes

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

Operations Security Management Processes

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

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

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

Operational Enterprise Networks

- Trust Management
- Enterprise IT Change Management
- Identity Management
- Centralized Reporting

Enterprise IT Asset Management
<table>
<thead>
<tr>
<th>Question</th>
<th>Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>What IT systems do I have in my enterprise?</td>
<td>CPE (Platforms)</td>
</tr>
<tr>
<td>What known vulnerabilities do I need to worry about?</td>
<td>CVE (Vulnerabilities)</td>
</tr>
<tr>
<td>What vulnerabilities do I need to worry about right now?</td>
<td>CVSS (Scoring System)</td>
</tr>
<tr>
<td>How can I configure my systems more securely?</td>
<td>CCE (Configurations)</td>
</tr>
<tr>
<td>How do I define a policy of secure configurations?</td>
<td>XCCDF (Configuration Checklists)</td>
</tr>
<tr>
<td>How can I be sure my systems conform to policy?</td>
<td>OVAL (Assessment Language)</td>
</tr>
<tr>
<td>How can I be sure the operation of my systems conforms to policy?</td>
<td>OCIL (Interactive Language)</td>
</tr>
<tr>
<td>What weaknesses in my software could be exploited?</td>
<td>CWE (Weaknesses)</td>
</tr>
<tr>
<td>What attacks can exploit which weaknesses?</td>
<td>CAPEC (Attack Patterns)</td>
</tr>
<tr>
<td>How can we recognize malware &amp; share that info?</td>
<td>MAEC (Malware Attributes)</td>
</tr>
<tr>
<td>What observable behavior might put my enterprise at risk?</td>
<td>CybOX (Cyber Observables)</td>
</tr>
<tr>
<td>What events should be logged, and how?</td>
<td>CEE (Events)</td>
</tr>
<tr>
<td>How can I aggregate assessment results?</td>
<td>ARF (Assessment Results)</td>
</tr>
<tr>
<td>Question</td>
<td>Answer</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>What IT systems do I have in my enterprise?</strong></td>
<td>• CPE (Platforms)</td>
</tr>
<tr>
<td><strong>What known vulnerabilities do I need to worry about?</strong></td>
<td>• CVE (Vulnerabilities)</td>
</tr>
<tr>
<td><strong>What vulnerabilities do I need to worry about right now?</strong></td>
<td>• CVSS (Scoring System)</td>
</tr>
<tr>
<td><strong>How can I configure my systems more securely?</strong></td>
<td>• CCE (Configurations)</td>
</tr>
<tr>
<td><strong>How do I define a policy of secure configurations?</strong></td>
<td>• XCCDF (Configuration Checklists)</td>
</tr>
<tr>
<td><strong>How can I be sure my systems conform to policy?</strong></td>
<td>• OVAL (Assessment Language)</td>
</tr>
<tr>
<td><strong>How can I be sure the operation of my systems conforms to policy?</strong></td>
<td>• OCIL (Interactive Language)</td>
</tr>
<tr>
<td><strong>What weaknesses in my software could be exploited?</strong></td>
<td>• CWE (Weaknesses)</td>
</tr>
<tr>
<td><strong>What attacks can exploit which weaknesses?</strong></td>
<td>• CAPEC (Attack Patterns)</td>
</tr>
<tr>
<td><strong>How can we recognize malware &amp; share that info?</strong></td>
<td>• MAEC (Malware Attributes)</td>
</tr>
<tr>
<td><strong>What observable behavior might put my enterprise at risk?</strong></td>
<td>• CybOX (Cyber Observables)</td>
</tr>
<tr>
<td><strong>What events should be logged, and how?</strong></td>
<td>• CEE (Events)</td>
</tr>
<tr>
<td><strong>How can I aggregate assessment results?</strong></td>
<td>• ARF (Assessment Results)</td>
</tr>
</tbody>
</table>

**Standardization Efforts leveraged by the Security Content Automation Protocol (SCAP)**
<table>
<thead>
<tr>
<th>Question</th>
<th>Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>What IT systems do I have in my enterprise?</td>
<td>• CPE (Platforms)</td>
</tr>
<tr>
<td>What known vulnerabilities do I need to worry about?</td>
<td>• CVE (Vulnerabilities)</td>
</tr>
<tr>
<td>What vulnerabilities do I need to worry about right now?</td>
<td>• CVSS (Scoring System)</td>
</tr>
<tr>
<td>How can I configure my systems more securely?</td>
<td>• CCE (Configurations)</td>
</tr>
<tr>
<td>How do I define a policy of secure configurations?</td>
<td>• XCCDF (Configuration Checklists)</td>
</tr>
<tr>
<td>How can I be sure my systems conform to policy?</td>
<td>• OVAL (Assessment Language)</td>
</tr>
<tr>
<td>How can I be sure the operation of my systems conforms to policy?</td>
<td>• OCIL (Interactive Language)</td>
</tr>
<tr>
<td>What weaknesses in my software could be exploited?</td>
<td>• CWE (Weaknesses)</td>
</tr>
<tr>
<td>What attacks can exploit which weaknesses?</td>
<td>• CAPEC (Attack Patterns)</td>
</tr>
<tr>
<td>How can we recognize malware &amp; share that info?</td>
<td>• MAEC (Malware Attributes)</td>
</tr>
<tr>
<td>What observable behavior might put my enterprise at risk?</td>
<td>• CybOX (Cyber Observables)</td>
</tr>
<tr>
<td>What events should be logged, and how?</td>
<td>• CEE (Events)</td>
</tr>
<tr>
<td>How can I aggregate assessment results?</td>
<td>• ARF (Assessment Results)</td>
</tr>
</tbody>
</table>

Efforts focused on mitigating risks and enabling more robust continuous monitoring and faster incident response.
Evolution of Standardized Representations - Sharing

Vulnerabilities
Weaknesses
Attack Patterns
Malware Behavior
Cyber Observables

? Threat Indicators

Imports & Extends:
- Object
- Defined Objects
- Actions

MAEC
CAPEC
CybOX

Malware
Attack Patterns
Log Events
What is a cyber observable?

- a **measurable event** or **stateful property** in the cyber domain

  - Some measurable events: a registry key is created, a file is deleted, an http GET is received, …
  
  - Some stateful properties: MD5 hash of a file, value of a registry key, existence of a mutex, …

Cyber Observable eXpression (CybOX) is a standardized language for encoding and communicating information about cyber observables ([http://cybox.mitre.org](http://cybox.mitre.org))
What is STIX™

Structured Threat Information eXpression

Language

Specify  Capture  Characterize  Communicate

Cyber Threat Information

Community-driven

Consistency  Clarity  Support automation
Structuring Threat Information for Sharing

Why were they doing it?

Why should you care about it?

What exactly were they doing?

What were they looking to exploit?

Who was doing it?

What you are looking for

Where was it seen?

What should you do about it?
• Org C must understand each format in use and try to map across formats – sacrificing time and potentially losing information
• Duplication of effort at each organization in the exchange is expensive and does not scale
Enabling Cross-Vendor Sharing

• Org C only needs to understand one format – no need to map and no information loss
• Each vendor maps their internal representations to the common format once – efficient and scalable
SwA Working Group Sessions: 27-29 Nov 2012 @ MITRE in McLean, VA

SwA Forum – Next session: 5-7 Mar 2013 @ NIST in Gaithersburg, MD

SwA Websites: www.us-cert.gov/swa

Email: software.assurance@dhs.gov

Making Security Measureable: measurablesecurity.mitre.org

See Language for sharing exchange of indicators and correlation of incident information -- Cyber Observables eXpression (CybOX) at http://cybox.mitre.org
Public/Private Collaboration Efforts for Security Automation and Software Supply Chain Risk Management

Next SwA Working Groups sessions: 27-29 Nov 2012 at MITRE, McLean, VA
SOFTWARE ASSURANCE FORUM

“Building Security In”
https://buildsecurityin.us-cert.gov/swa

Joe Jarzombek, PMP, CSSLP
Director for Software Assurance
Cyber Security & Communications
Department of Homeland Security
Joe.Jarzombek@hq.dhs.gov
(703) 235-3673