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

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

Homeland Security

Commerce

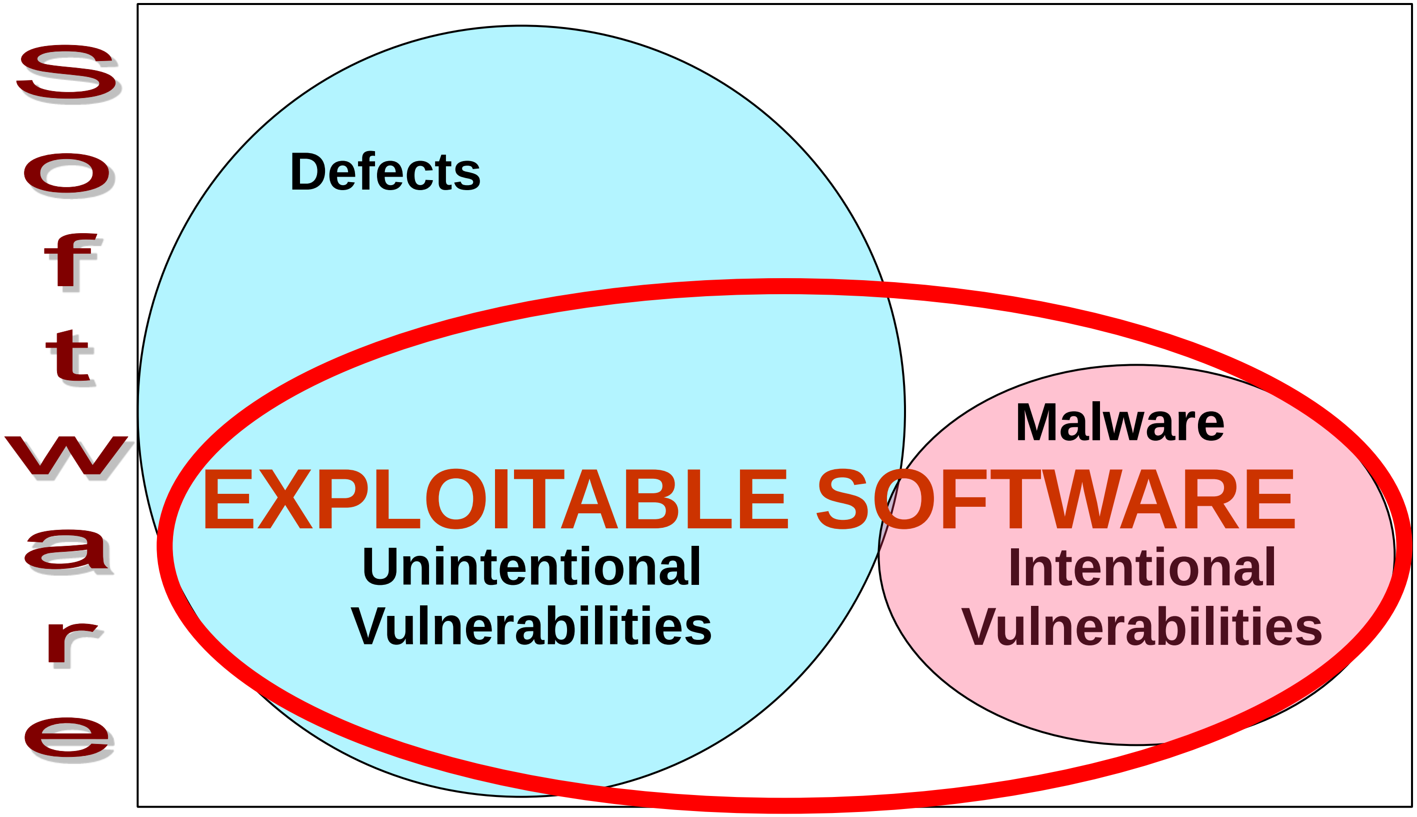
National Defense

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"



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

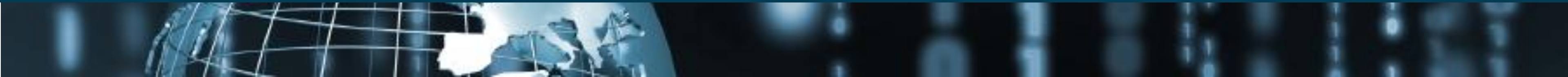
*Intentional vulnerabilities: spyware & malicious logic deliberately imbedded (might not be considered defects)

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.



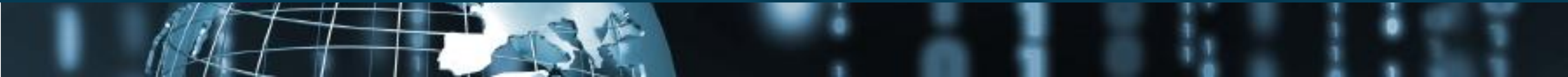
Challenges in Mitigating Risks Attributable to Exploitable Software and Supply Chains

- 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.

Challenges in Mitigating Risks Attributable to Exploitable Software and Supply Chains (cont.)

Enterprises seek comprehensive capabilities to:

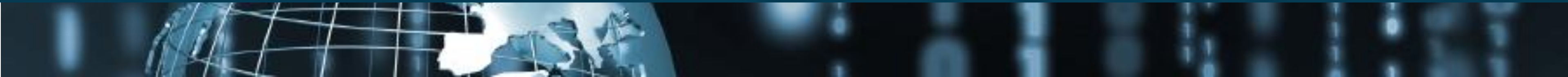
- Avoid installing software with **MALWARE** pre-installed. **MAEC**
- Determine that no publicly reported **VULNERABILITIES** **CVE** remain in code prior to operational acceptance, and that future discoveries of common vulnerabilities and exposures can be quickly patched.
- Determine that exploitable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

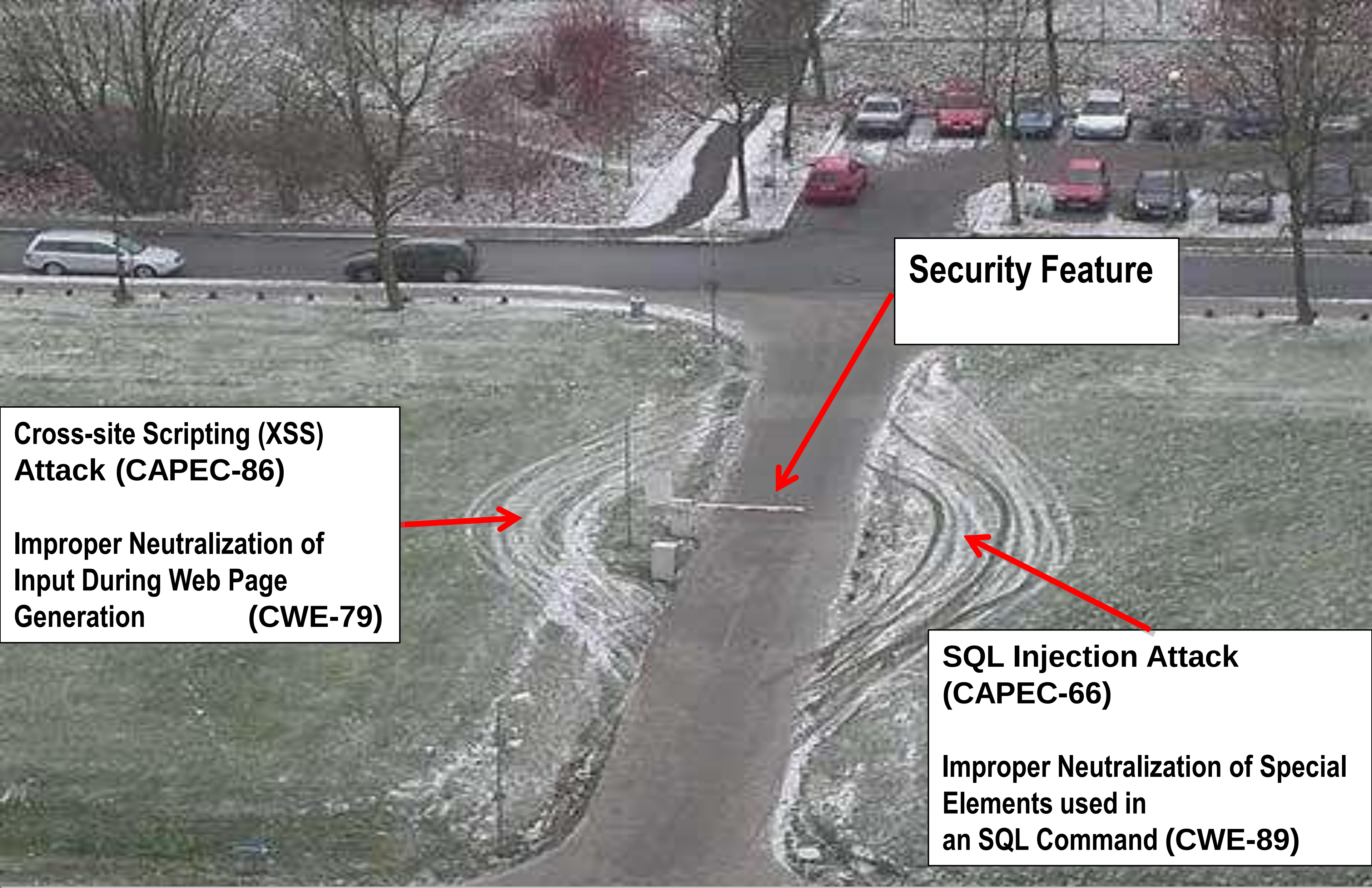


Challenges in Preventing and Responding to Cyber Incidents

- “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.





Security Feature

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)

Exploitable Software Weaknesses (CWEs) are sources for future Zero-Day Attacks

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:

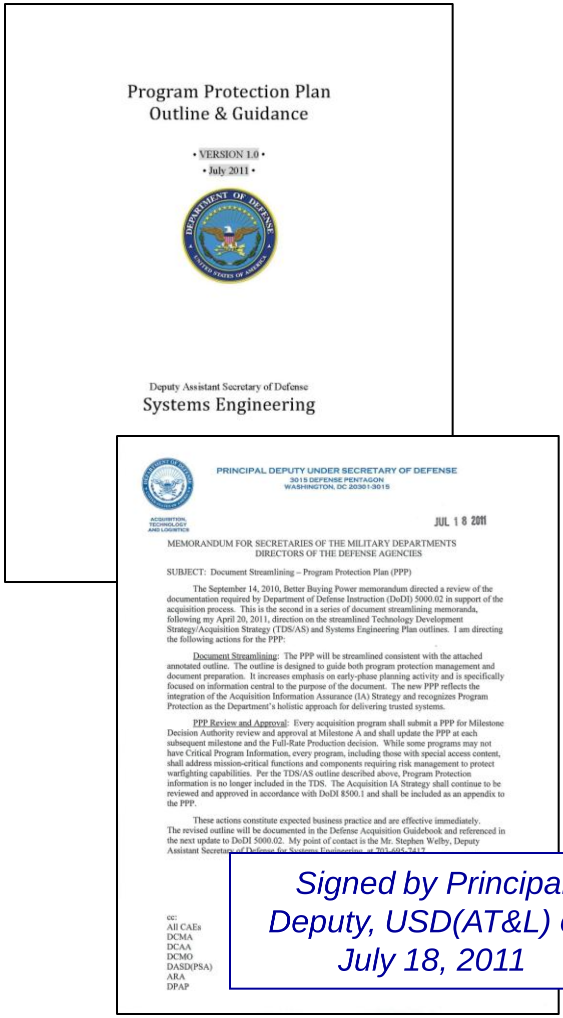
1. Implement software configurations such as the U.S. Government Configuration Baseline (formerly the Federal Desktop Core Configuration), strong authentication, and strict, documented internal policies and procedures.
2. Ask vendors to provide guarantees of software security as required by HR 6523.
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.

Building software without accounting for security is no longer an acceptable risk.

"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

Program Protection Plan Outline and Guidance as "Expected Business Practice"



Program Protection Plan
Outline & Guidance

• VERSION 1.0 •
• July 2011 •

Deputy Assistant Secretary of Defense
Systems Engineering

PRINCIPAL DEPUTY UNDER SECRETARY OF DEFENSE
3015 DEFENSE PENTAGON
WASHINGTON, DC 20301-3015

ACQUISITION, TECHNOLOGY AND LOGISTICS

JUL 18 2011

MEMORANDUM FOR SECRETARIES OF THE MILITARY DEPARTMENTS
DIRECTORS OF THE DEFENSE AGENCIES

SUBJECT: Document Streamlining – Program Protection Plan (PPP)

The September 14, 2010, Better Buying Power memorandum directed a review of the documentation required by Department of Defense Instruction (DoDI) 5000.02 in support of the acquisition process. This is the second in a series of document streamlining memoranda, following my April 20, 2011, direction on the streamlined Technology Development Strategy/Acquisition Strategy (TDS/AS) and Systems Engineering Plan outlines. I am directing the following actions for the PPP:

Document Streamlining: The PPP will be streamlined consistent with the attached annotated outline. The outline is designed to guide both program protection management and document preparation. It increases emphasis on early-phase planning activity and is specifically focused on information central to the purpose of the document. The new PPP reflects the integration of the Acquisition Information Assurance (IA) Strategy and recognizes Program Protection as the Department's holistic approach for delivering trusted systems.

PPP Review and Approval: 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. While some programs may not have Critical Program Information, every program, including those with special access content, shall address mission-critical functions and components requiring risk management to protect warfighting capabilities. Per the TDS/AS outline described above, Program Protection information is no longer included in the TDS. The Acquisition IA Strategy shall continue to be reviewed and approved in accordance with DoDI 8500.1 and shall be included as an appendix to the PPP.

These actions constitute expected business practice and are effective immediately. The revised outline will be documented in the Defense Acquisition Guidebook and referenced in the next update to DoDI 5000.02. My point of contact is the Mr. Stephen Welby, Deputy Assistant Secretary of Defense for Systems Engineering, at 703-695-7417.

cc:
All CAEs
DCMA
DCAA
DCMO
DASD(PSA)
ARA
DPAP

Signed by Principal Deputy, USD(AT&L) on July 18, 2011

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

Distribution Statement A – Cleared for public release by OSR on 4/25/2012, SR Case # 12-S-1841 applies.

Software Assurance Methods

Counter-measure Selection

Development Process

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

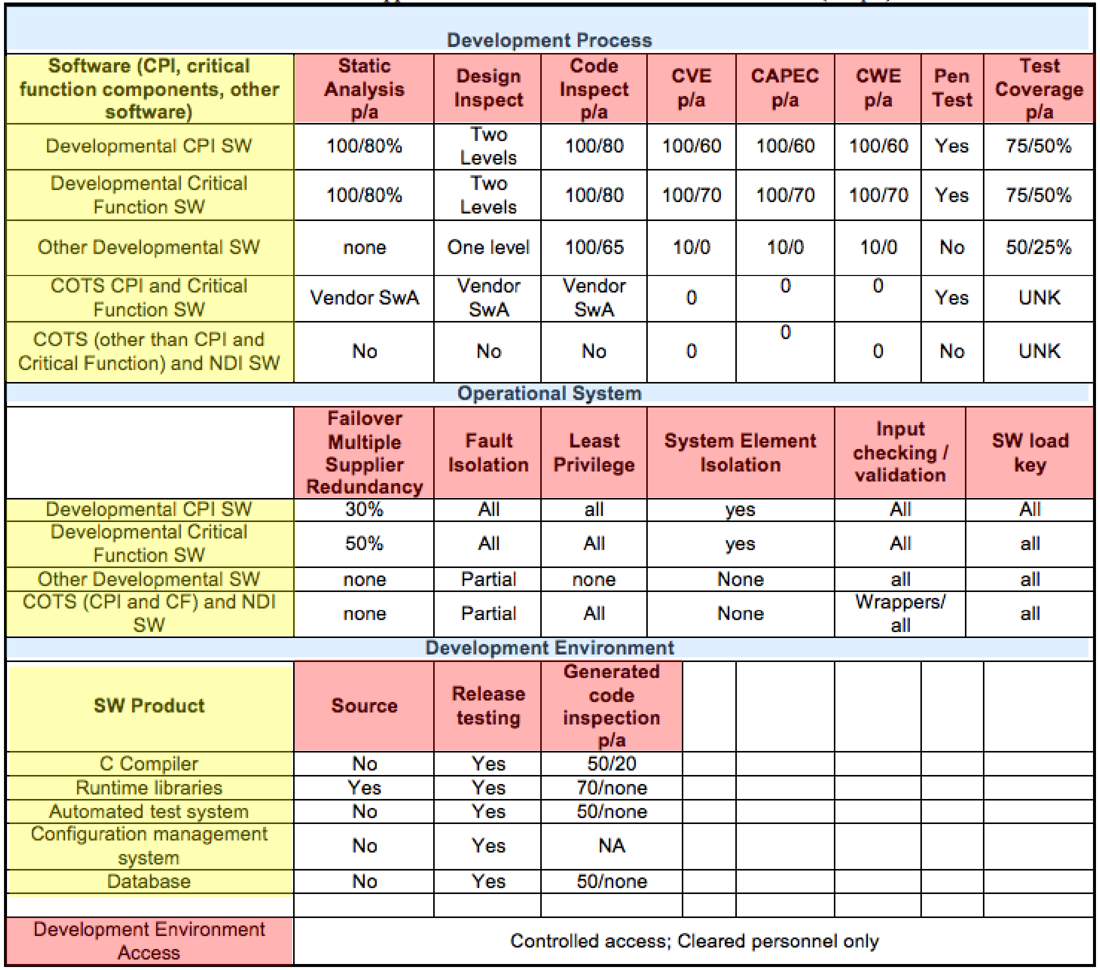
Operational System

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

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



Development Process

Software (CPI, critical function components, other software)	Static Analysis p/a	Design Inspect	Code Inspect p/a	CVE p/a	CAPEC p/a	CWE p/a	Pen Test	Test Coverage p/a
Developmental CPI SW	100/80%	Two Levels	100/80	100/60	100/60	100/60	Yes	75/50%
Developmental Critical Function SW	100/80%	Two Levels	100/80	100/70	100/70	100/70	Yes	75/50%
Other Developmental SW	none	One level	100/65	10/0	10/0	10/0	No	50/25%
COTS CPI and Critical Function SW	Vendor SwA	Vendor SwA	Vendor SwA	0	0	0	Yes	UNK
COTS (other than CPI and Critical Function) and NDI SW	No	No	No	0	0	0	No	UNK

Operational System

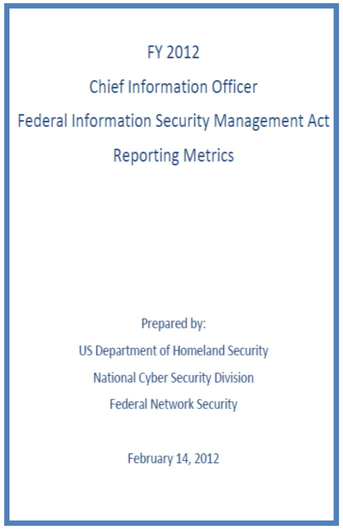
	Failover Multiple Supplier Redundancy	Fault Isolation	Least Privilege	System Element Isolation	Input checking / validation	SW load key
Developmental CPI SW	30%	All	all	yes	All	All
Developmental Critical Function SW	50%	All	All	yes	All	all
Other Developmental SW	none	Partial	none	None	all	all
COTS (CPI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

Development Environment

SW Product	Source	Release testing	Generated code inspection p/a
C Compiler	No	Yes	50/20
Runtime libraries	Yes	Yes	70/none
Automated test system	No	Yes	50/none
Configuration management system	No	Yes	NA
Database	No	Yes	50/none
Development Environment Access	Controlled access; Cleared personnel only		

Additional Guidance in PPP Outline and Guidance

FY 2012 FISMA Reporting Criteria



FY 2012

Chief Information Officer

Federal Information Security Management Act

Reporting Metrics

Prepared by:

US Department of Homeland Security

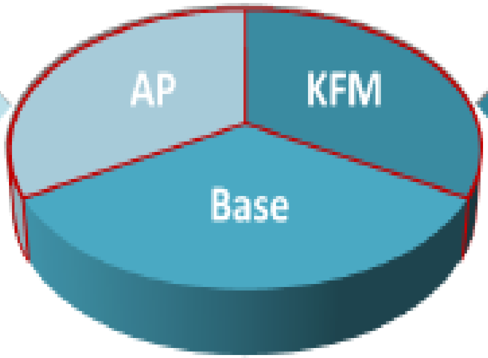
National Cyber Security Division

Federal Network Security

February 14, 2012

AP Performance Areas:

- Continuous Monitoring
  - -Automated Asset Management
  - -Automated Configuration Management
  - -Automated Vulnerability Management
- HSPD-12
- TIC v1.0 Capabilities
- TIC v2.0 Capabilities
- TIC Traffic Consolidation



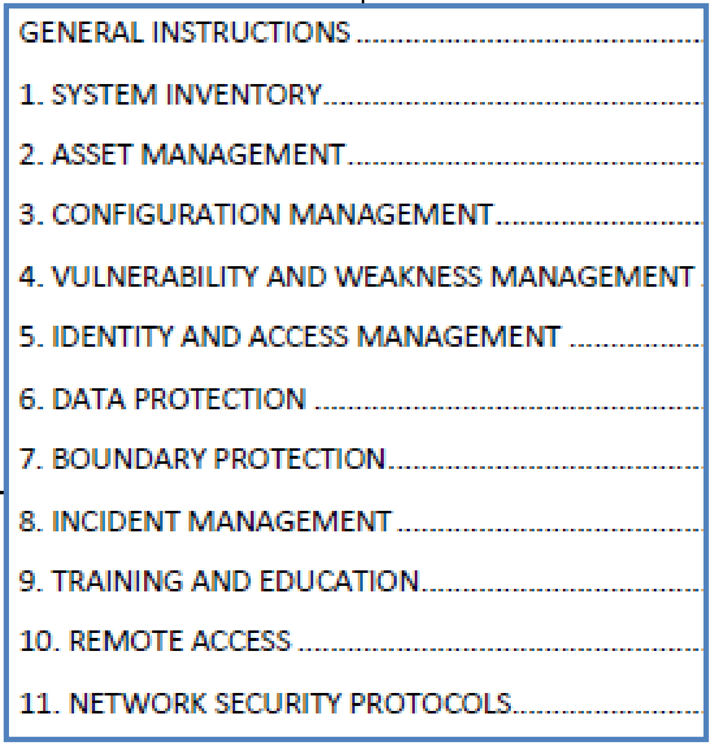
KFM Performance Areas:

- 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 perfomance, against which future perfomance may be measured
- Some of these questions are also intended to determine whether such future performance measures are needed



- Administration Priorities[1] (AP)
- Key FISMA Metrics[2] (KFM)
- Baseline Questions[3] (Base)

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

DEVELOPMENT SCHEDULE FOR FISMA IMPLEMENTATION PROJECT PUBLICATIONS

As of August 20, 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.*

		Jul 2012	Aug 2012	Sep 2012	Oct 2012	Nov 2012	Dec 2012	Jan 2013	Feb 2013	Mar 2013	Apr 2013	May 2013	Jun 2013	July 2013	Aug 2013	Sep 2013	Oct 2013
FIPS 199	**Final**																
FIPS 200	**Final**																
SP 800-18, Rev 2		**RVC**	**RVC**	**RVC**	**RVC**	**RVC**	**RVC**	**RVC**	**RVC**	**RVC**	**RVC**	**RVC**	**1PD**	**RVC**	**FPD**	**RVC**	**Final**
SP 800-30, Rev 1 *JTF*		**RVC**	**RVC**	**Final**													
SP 800-37, Rev 1 *JTF*	**Final**																
SP 800-39 *JTF*	**Final**																
SP 800-53, Rev 3 *JTF*	**Final**																
SP 800-53, Rev 4 *JTF*		**RVC**	**RVC**	**RVC**	**RVC**	**FPD**	**RVC**	**Final**									
SP 800-53A, Rev 1 *JTF*	**Final**																
SP 800-53A, Rev 2 *JTF*		**RVC**	**RVC**	**RVC**	**RVC**	**RVC**	**RVC**	**RVC**	**IPD**	**RVC**	**Final**						
SP 800-59	**Final**																
SP 800-60, Rev 1	**Final**																
SP 800-137	**Final**																

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* [1][2][3]
SP 800-37, Revision 1: *Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach* [2]
SP 800-39: *Managing Information Security Risk: Organization, Mission, and Information Systems View* [2]
SP 800-53, Revision 4: *Recommended Security and Privacy Controls for Federal Information Systems and Organizations* [2][4][5]
SP 800-53A, Revision 2: *Guide for Assessing the Security and Privacy Controls in Federal Information Systems and Organizations* [2][6]
SP 800-59: *Guideline for Identifying an Information System as a National Security System*
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)

CROSSTALK
The Journal of Defense Software Engineering
THE RUGGED SOFTWARE MANIFESTO
CROSST
March 2007 The Journal of Defense Softwa
THE FIREWALLS
ACCESS DENIED
SOFTWARE SECURITY
SPECIAL ISSUE: ENSURING SECURE SOFTWARE
CROSSTALK
October 2005 The Journal of Defense Software Engineering Vol. 18 No. 10
SOFTWARE SECURITY
CR
September 200
CRO
September 2006 The Journ
Application Security
SOFTWARE ASSURANCE

SwA Collaboration for Content & Peer Review

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

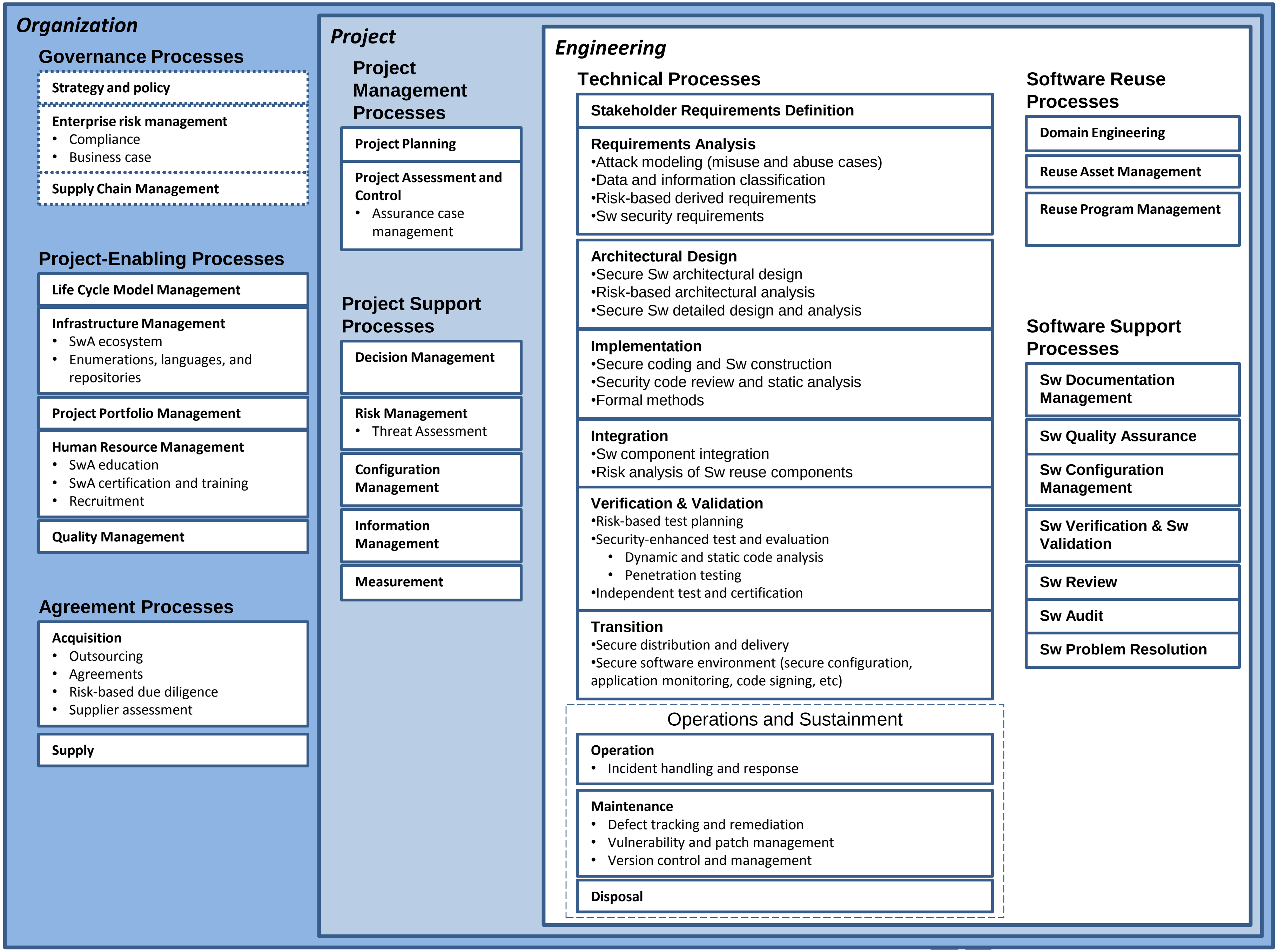
SwA Community Resources and Information Clearinghouse (CRIC)

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.

Life-Cycle Standards View Categories (ISO/IEC 15288 and 12207)



Organization

Governance Processes

Strategy and policy

Enterprise risk management
- Compliance
- Business case

Supply Chain Management

Project-Enabling Processes

Life Cycle Model Management

Infrastructure Management
- SwA ecosystem
- Enumerations, languages, and repositories

Project Portfolio Management

Human Resource Management
- SwA education
- SwA certification and training
- Recruitment

Quality Management

Agreement Processes

Acquisition
- Outsourcing
- Agreements
- Risk-based due diligence
- Supplier assessment

Supply

Project

Project Management Processes

Project Planning

Project Assessment and Control
- Assurance case management

Project Support Processes

Decision Management

Risk Management
- Threat Assessment

Configuration Management

Information Management

Measurement

Engineering

Technical Processes

Stakeholder Requirements Definition

Requirements Analysis
•Attack modeling (misuse and abuse cases)
•Data and information classification
•Risk-based derived requirements
•Sw security requirements

Architectural Design
•Secure Sw architectural design
•Risk-based architectural analysis
•Secure Sw detailed design and analysis

Implementation
•Secure coding and Sw construction
•Security code review and static analysis
•Formal methods

Integration
•Sw component integration
•Risk analysis of Sw reuse components

Verification & Validation
•Risk-based test planning
•Security-enhanced test and evaluation
- Dynamic and static code analysis
- Penetration testing

•Independent test and certification

Transition
•Secure distribution and delivery
•Secure software environment (secure configuration, application monitoring, code signing, etc)

Operations and Sustainment

Operation
- Incident handling and response

Maintenance
- Defect tracking and remediation
- Vulnerability and patch management
- Version control and management

Disposal

Software Reuse Processes

Domain Engineering

Reuse Asset Management

Reuse Program Management

Software Support Processes

Sw Documentation Management

Sw Quality Assurance

Sw Configuration Management

Sw Verification & Sw Validation

Sw Review

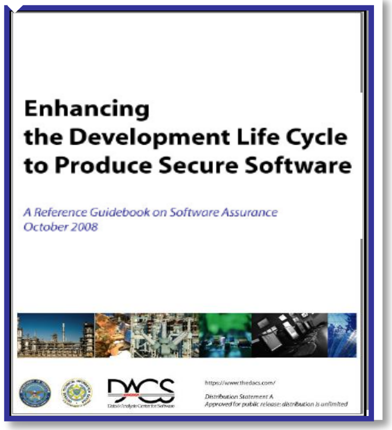
Sw Audit

Sw Problem Resolution

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

Many SwA Resources Focus On Development

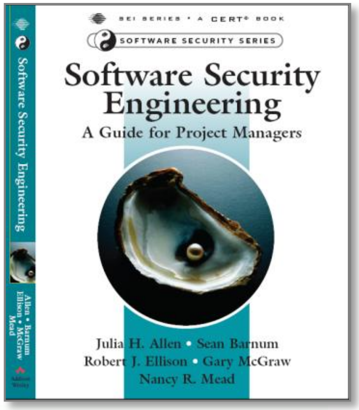

Enhancing the Development Life Cycle to Produce Secure Software

A Reference Guidebook on Software Assurance
October 2008

DACS

https://www.thedacs.com/

Distribution Statement A
Approved for public release: distribution is unlimited

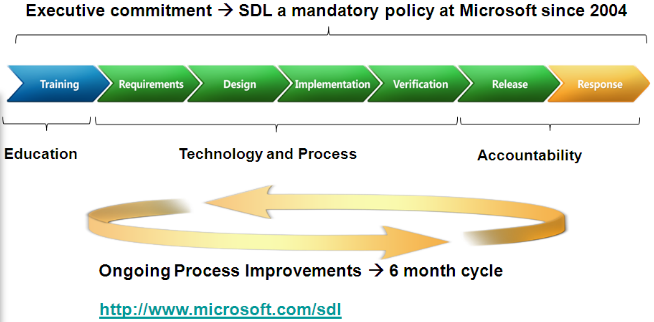

SEI SERIES • A CERT® BOOK

SOFTWARE SECURITY SERIES

Software Security Engineering

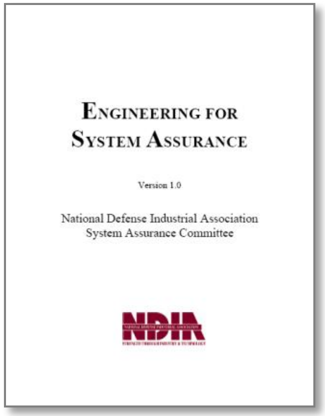
A Guide for Project Managers

Julia H. Allen • Sean Barnum
Robert J. Ellison • Gary McGraw
Nancy R. Mead


Executive commitment → SDL a mandatory policy at Microsoft since 2004

Ongoing Process Improvements → 6 month cycle

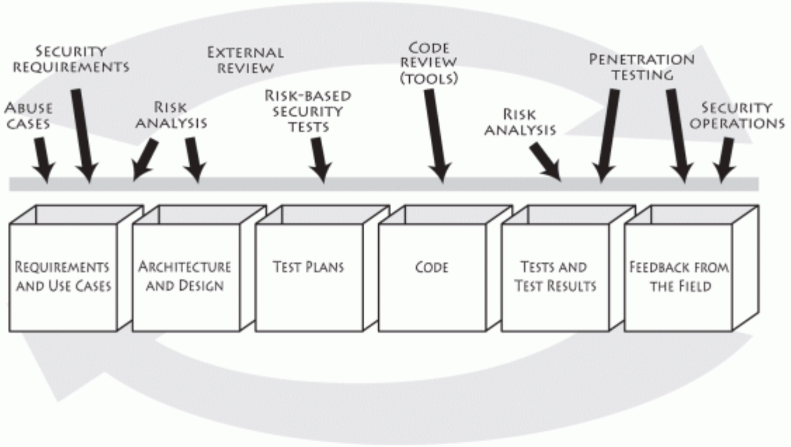
http://www.microsoft.com/sdl


ENGINEERING FOR SYSTEM ASSURANCE

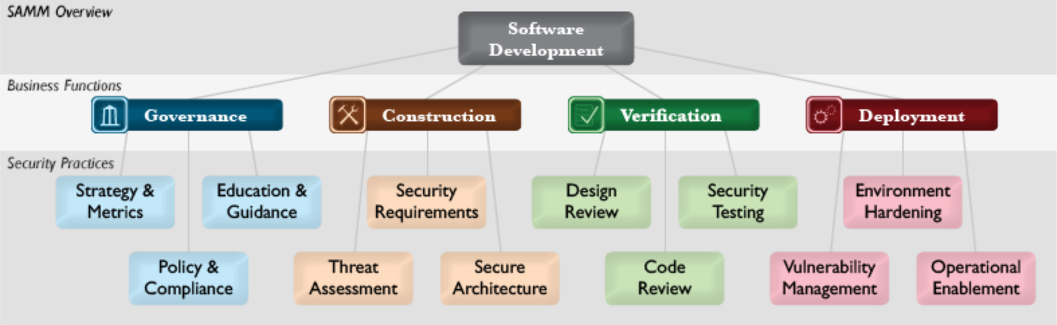
Version 1.0

National Defense Industrial Association
System Assurance Committee

NDIA



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

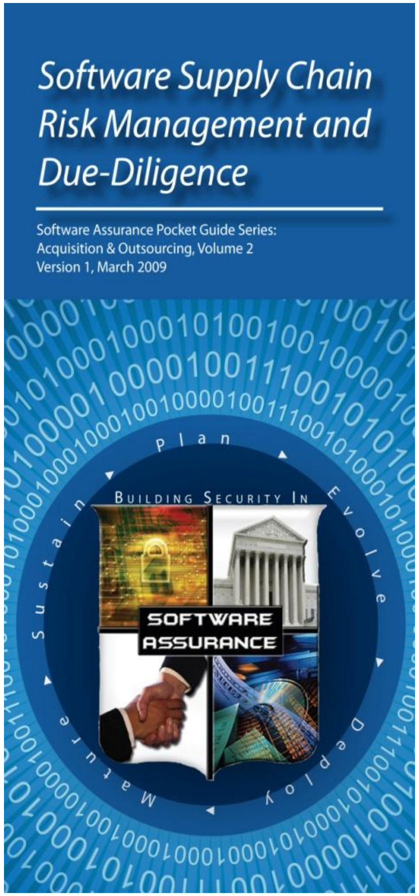
- Integrating Security into the Software Development Life Cycle
- Key Practices for Mitigating the Most Egregious Exploitable Software Weaknesses
- Risk-based Software Security Testing
- Requirements and Analysis for Secure Software
- Architecture and Design Considerations for Secure Software
- Secure 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



Software Supply Chain Risk Management and Due-Diligence

Software Assurance Pocket Guide Series:
Acquisition & Outsourcing, Volume 2
Version 1, March 2009

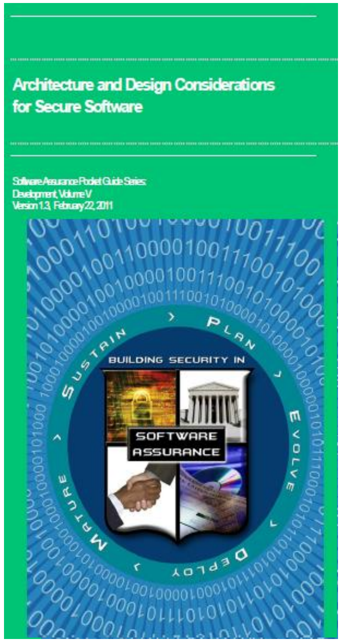
SwA Pocket Guides and SwA-related documents are collaboratively developed with peer review; they are subject to update and are freely available for download via the DHS Software Assurance Community Resources and Information Clearinghouse at https://buildsecurityin.us-cert.gov/swa (see SwA Resources)

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



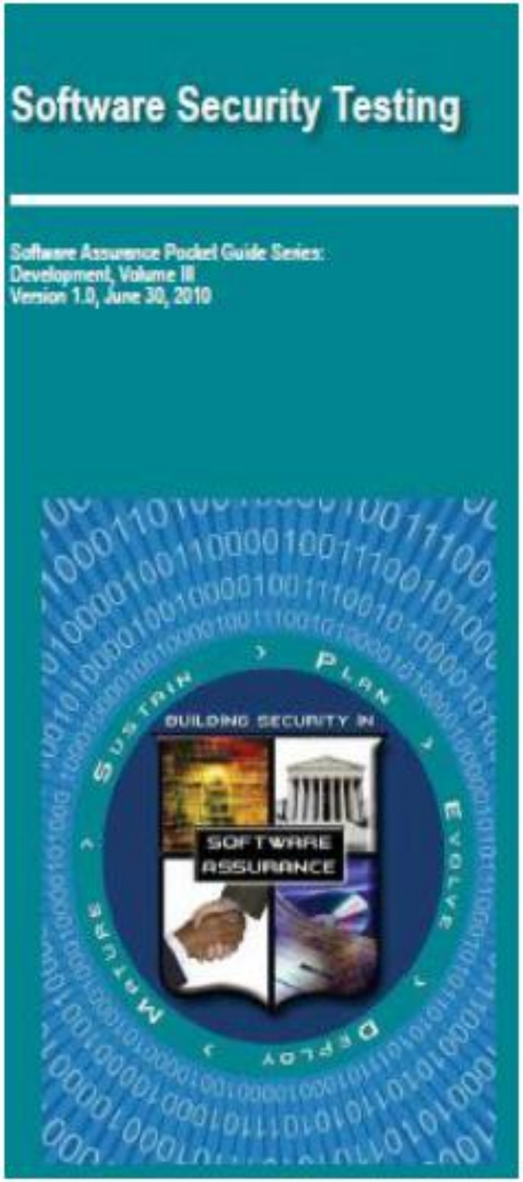
Architecture and Design Considerations for Secure Software

Software Assurance Pocket Guide Series:
Development, Volume V
Version 1.3, February 22, 2011

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

Software Security Testing –vs– Security Requirements Testing

- 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.
- Download FREE SwA Pocket Guide on Software Security Testing at https://buildsecurityin.us-cert.gov/swa



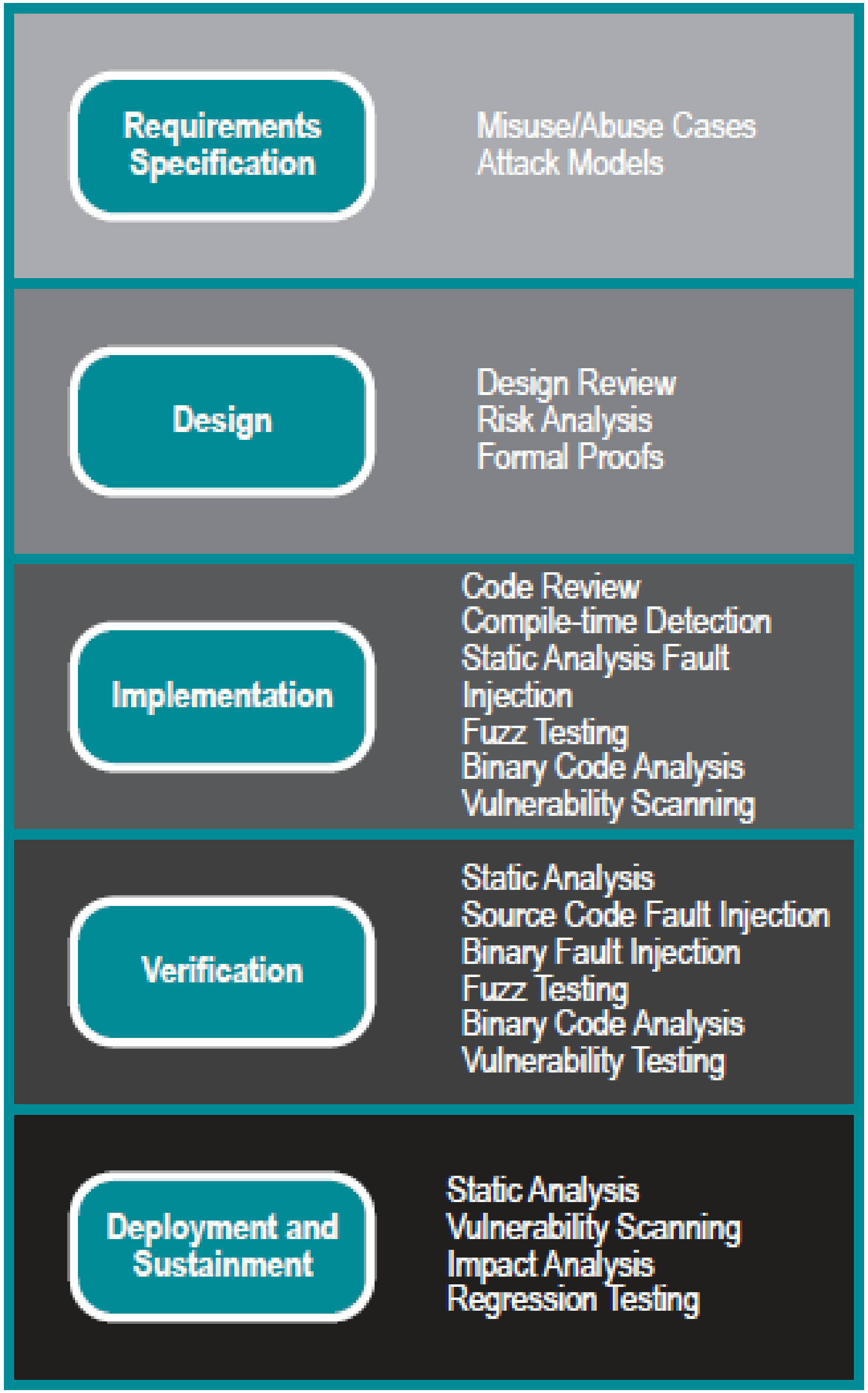
Software Security Testing

Software Assurance Pocket Guide Series:
Development, Volume III
Version 1.0, June 30, 2010

Homeland Security

Software Security Test Techniques throughout SDLC

See details in FREE SwA Pocket Guide on Software Security Testing at https://buildsecurityin.us-cert.gov/swa

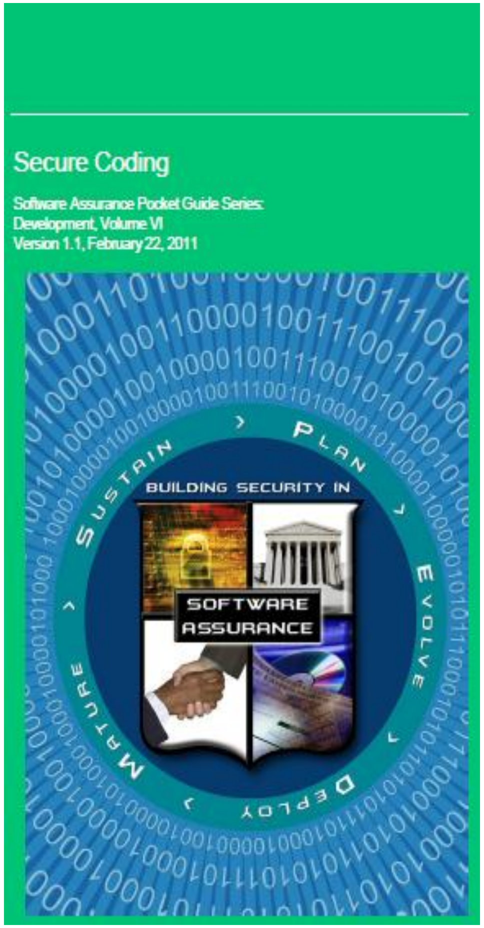


Requirements Specification	Misuse/Abuse Cases Attack Models
Design	Design Review Risk Analysis Formal Proofs
Implementation	Code Review Compile-time Detection Static Analysis Fault Injection Fuzz Testing Binary Code Analysis Vulnerability Scanning
Verification	Static Analysis Source Code Fault Injection Binary Fault Injection Fuzz Testing Binary Code Analysis Vulnerability Testing
Deployment and Sustainment	Static Analysis Vulnerability Scanning Impact Analysis Regression Testing

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



Secure Coding

Software Assurance Pocket Guide Series:
Development, Volume VI
Version 1.1, February 22, 2011

Are any compiler warnings disabled in code being delivered?

Homeland
Security

Key Practices for Mitigating the Most Egregious Exploitable Software Weaknesses

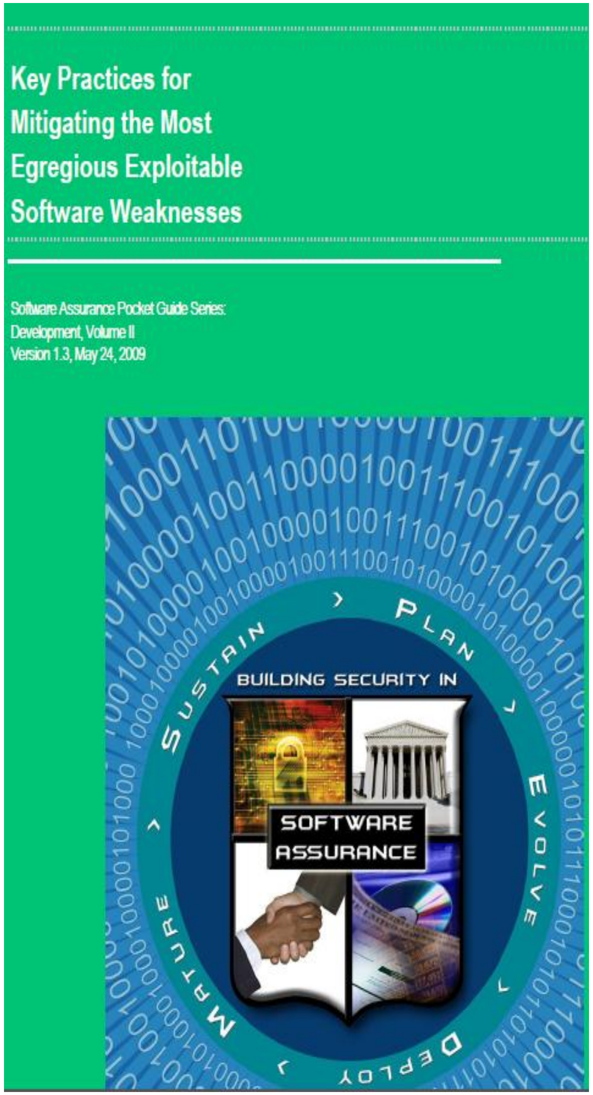
- *Identifies mission/business risks attributable to the respective weaknesses; identifies common attacks that exploit those weaknesses, and provides recommended practices for preventing the weaknesses.*

- 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.

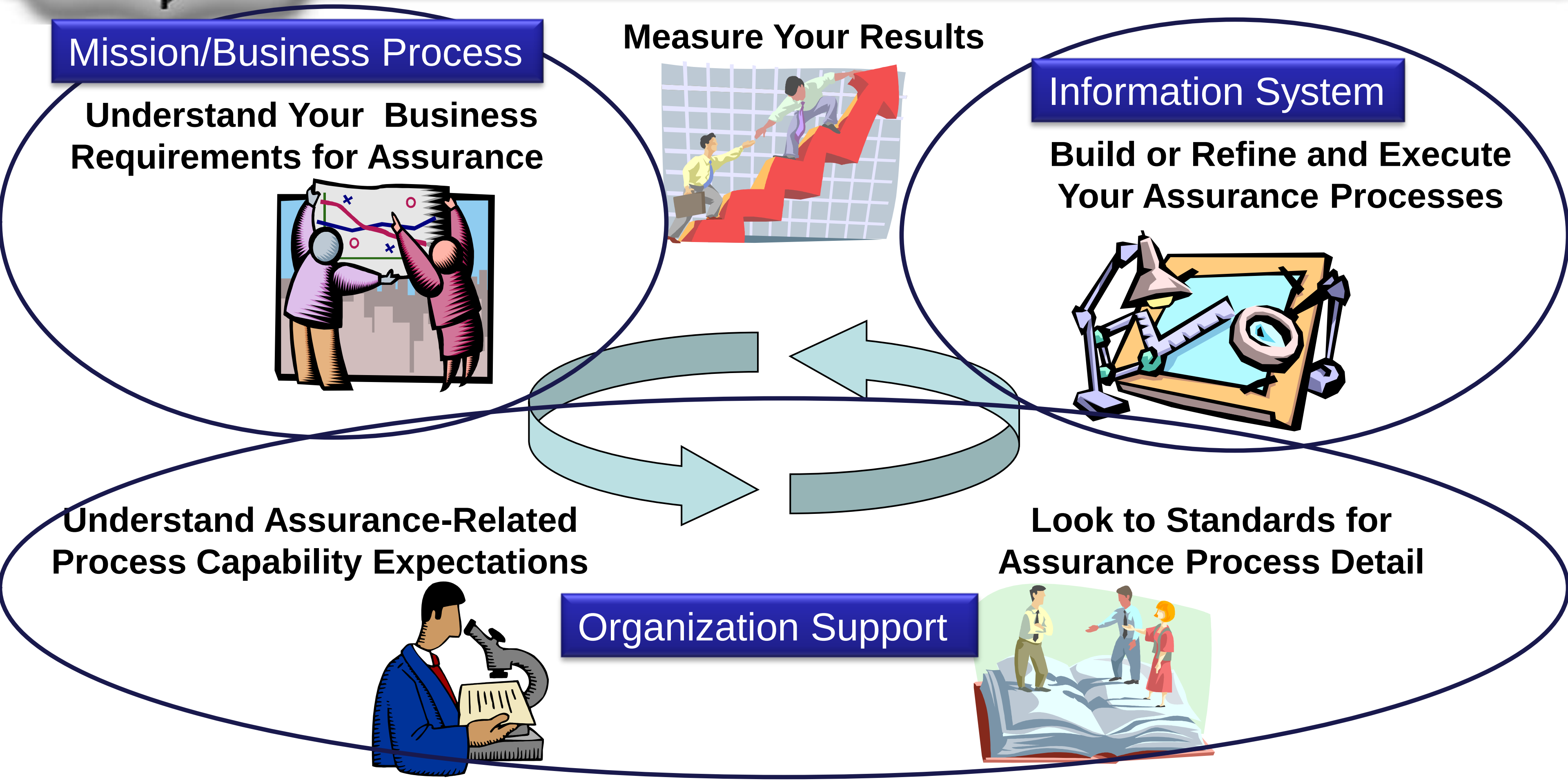


Key Practices for Mitigating the Most Egregious Exploitable Software Weaknesses

Software Assurance Pocket Guide Series:
Development, Volume II
Version 1.3, May 24, 2009

https://buildsecurityin.us-cert.gov/swa/downloads/KeyPracticesMWV13_02AM091111.pdf

Process Improvement Lifecycle - A Process for Achieving Assurance



Mission/Business Process

Understand Your Business Requirements for Assurance

Measure Your Results

Information System

Build or Refine and Execute Your Assurance Processes

Understand Assurance-Related Process Capability Expectations

Look to Standards for Assurance Process Detail

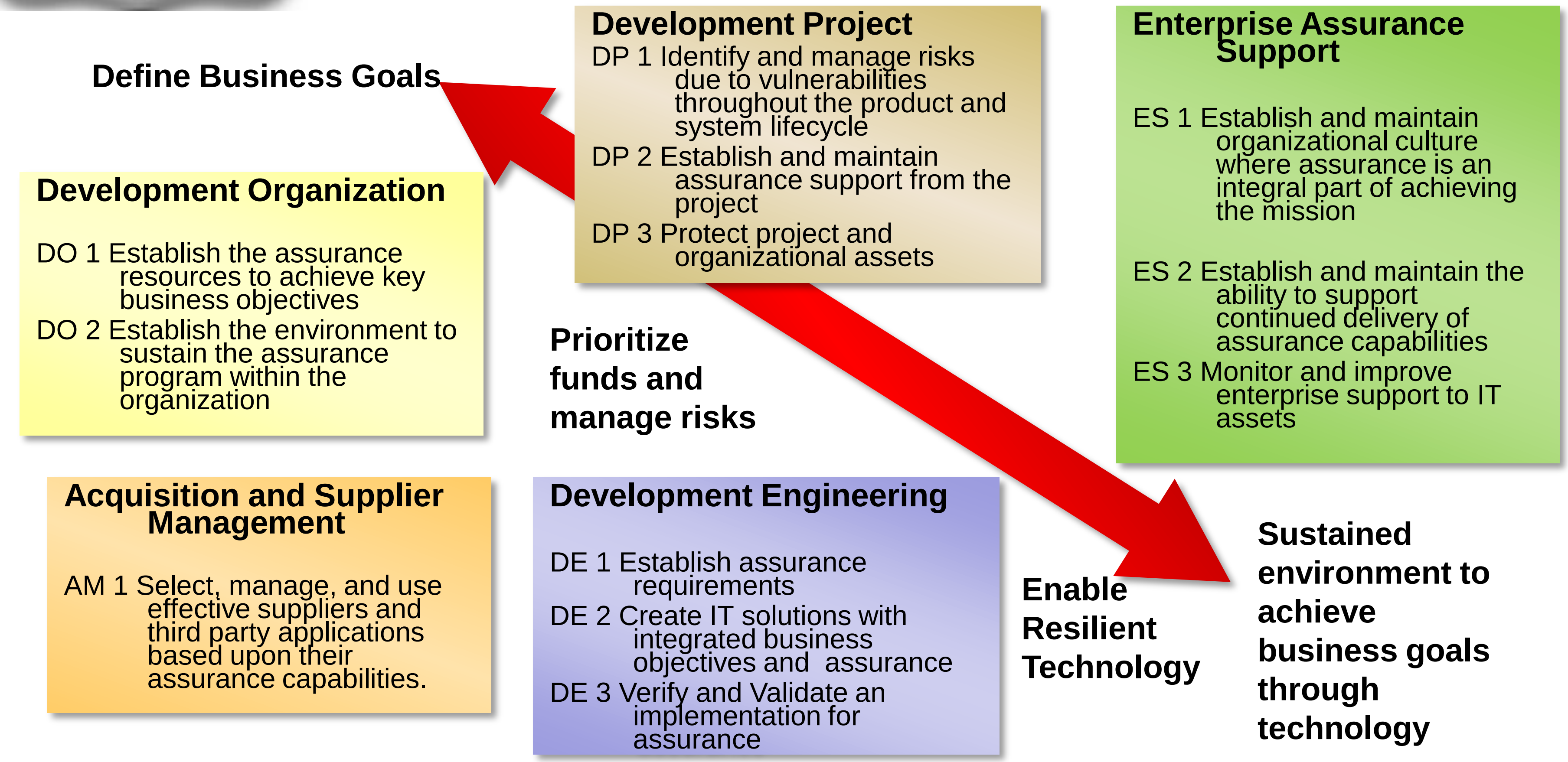
Organization Support

Adapted from: Paul Croll, Computer Sciences Corporation, August 2007

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

Assurance Process Reference Model



Define Business Goals

Development Project

DP 1 Identify and manage risks due to vulnerabilities throughout the product and system lifecycle

DP 2 Establish and maintain assurance support from the project

DP 3 Protect project and organizational assets

Enterprise Assurance Support

ES 1 Establish and maintain organizational culture where assurance is an integral part of achieving the mission

ES 2 Establish and maintain the ability to support continued delivery of assurance capabilities

ES 3 Monitor and improve enterprise support to IT assets

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

Prioritize funds and manage risks

Acquisition and Supplier Management

AM 1 Select, manage, and use effective suppliers and third party applications based upon their assurance capabilities.

Development Engineering

DE 1 Establish assurance requirements

DE 2 Create IT solutions with integrated business objectives and assurance

DE 3 Verify and Validate an implementation for assurance

Enable Resilient Technology

Sustained environment to achieve business goals through technology

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

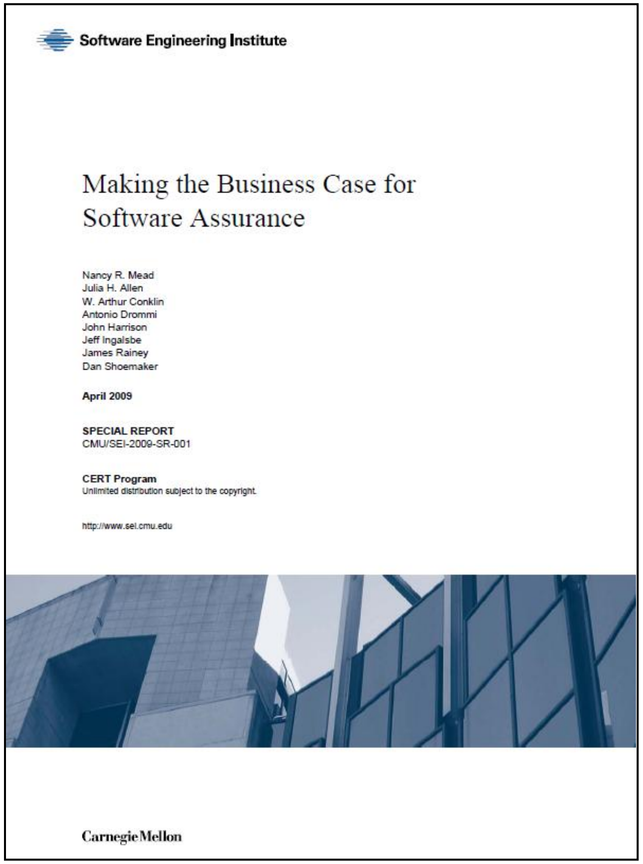
Courtesy of Michele Moss, BAH, SwA Processes & Practices

https://buildsecurityin.us-cert.gov/swa/proself_assm.html

Business Case for Software Assurance

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



Software Engineering Institute

Making the Business Case for
Software Assurance

Nancy R. Mead
Julia H. Allen
W. Arthur Conklin
Antonio Drommi
John Harrison
Jeff Ingalsbe
James Rainey
Dan Shoemaker

April 2009

SPECIAL REPORT
CMU/SEI-2009-SR-001

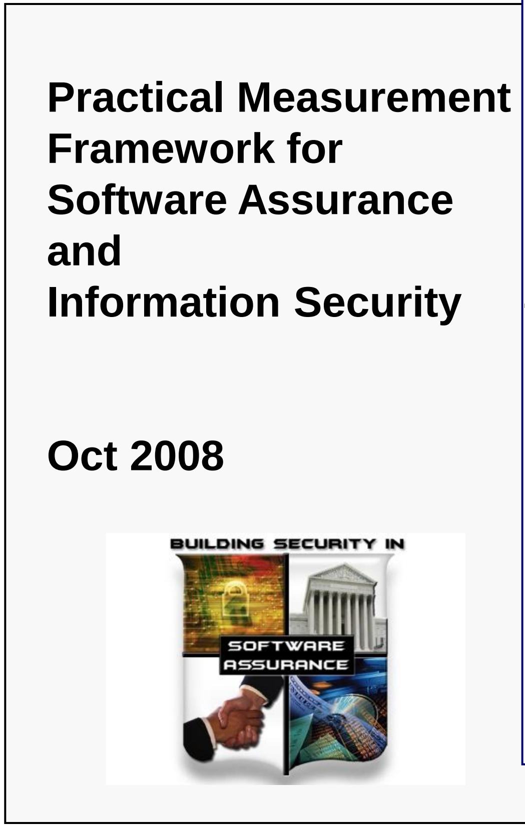
CERT Program
Unlimited distribution subject to the copyright.

http://www.sei.cmu.edu

CarnegieMellon

Security Measurement Resources

Oct 08 → Feb 09 → May 09 →

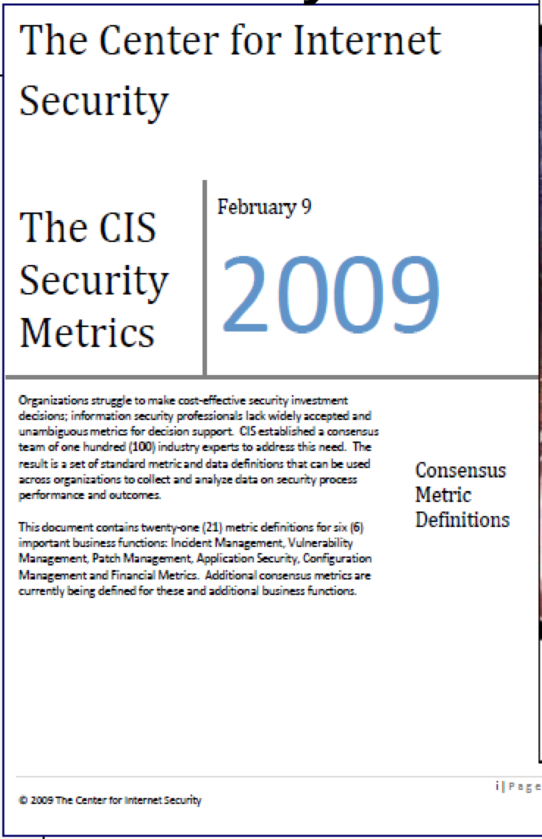


Practical Measurement Framework for Software Assurance and Information Security

Oct 2008

BUILDING SECURITY IN

SOFTWARE ASSURANCE



The Center for Internet Security

The CIS Security Metrics

February 9

2009

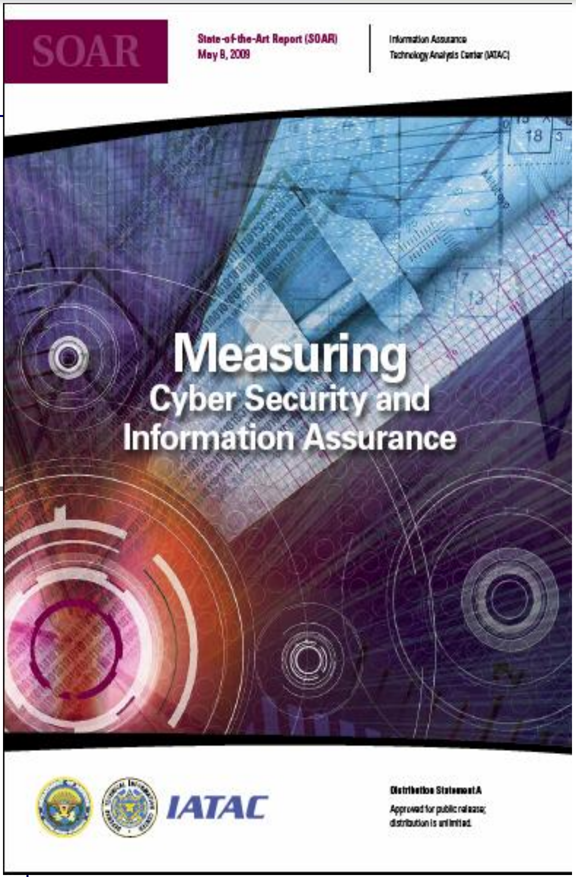
Organizations struggle to make cost-effective security investment decisions; information security professionals lack widely accepted and unambiguous metrics for decision support. CIS established a consensus team of one hundred (100) industry experts to address this need. The result is a set of standard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

This document contains twenty-one (21) metric 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.

Consensus Metric Definitions

© 2009 The Center for Internet Security

i | Page



SOAR

State-of-the-Art Report (SOAR)
May 8, 2009

Information Assurance
Technology Analysis Center (IATAC)

Measuring
Cyber Security and
Information Assurance

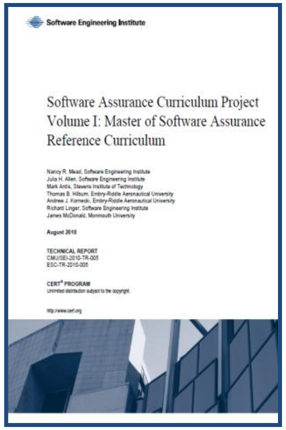
IATAC

Distribution Statement A
Approved for public release;
distribution is unlimited.

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.



Software Engineering Institute

Software Assurance Curriculum Project
Volume I: Master of Software Assurance
Reference Curriculum

Nancy R. Mead, Software Engineering Institute
Julia H. Allen, Software Engineering Institute
Mark Ardis, Stevens Institute of Technology
Thomas B. Hilburn, Embry-Riddle Aeronautical University
Andrew J. Kornecki, Embry-Riddle Aeronautical University
Richard Linger, Software Engineering Institute
James McDonald, Monmouth University

August 2010

TECHNICAL REPORT
CMU/SEI-2010-TR-005
ESC-TR-2010-005

CERT® PROGRAM
Unlimited distribution subject to the copyright.

http://www.cert.org

- **Vol II: SwA Undergraduate Course Outlines**

  see 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**

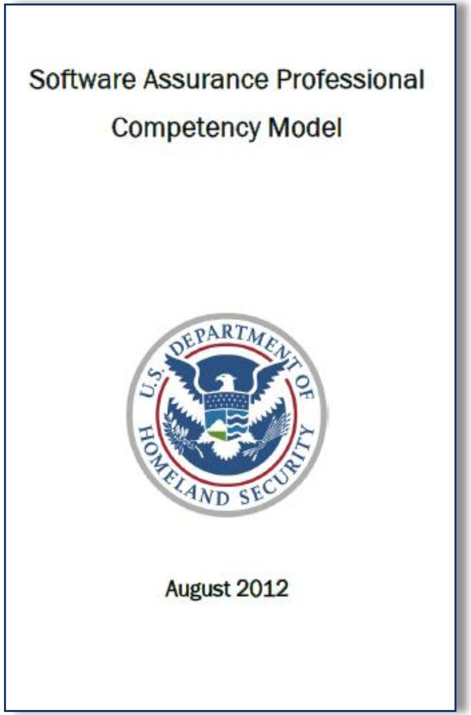
• Report on "Integrating the MSwA Reference Curriculum into Model Curriculum and Guidelines for Graduate Degree Programs in Information Systems" provides reference and guidance material.

•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.

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



Software Assurance Professional Competency Model

August 2012

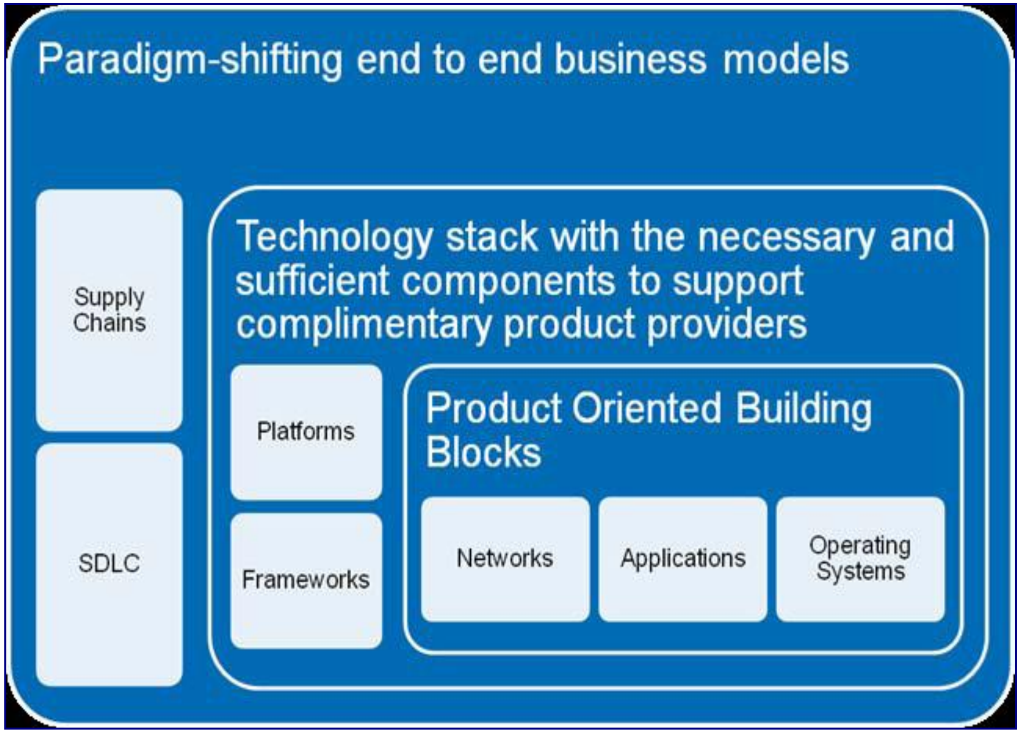
* Specialty Areas aligned with Framework for National Initiative for Cybersecurity Education

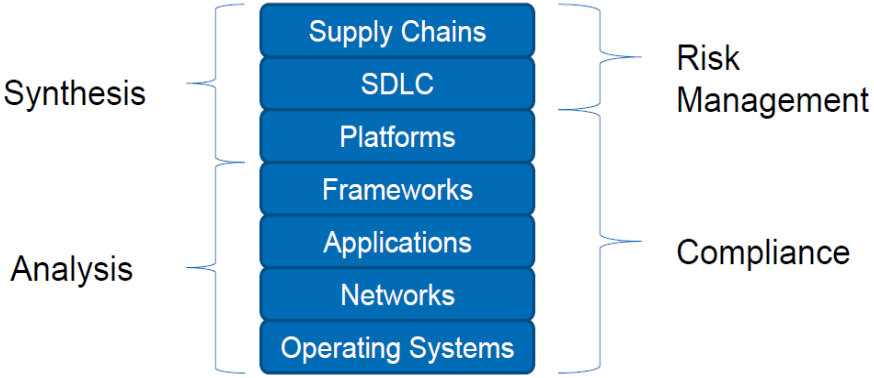
IT/software security risk landscape is a convergence between "defense in depth" and "defense in breadth"

Enterprise Risk Management and Governance are security motivators

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

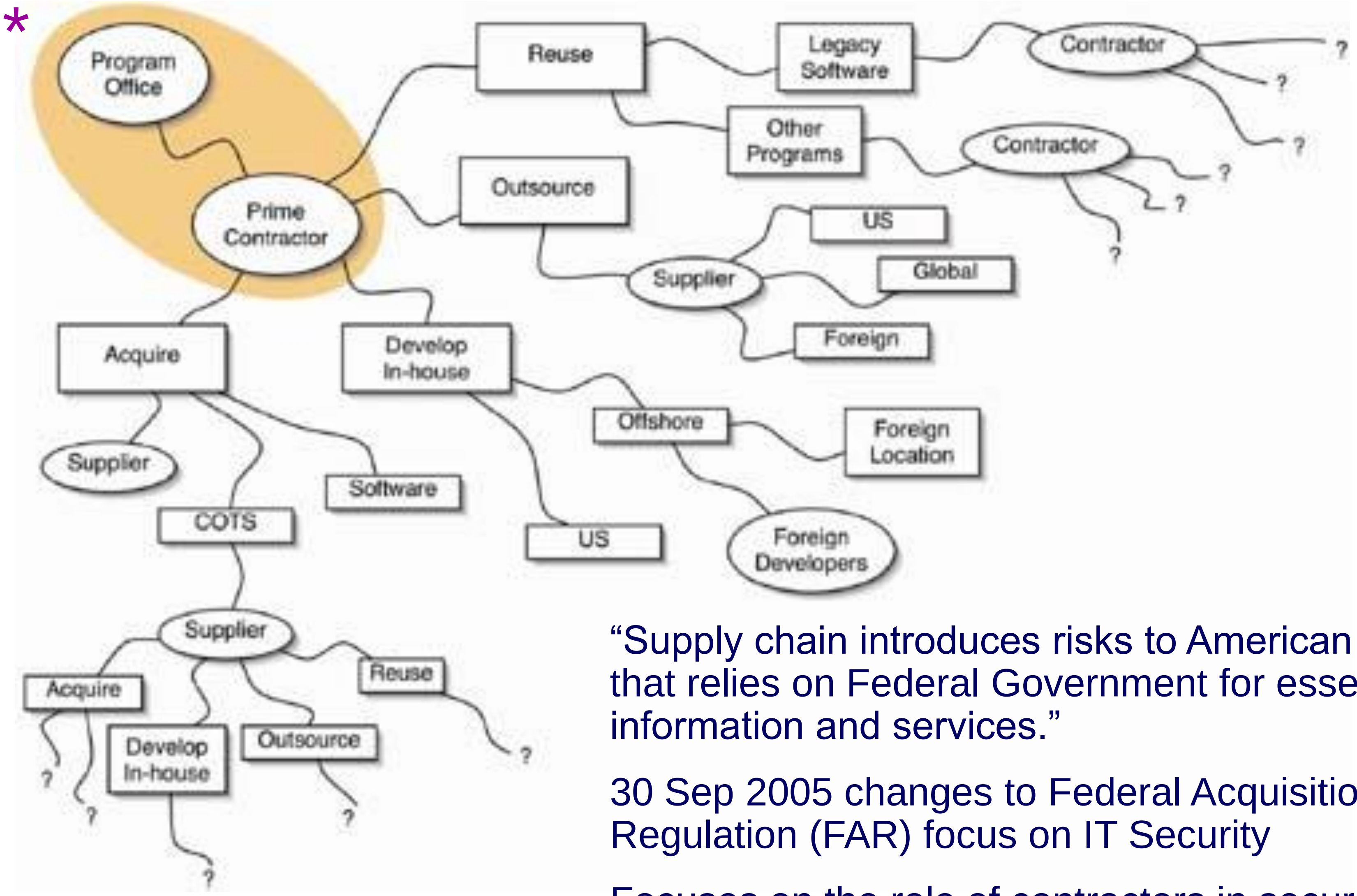
> "In the digital age, sovereignty is demarcated not by territorial frontiers but by supply chains."
>
> – Dan Geer, CISO In-Q-Tel


Paradigm-shifting end to end business models
Supply Chains
SDLC
Technology stack with the necessary and sufficient components to support complimentary product providers
Platforms
Frameworks
Product Oriented Building Blocks
Networks
Applications
Operating Systems



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.”

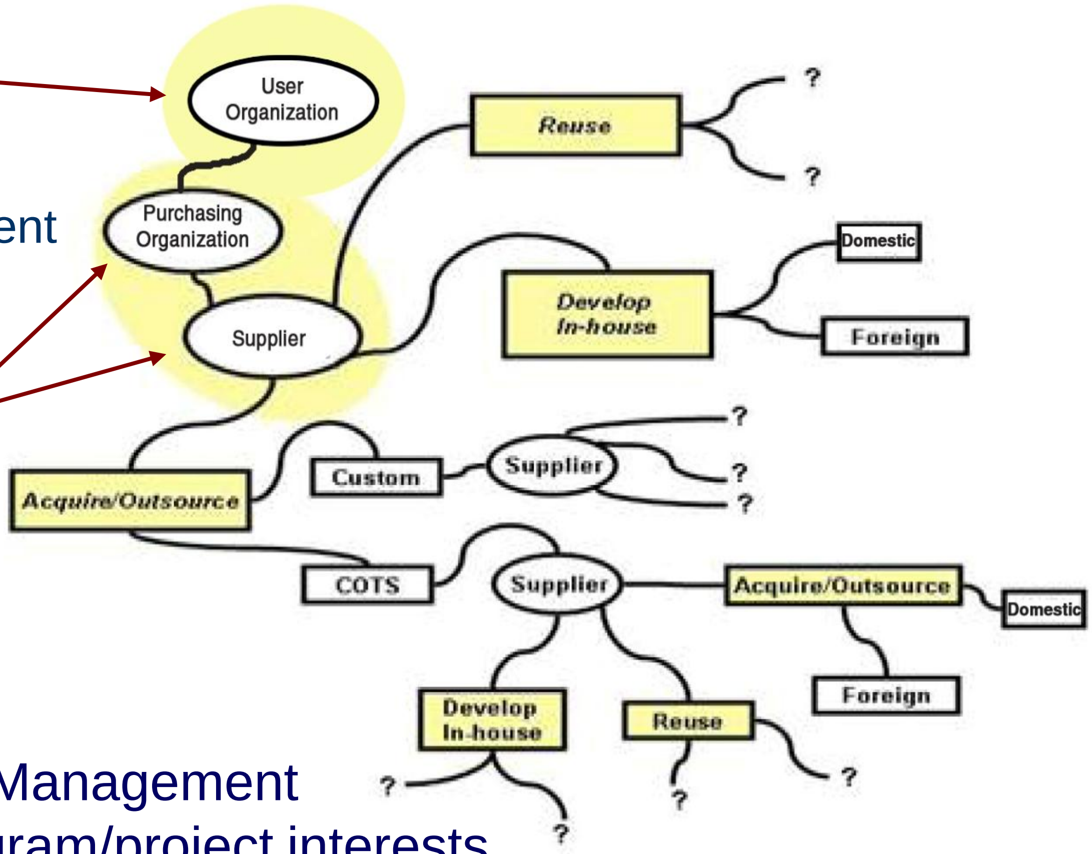
30 Sep 2005 changes to Federal Acquisition Regulation (FAR) focus on IT Security

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

“Scope of Supplier Expansion and Foreign Involvement” graphic in DACS www.softwaretechnews.com Secure Software Engineering, July 2005 article “Software Development Security: A Risk Management Perspective” synopsis of May 2004 GAO-04-678 report “Defense Acquisition: Knowledge of Software Suppliers Needed to Manage Risks”

Risk Management (Enterprise ↔ Project):
Shared Processes & Practices ↔ Different Focuses

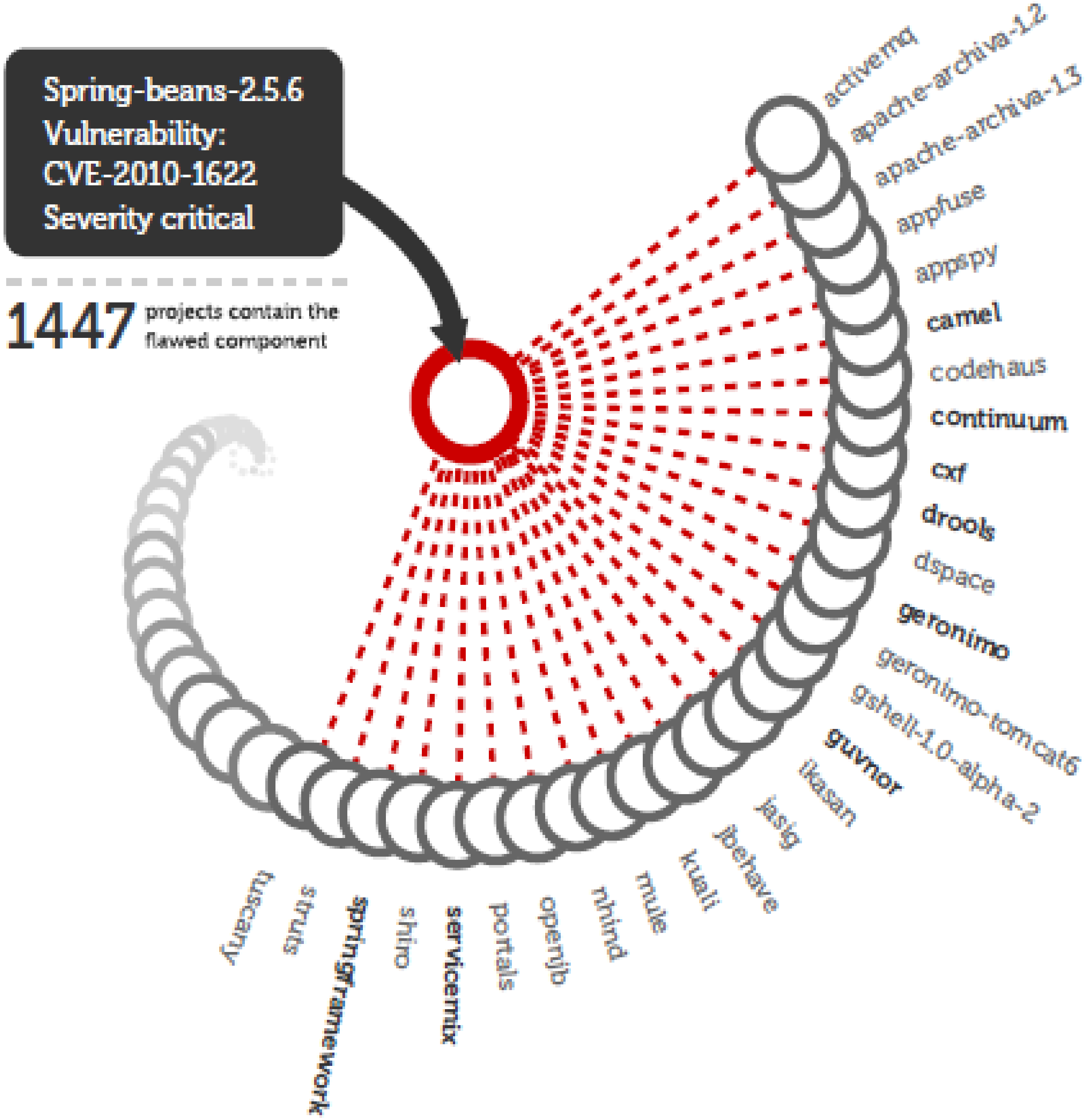
- Enterprise-Level:
  - Regulatory compliance
  - Changing threat environment
  - Business Case
- Program/Project-Level:
  - Cost
  - Schedule
  - Performance



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


Spring-beans-2.5.6
Vulnerability:
CVE-2010-1622
Severity critical

1447 projects contain the flawed component

Who makes risk decisions?

Who inherits the residual risk?

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

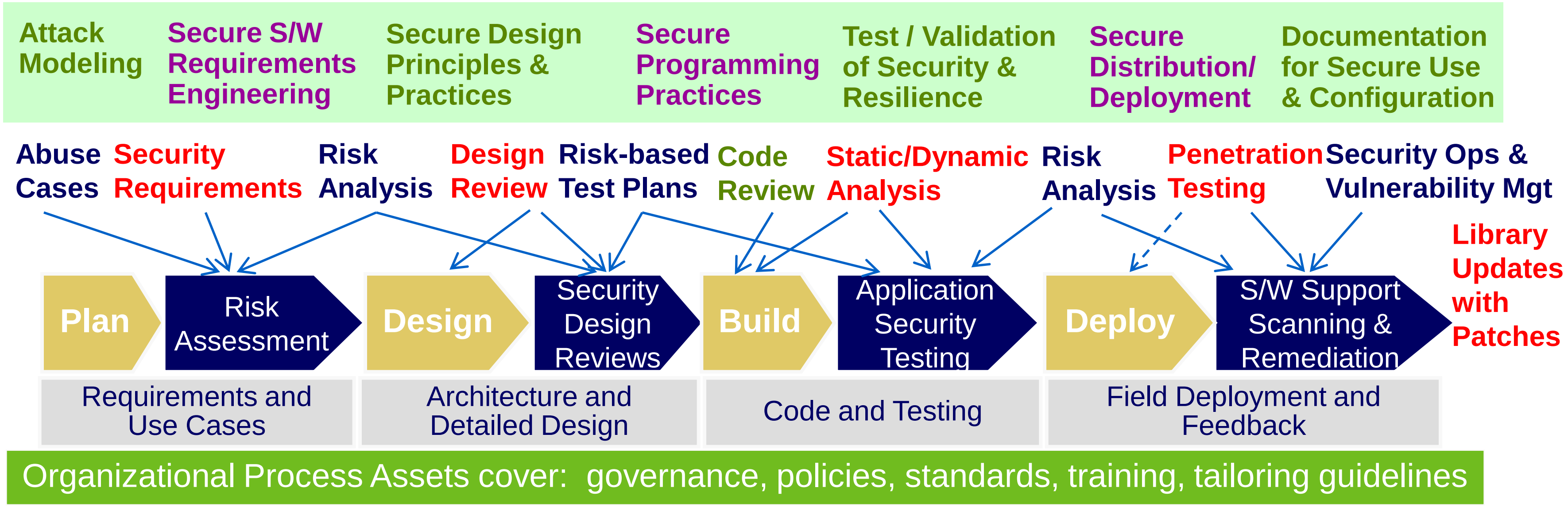
Source: Maximizing Benefits and Mitigating Risks of Open Source Components in Application Development, by Sonatype

Security-Enhanced Process Improvements

BUILDING SECURITY IN
SOFTWARE ASSURANCE

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

"Build Security In" throughout the lifecycle



Attack Modeling | Secure S/W Requirements Engineering | Secure Design Principles & Practices | Secure Programming Practices | Test / Validation of Security & Resilience | Secure Distribution/ Deployment | Documentation for Secure Use & Configuration

Abuse Cases | Security Requirements | Risk Analysis | Design Review | Risk-based Test Plans | Code Review | Static/Dynamic Analysis | Risk Analysis | Penetration Testing | Security Ops & Vulnerability Mgt

Plan → Risk Assessment → Design → Security Design Reviews → Build → Application Security Testing → Deploy → S/W Support Scanning & Remediation

Library Updates with Patches

Requirements and Use Cases | Architecture and Detailed Design | Code and Testing | 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.

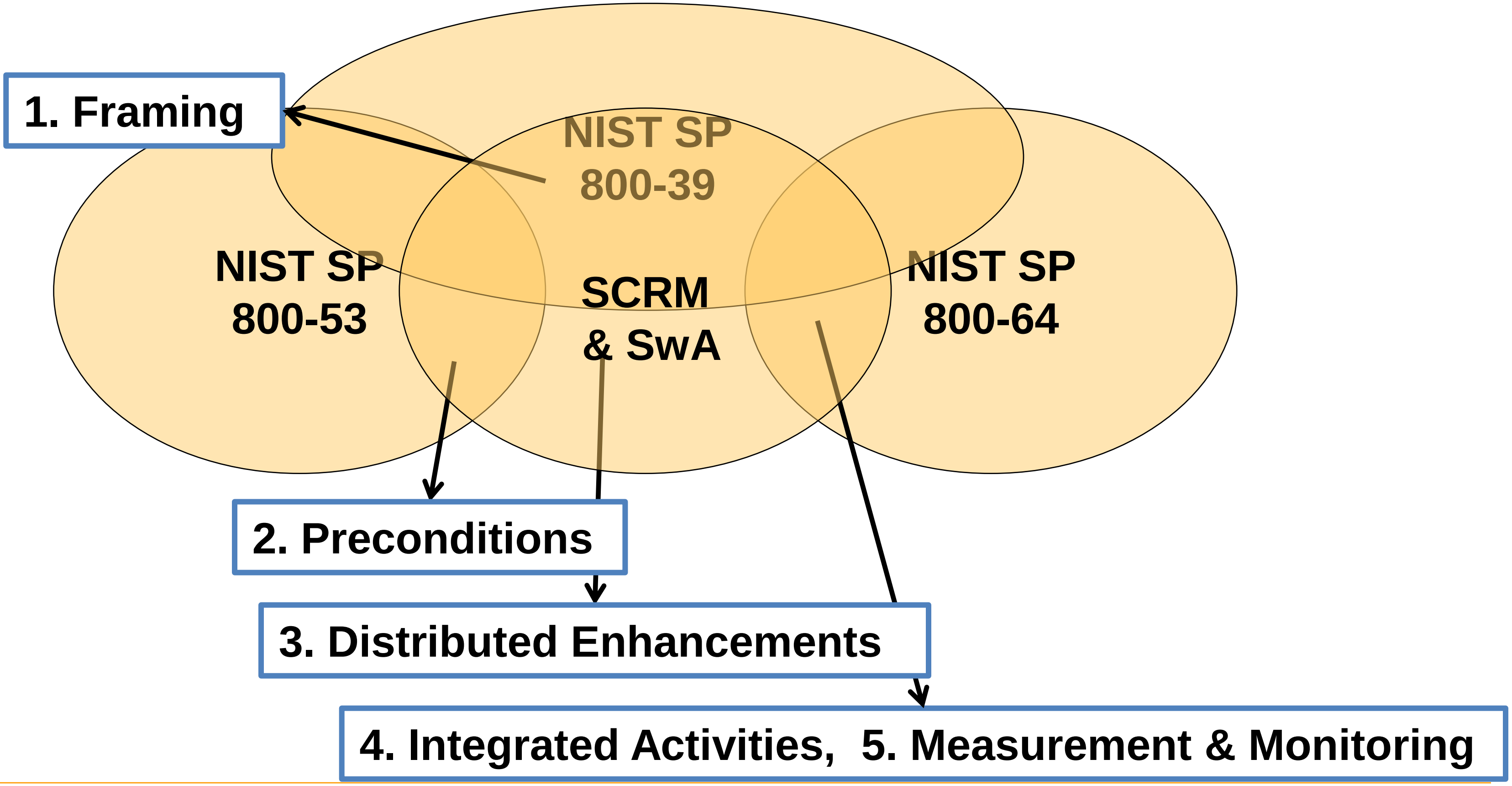
Homeland Security

* 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
NIST SP 800-39
NIST SP 800-53
SCRM & SwA
NIST SP 800-64
2. Preconditions
3. Distributed Enhancements
4. Integrated Activities, 5. Measurement & Monitoring

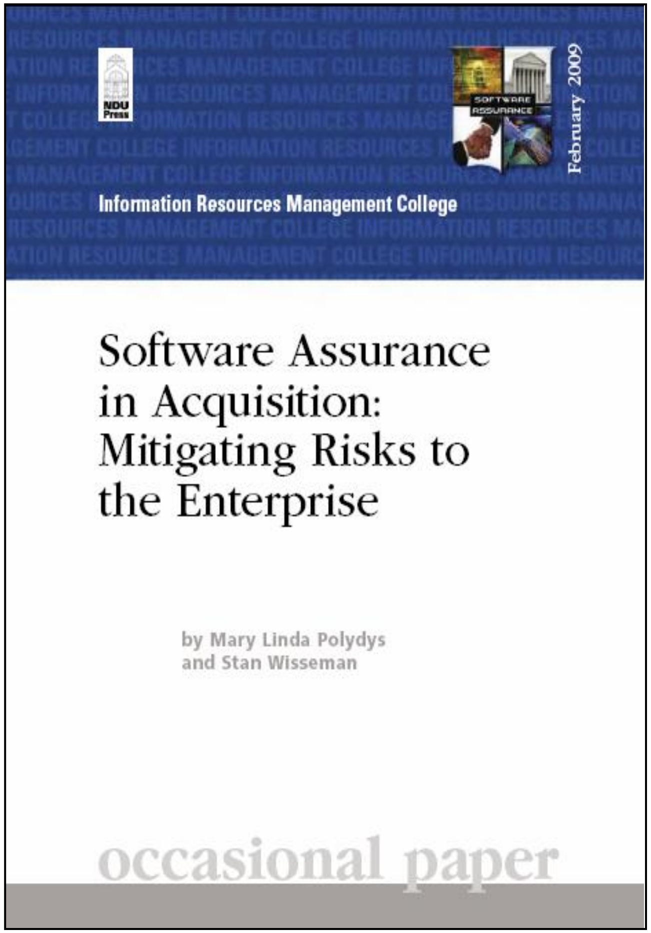
SwA Acquisition & Outsourcing Handbook

“Software Assurance in Acquisition:

Mitigating Risks to the Enterprise“

Version 1.0, Oct 2008, available for community use;

published by National Defense University Press, Feb 2009



NDU Press

SOFTWARE ASSURANCE

February 2009

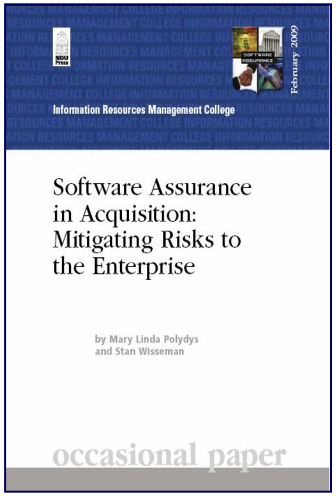
Information Resources Management College

Software Assurance in Acquisition: Mitigating Risks to the Enterprise

by Mary Linda Polydys
and Stan Wisseman

occasional paper

SwA Acquisition & Outsourcing Handbook



Information Resources Management College

NDU Press

February 2009

SOFTWARE ASSURANCE

Software Assurance in Acquisition: Mitigating Risks to the Enterprise

by Mary Linda Polydys
and Stan Wisseman

occasional paper

Software Supply Chain Risk Management and Due-Diligence -- Table 1 –SwA Concern Categories

SwA Concern Categories	Risks	Purpose for Questions
Software History and Licensing		
Development Process Management		
Software Security Training and Awareness		
Planning and Requirements		
Architecture and Design		
Software Development		
Built-in Software Defenses		
Component Assembly		
Testing		
Software Manufacture and Packaging		
Installation		
Assurance Claims and Evidence		
Support		
Software Change Management		
Timeliness of Vulnerability Mitigation		
Individual Malicious Behavior		
Security “Track Record”		
Financial History and Status		
Organizational History		
Foreign Interests and Influences		
Service Confidentiality Policies		
Operating Environment for Services		
Security Services and Monitoring		

Software Supply Chain Risk Management and Due-Diligence -- Table 1 –SwA Concern Categories

SwA Concern Categories	Risks	Purpose for Questions
Software History and Licensing	The software supplier's development practice in using code of unknown origin may be unable to produce trustworthy software.	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.
Development Process Management	If supplier project management does not perceive the value of SwA and enforce best practices, they will not be consistently implemented.	To determine whether project management enforces software assurance–related best practices.
Software Security Training and Awareness	Developers unaware of software assurance best practices are likely to implement software with security flaws (making it more susceptible to attack).	To determine whether training of developers in SwA best practices is a supplier policy and practice.
Planning and Requirements	If nonfunctional requirements (security, quality, safety) are not specified, developers will not implement them.	To determine whether the supplier's requirements analysis process explicitly addresses SwA requirements.
Architecture and Design	The software may be designed without considering security or minimization of exploitable defects.	To determine how security is considered during the design phase.
Software Development	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.	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.
Built-in Software Defenses	The software may lack preventive measures to help it resist attack effectively and proactively.	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.

Software Supply Chain Risk Management and Due-Diligence -- Table 1 –SwA Concern Categories

SwA Concern Categories	Risks	Purpose for Questions
Component Assembly	Insufficient analysis of software components used to assemble larger software packages may introduce vulnerabilities to the overall package.	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.
Testing	Software released with insufficient testing may contain an unacceptable number of exploitable defects.	To determine whether the appropriate set of analyses, reviews, and tests are performed on the software throughout the life cycle which evaluate security criteria.
Software Manufacture and Packaging	Vulnerabilities or malicious code could be introduced in the manufacturing or packaging process.	To determine how the software goes through the manufacturing process, how it is packaged, and how it remains secure.
Installation	The software may not install as advertised and the acquirer may not get the software to function as expected.	To ensure the supplier provides an acceptable level of support during the installation process.
Assurance Claims and Evidence	Supplier assurance claims (with supporting evidence) may be non-existent or insufficiently verified.	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.
Support	Supplier ceases to supply patches and new releases prior to the acquirer ending use of software. Vulnerabilities may go unmitigated.	To ensure understanding of supplier policy for security fixes and when products are no longer supported.
Software Change Management	Weak change control procedures can corrupt software and introduce new security vulnerabilities.	To determine whether software changes are adequately assessed and verified by supplier management.
Timeliness of Vulnerability Mitigation	Sometimes it can be extremely difficult to make a software supplier take notice and repair software to mitigate reported vulnerabilities.	To ensure security defects and configuration errors are fixed properly and in a timely fashion.

Software Supply Chain Risk Management and Due-Diligence -- Table 1 –SwA Concern Categories

SwA Concern Categories	Risks	Purpose for Questions
Individual Malicious Behavior	A developer purposely inserts malicious code, and supplier lacks procedures to mitigate risks from insider threats within the supply chain.	To determine whether the supplier has and enforces policies to minimize individual malicious behavior.
Security “Track Record”	A software supplier that is unresponsive to known software vulnerabilities may not mitigate/patch vulnerabilities in a timely manner.	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.
Financial History and Status	A software supplier that goes out of business will be unable to provide support or mitigate product defects and vulnerabilities.	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.
Organizational History	There may be conflicting circumstances or competing interests within the organization that may lead to increased risk in the software development.	To understand the supplier’s organizational background, roles, and relationships that might have an impact on supporting the software.
Foreign Interests and Influences	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.	To help identify supplier companies that may have individuals with competing interests or malicious intent to a domestic buyer/user.
Service Confidentiality Policies	Without policies to enforce client data confidentiality/ privacy, acquirer’s data could be at risk without service supplier liability.	To determine the service provider’s confidentiality and privacy policies and ensure their enforcement.
Operating Environment for Services	Operating environment for the services may not be hardened or otherwise secure.	To understand the controls the supplier has established to operate the software securely.
Security Services and Monitoring	Insufficient security monitoring may allow attacks to impact services.	To ensure software and its operating environment are regularly reviewed for adherence to SwA requirements through periodic testing and evaluation.

Table 2- Questions for COTS (Proprietary & Open Source), GOTS, & Custom Software

No	Question	COTS Propri-etary	COTS Open-Source	GOTS	Custom
→ 1	Can the pedigree of the software be established? Briefly explain what is known of the people and processes that created the software.	✓	✓	✓	✓
→ 2	Explain the change management procedure that identifies the type and extent of changes conducted on the software throughout its life cycle.	✓		✓	✓
→ 3	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.	✓	✓		✓
4	Is there a clear chain of licensing from original author to latest modifier? Describe the chain of licensing.	✓			
5	What assurances are provided that the licensed software does not infringe upon any copyright or patent? Explain.	✓		✓	✓
6	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.	✓			✓
→ 7	Are licensed software components still valid for the intended use?	✓		✓	
→ 8	Is the software in question original source or a modified version?		✓		
9	Has the software been reviewed to confirm that it does not infringe upon any copyright or patent?	✓	✓		✓
→ 10	How long has the software source been available? Is there an active user community providing peer review and actively evolving the software?	✓	✓		

Table 2- Questions for COTS (Proprietary & Open Source), GOTS, and Custom Software

No.	Question	COTS Propri-etary	COTS Open-Source	GOTS	Custom
→ 11	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)?	✓			✓
→ 12	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?	✓			✓
13	Does software have a positive reputation? Does software have a positive reputation relative to security? Are there reviews that recommend it?	✓	✓		
→ 14	Is the level of security where the software was developed the same as where the software will operate?			✓	✓
	Development Process Management				
15	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)?	✓		✓	✓
→ 16	What security measurement practices and data does the company use to assist product planning?	✓			✓
→ 17	Is software assurance considered in all phases of development? Explain.	✓		✓	✓
→ 18	How is software risk managed? Are anticipated threats identified, assessed, and prioritized?	✓		✓	✓

Table 1 –SwA Concern Categories -- (with interests relevant to security and privacy)

SwA Concern Categories	Risks	Purpose for Questions
Service Confidentiality Policies	Without policies to enforce client data confidentiality/ privacy, acquirer's data could be at risk without service supplier liability.	To determine the service provider's confidentiality and privacy policies and ensure their enforcement.

Table 3 - Questions for Hosted Applications

No.	Questions
	Service Confidentiality Policies
1	What are the customer confidentiality policies? How are they enforced?
2	What are the customer privacy policies? How are they enforced?
3	What are the policies and procedures used to protect sensitive information from unauthorized access? How are the policies enforced?
4	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?
	Operating Environment for Services
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 of vulnerabilities, either intentionally or unintentionally designed or inserted as part of the software throughout the life cycle.*

the level of confidence that software functions as intended and is free from vulnerabilities

Derived From: CNSSI-4009

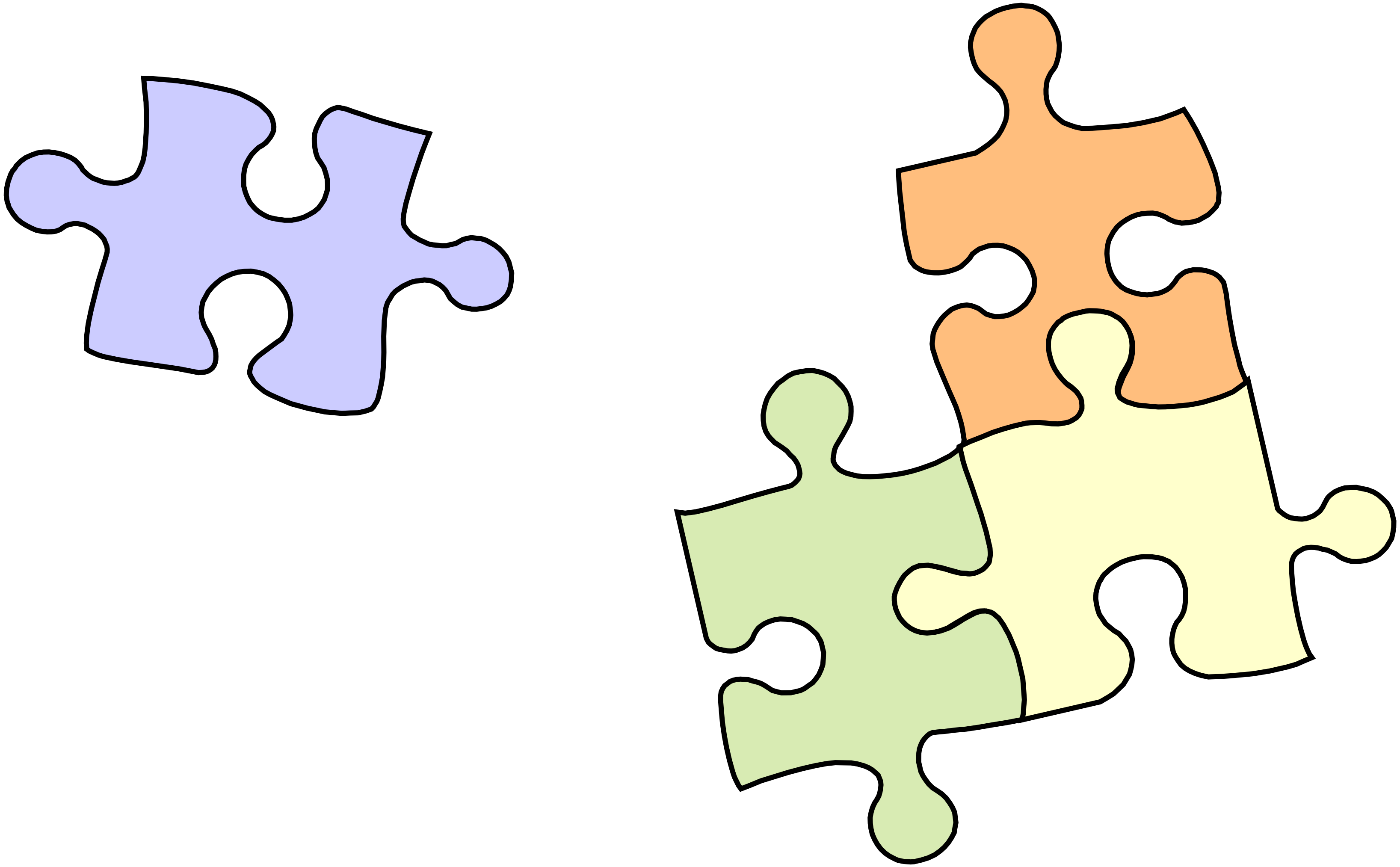
Automation

Languages, enumerations, registries, tools, and repositories

throughout the Lifecycle

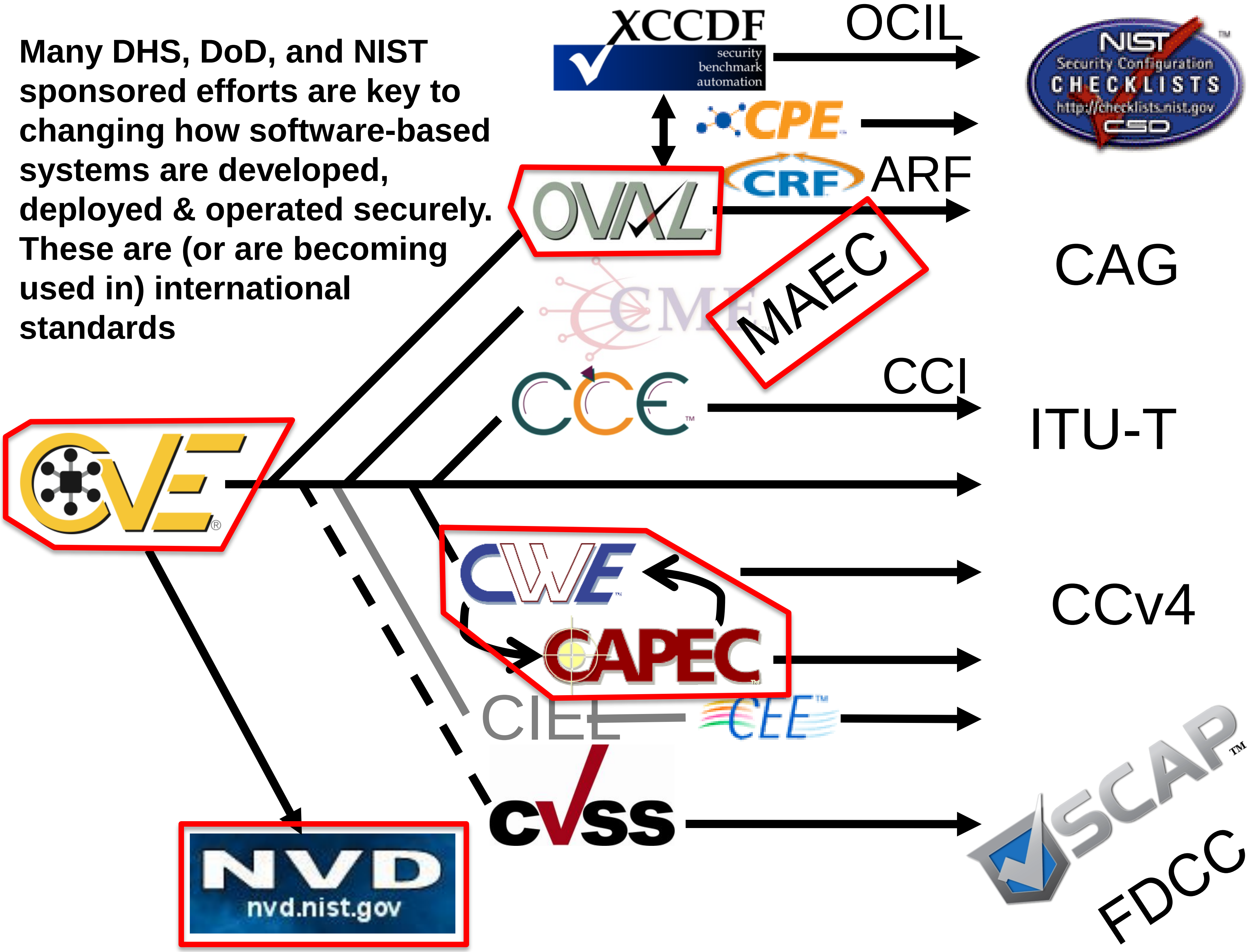
Including design, coding, testing, deployment, configuration and operation

Automation is *one piece*

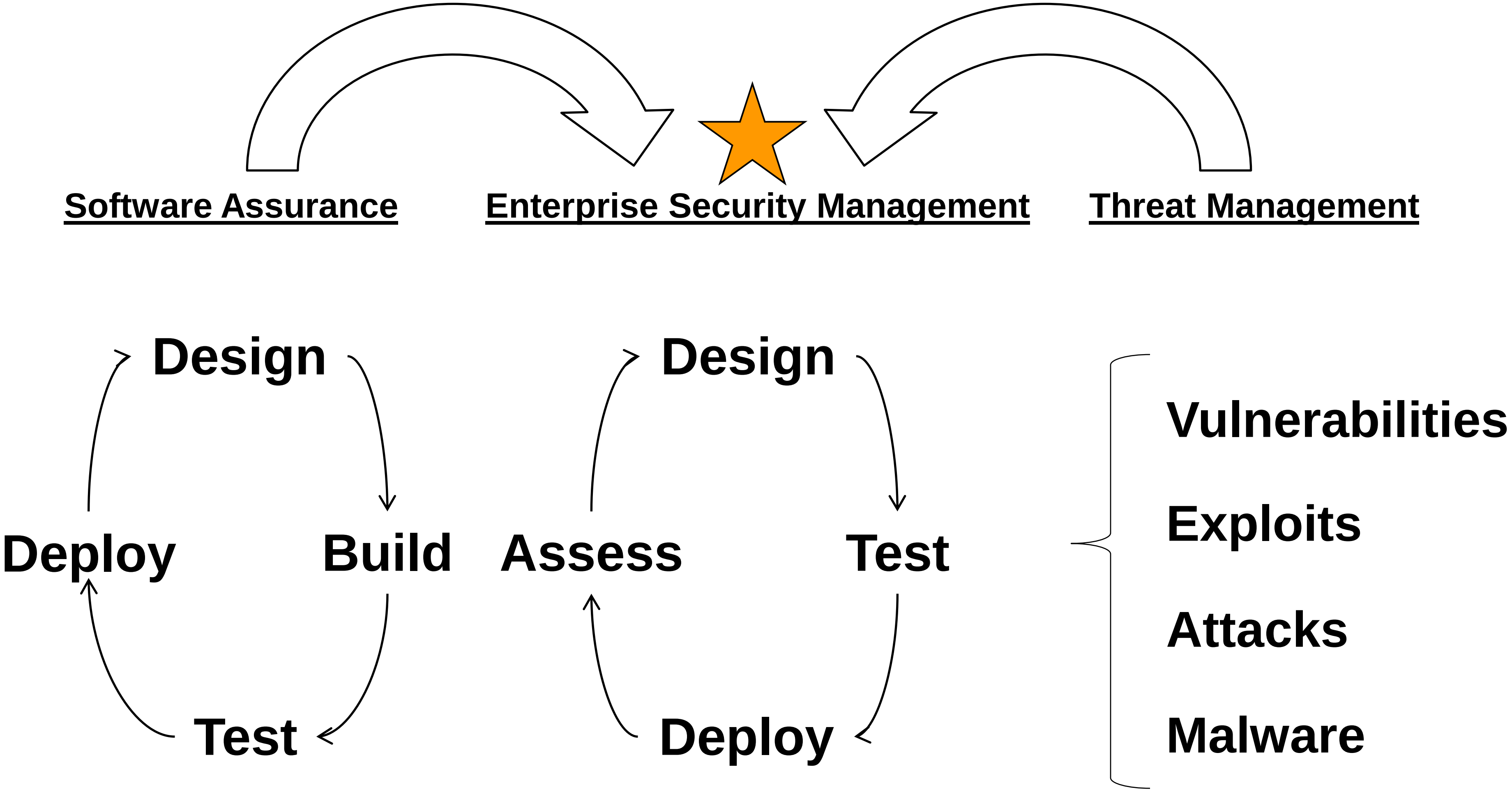


of the SwA puzzle.

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


XCCDF
security benchmark automation
OCIL
NIST Security Configuration CHECKLISTS http://checklists.nist.gov CSD
CPE
CRF
ARF
OVAL
MAEC
CME
CAG
CCE
CCI
ITU-T
CVE
CWE
CAPEC
CCv4
CIEL
CEE
CVSS
SCAP
FDCC
NVD
nvd.nist.gov

Making Security Measurable (MSM): You Are Here



Software Assurance

Enterprise Security Management

Threat Management

Design

Build

Test

Deploy

Design

Test

Deploy

Assess

Vulnerabilities

Exploits

Attacks

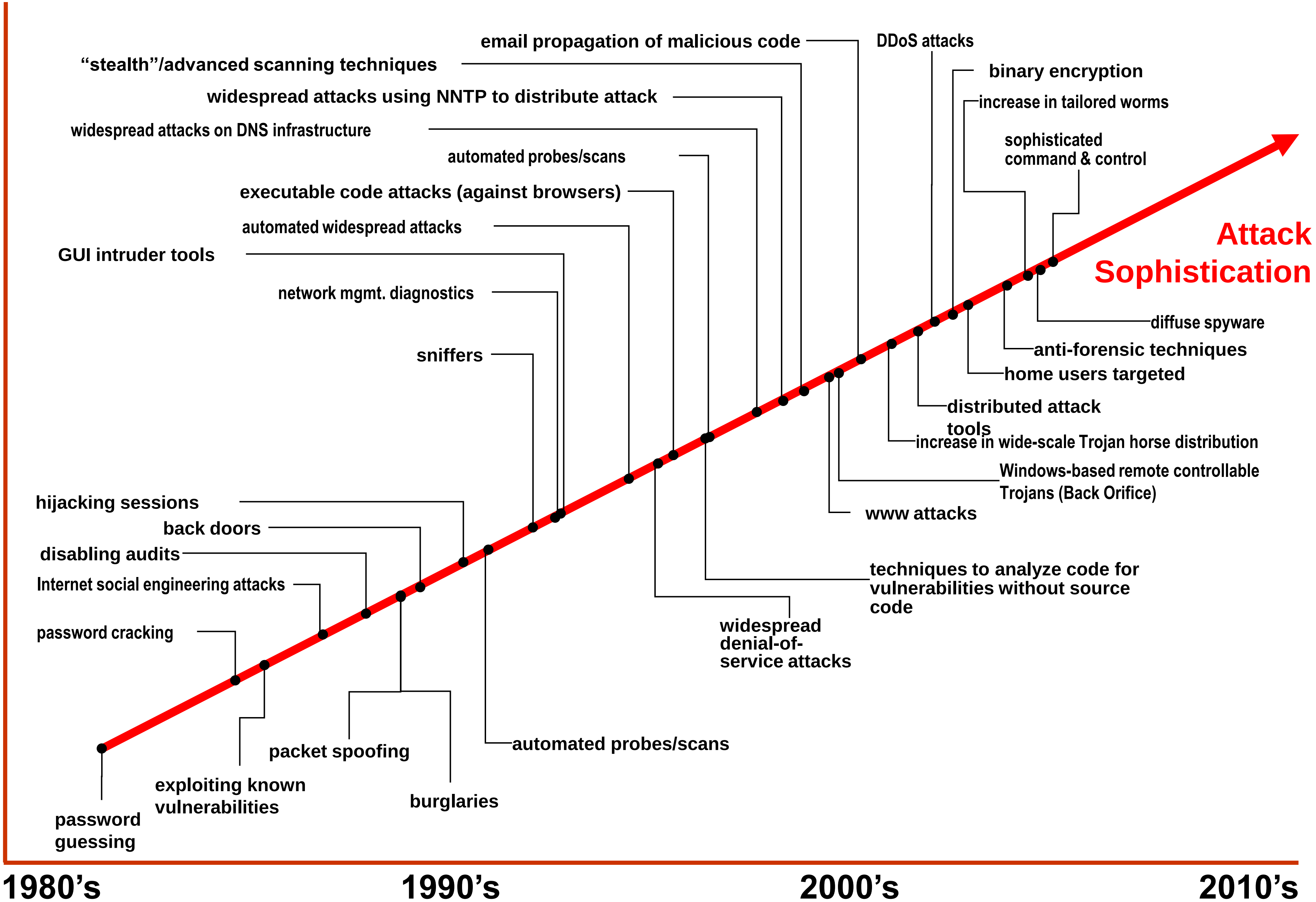
Malware

CWE, CAPEC, CWSS, CWRAF

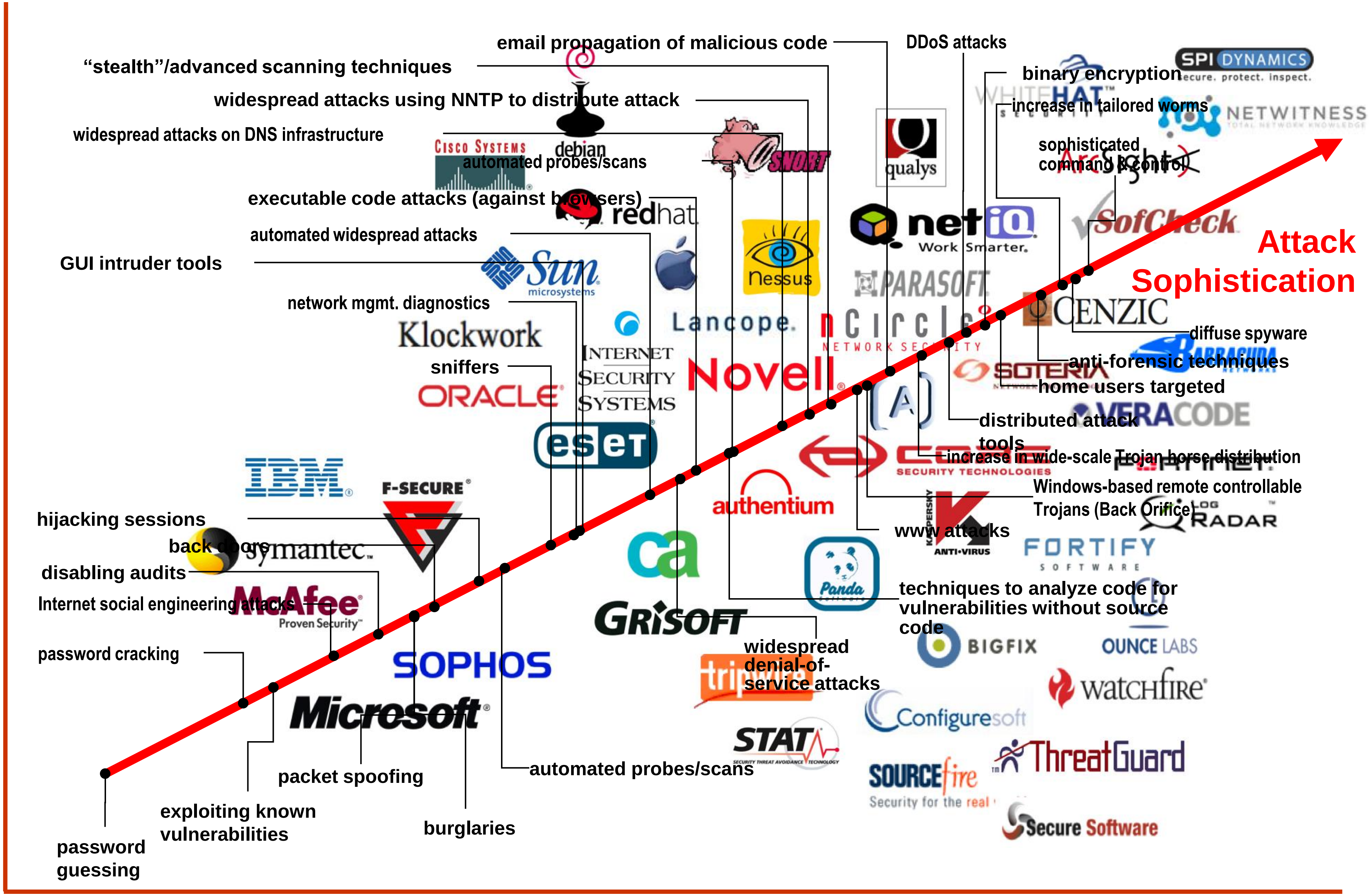
CPE, CCE, OVAL, OCIL,
XCCDF, AssetId, ARF

CVE, CWE, CAPEC, MAEC,
CybOX, IODEF

Cyber Threats Emerged Over Time


email propagation of malicious code
DDoS attacks
"stealth"/advanced scanning techniques
binary encryption
widespread attacks using NNTP to distribute attack
increase in tailored worms
widespread attacks on DNS infrastructure
sophisticated command & control
automated probes/scans
executable code attacks (against browsers)
automated widespread attacks
Attack Sophistication
GUI intruder tools
network mgmt. diagnostics
diffuse spyware
anti-forensic techniques
sniffers
home users targeted
distributed attack tools
increase in wide-scale Trojan horse distribution
Windows-based remote controllable Trojans (Back Orifice)
hijacking sessions
www attacks
back doors
disabling audits
techniques to analyze code for vulnerabilities without source code
Internet social engineering attacks
widespread denial-of-service attacks
password cracking
packet spoofing
automated probes/scans
exploiting known vulnerabilities
burglaries
password guessing
1980's
1990's
2000's
2010's

Solutions Also Emerged Over Time



password guessing

exploiting known vulnerabilities

password cracking

packet spoofing

Internet social engineering attacks

disabling audits

back doors

hijacking sessions

burglaries

automated probes/scans

sniffers

network mgmt. diagnostics

GUI intruder tools

automated widespread attacks

executable code attacks (against browsers)

automated probes/scans

widespread attacks on DNS infrastructure

widespread attacks using NNTP to distribute attack

“stealth”/advanced scanning techniques

email propagation of malicious code

techniques to analyze code for vulnerabilities without source code

widespread denial-of-service attacks

www attacks

Windows-based remote controllable Trojans (Back Orifice)

increase in wide-scale Trojan horse distribution

DDoS attacks

distributed attack tools

home users targeted

anti-forensic techniques

diffuse spyware

sophisticated command & control

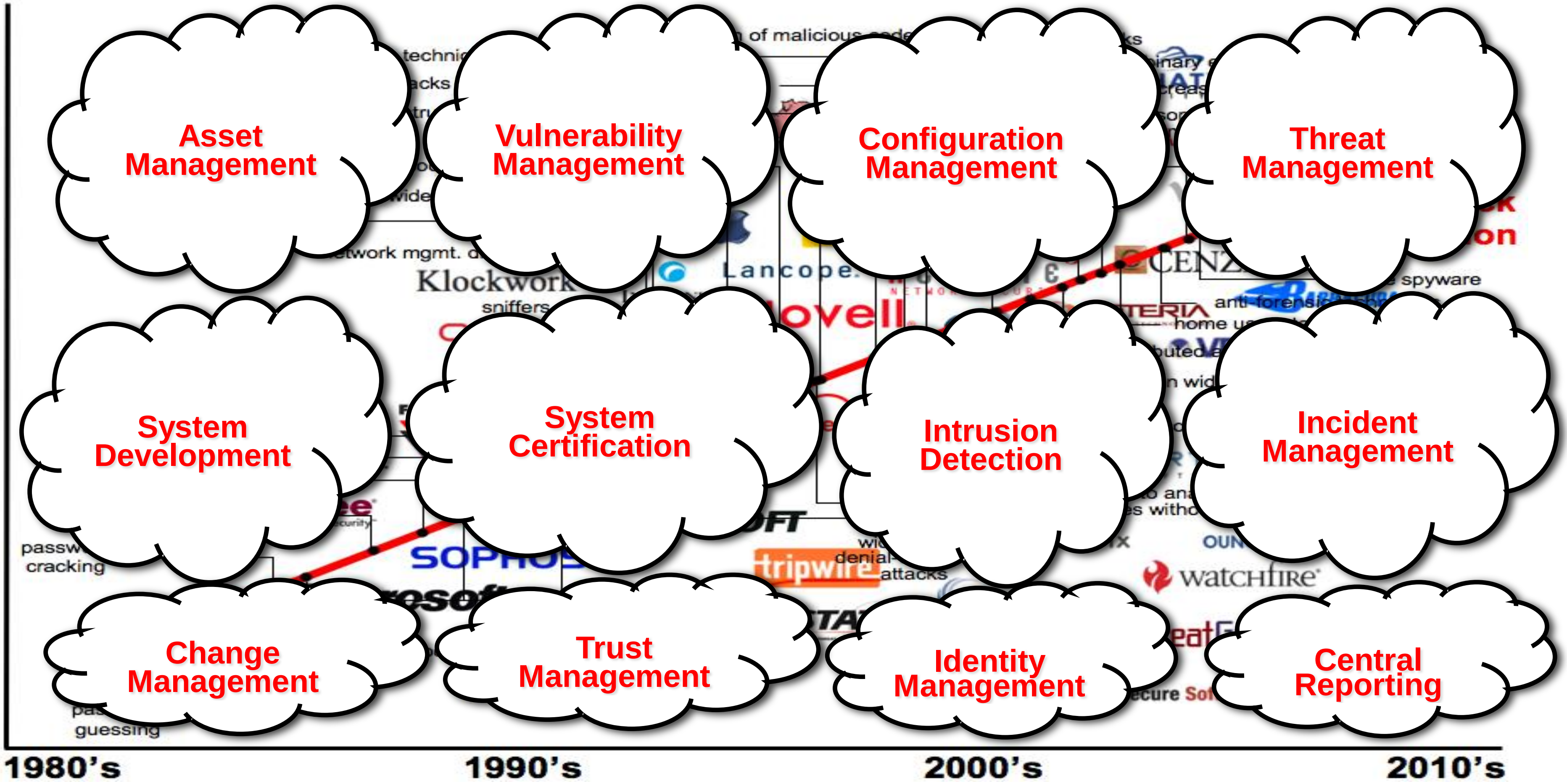
binary encryption

increase in tailored worms

Attack Sophistication

1980’s 1990’s 2000’s 2010’s

Architecting Security with Information Standards for COIs



Asset Management

Vulnerability Management

Configuration Management

Threat Management

System Development

System Certification

Intrusion Detection

Incident Management

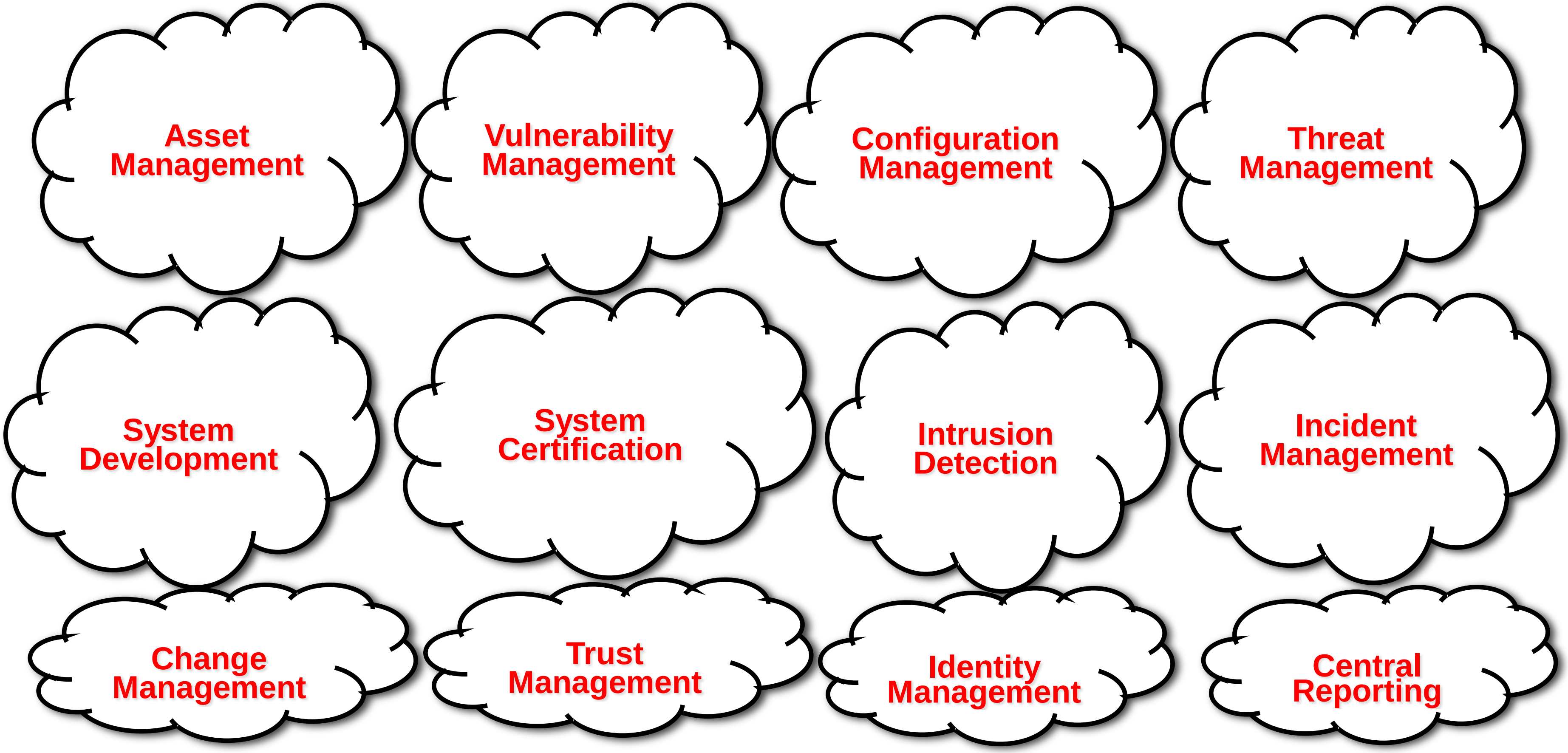
Change Management

Trust Management

Identity Management

Central Reporting

Making Security Measurable™



Asset Management

Vulnerability Management

Configuration Management

Threat Management

System Development

System Certification

Intrusion Detection

Incident Management

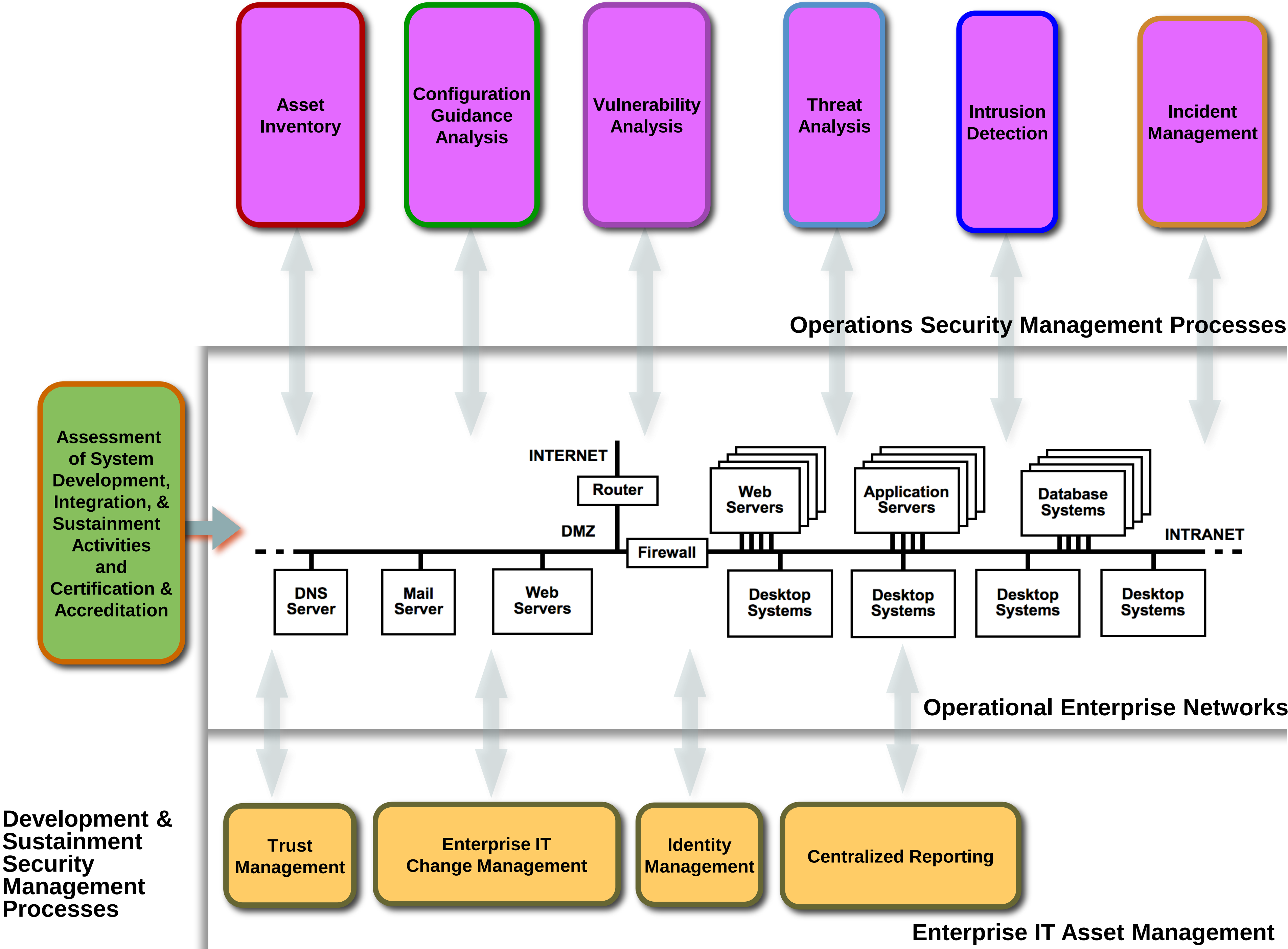
Change Management

Trust Management

Identity Management

Central Reporting

Making Security Measurable™



Asset Inventory

Configuration Guidance Analysis

Vulnerability Analysis

Threat Analysis

Intrusion Detection

Incident Management

Operations Security Management Processes

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

INTERNET

Router

DMZ

Firewall

Web Servers

Application Servers

Database Systems

INTRANET

DNS Server

Mail Server

Web Servers

Desktop Systems

Desktop Systems

Desktop Systems

Desktop Systems

Operational Enterprise Networks

Development & Sustainment Security Management Processes

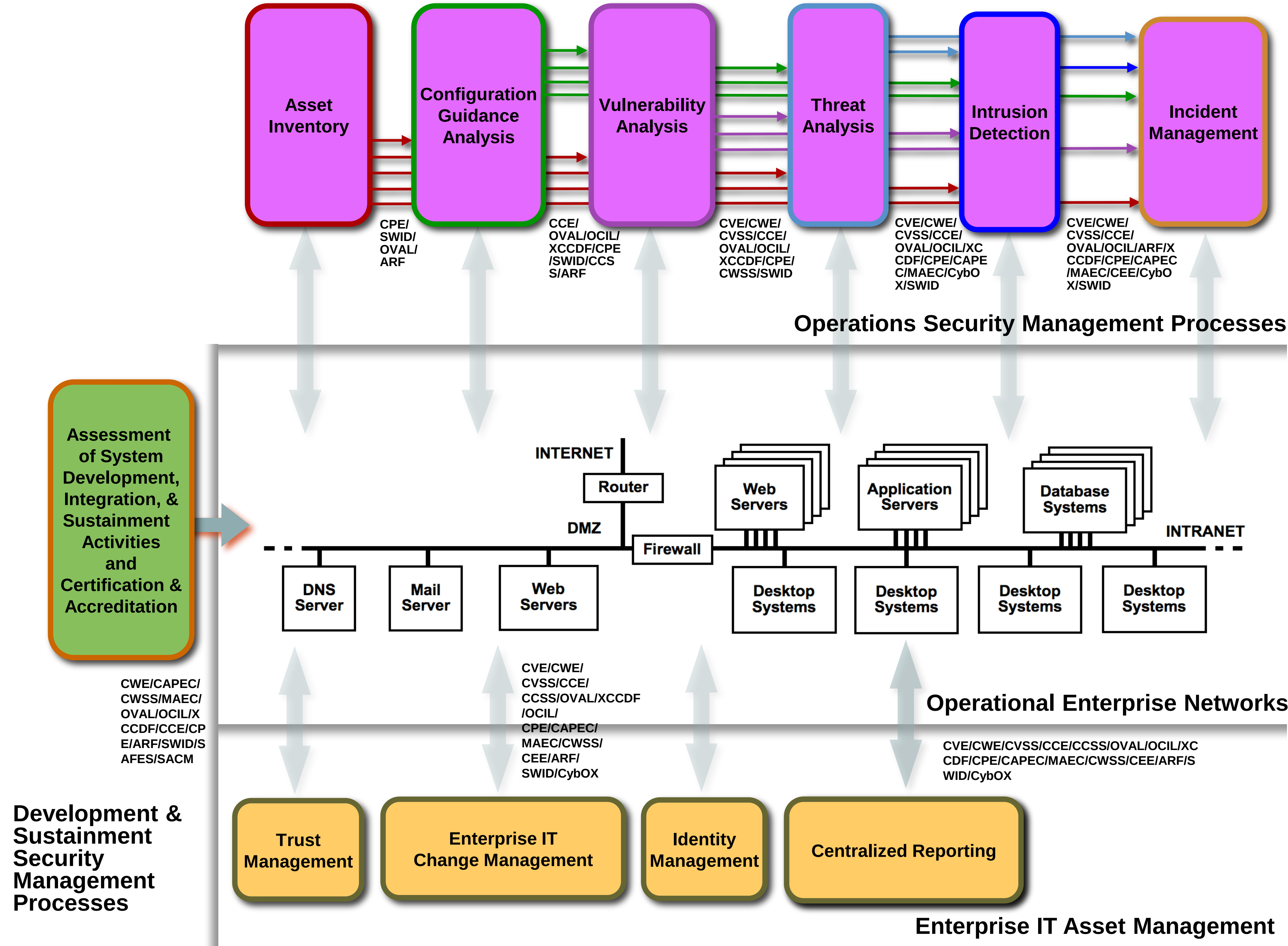
Trust Management

Enterprise IT Change Management

Identity Management

Centralized Reporting

Enterprise IT Asset Management



Asset Inventory

Configuration Guidance Analysis

Vulnerability Analysis

Threat Analysis

Intrusion Detection

Incident Management

CPE/
SWID/
OVAL/
ARF

CCE/
OVAL/OCIL/
XCCDF/CPE
/SWID/CCS
S/ARF

CVE/CWE/
CVSS/CCE/
OVAL/OCIL/
XCCDF/CPE/
CWSS/SWID

CVE/CWE/
CVSS/CCE/
OVAL/OCIL/XC
CDF/CPE/CAPE
C/MAEC/CybO
X/SWID

CVE/CWE/
CVSS/CCE/
OVAL/OCIL/ARF/X
CCDF/CPE/CAPEC
/MAEC/CEE/CybO
X/SWID

Operations Security Management Processes

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

CWE/CAPEC/
CWSS/MAEC/
OVAL/OCIL/X
CCDF/CCE/CP
E/ARF/SWID/S
AFES/SACM

CVE/CWE/
CVSS/CCE/
CCSS/OVAL/XCCDF
/OCIL/
CPE/CAPEC/
MAEC/CWSS/
CEE/ARF/
SWID/CybOX

Operational Enterprise Networks

CVE/CWE/CVSS/CCE/CCSS/OVAL/OCIL/XC
CDF/CPE/CAPEC/MAEC/CWSS/CEE/ARF/S
WID/CybOX

Development & Sustainment Security Management Processes

Trust Management

Enterprise IT Change Management

Identity Management

Centralized Reporting

Enterprise IT Asset Management

Cyber Ecosystem Standardization Efforts

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

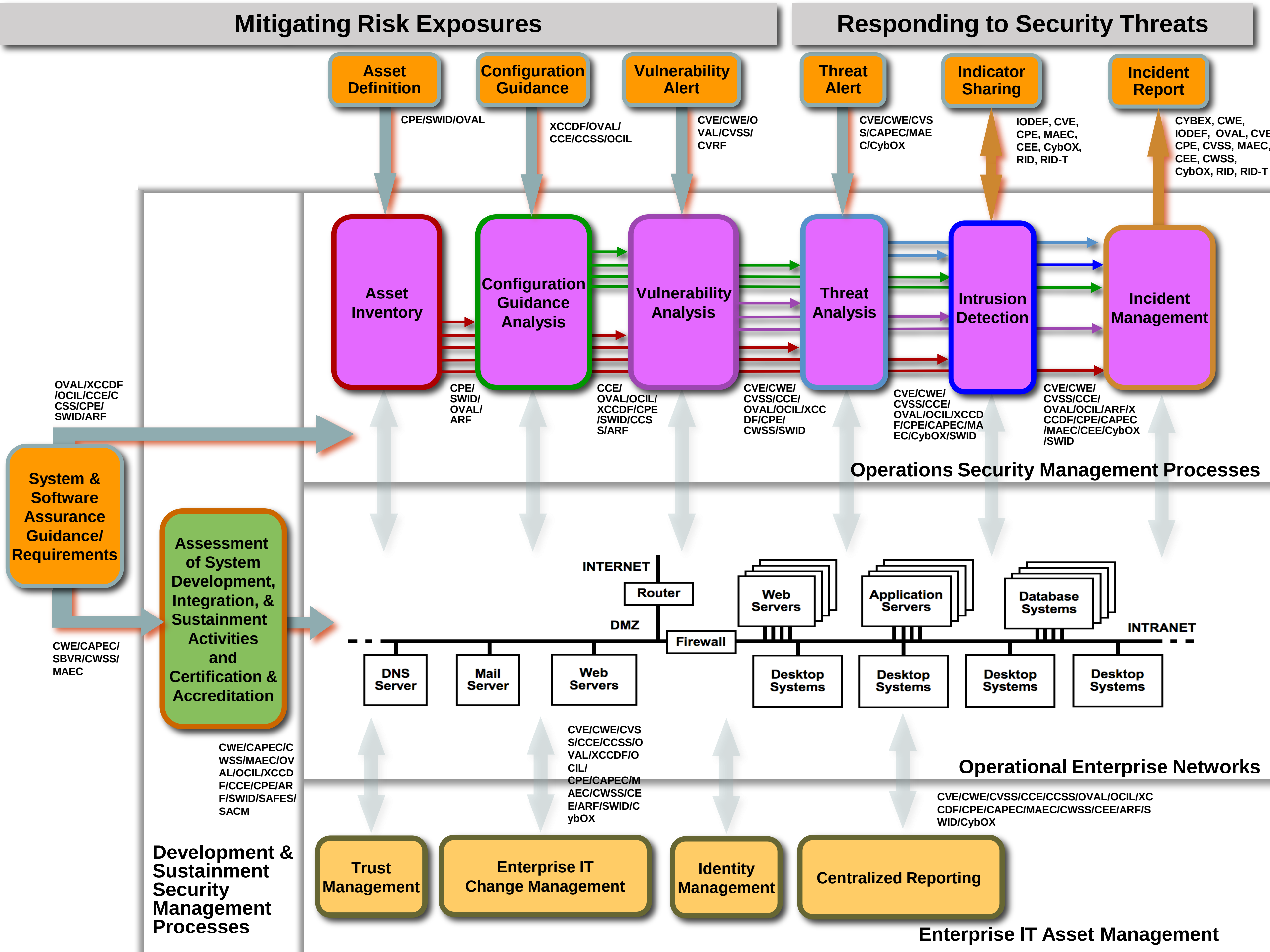
Standardization Efforts leveraged by the Security Content Automation Protocol (SCAP)

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

Efforts focused on mitigating risks and enabling more robust continuous monitoring and faster incident response

★ New FISMA reporting requirements

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



Mitigating Risk Exposures

Responding to Security Threats

Asset Definition

Configuration Guidance

Vulnerability Alert

Threat Alert

Indicator Sharing

Incident Report

CPE/SWID/OVAL

XCCDF/OVAL/
CCE/CCSS/OCIL

CVE/CWE/O
VAL/CVSS/
CVRF

CVE/CWE/CVS
S/CAPEC/MAE
C/CybOX

IODEF, CVE,
CPE, MAEC,
CEE, CybOX,
RID, RID-T

CYBEX, CWE,
IODEF, OVAL, CVE
CPE, CVSS, MAEC,
CEE, CWSS,
CybOX, RID, RID-T

Asset Inventory

Configuration Guidance Analysis

Vulnerability Analysis

Threat Analysis

Intrusion Detection

Incident Management

CPE/
SWID/
OVAL/
ARF

CCE/
OVAL/OCIL/
XCCDF/CPE
/SWID/CCS
S/ARF

CVE/CWE/
CVSS/CCE/
OVAL/OCIL/XCC
DF/CPE/
CWSS/SWID

CVE/CWE/
CVSS/CCE/
OVAL/OCIL/XCCD
F/CPE/CAPEC/MA
EC/CybOX/SWID

CVE/CWE/
CVSS/CCE/
OVAL/OCIL/ARF/X
CCDF/CPE/CAPEC
/MAEC/CEE/CybOX
/SWID

OVAL/XCCDF
/OCIL/CCE/C
CSS/CPE/
SWID/ARF

Operations Security Management Processes

System & Software Assurance Guidance/ Requirements

CWE/CAPEC/
SBVR/CWSS/
MAEC

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

INTERNET

Router

DMZ

Firewall

Web Servers

Application Servers

Database Systems

INTRANET

DNS Server

Mail Server

Web Servers

Desktop Systems

Desktop Systems

Desktop Systems

Desktop Systems

CWE/CAPEC/C
WSS/MAEC/OV
AL/OCIL/XCCD
F/CCE/CPE/AR
F/SWID/SAFES/
SACM

CVE/CWE/CVS
S/CCE/CCSS/O
VAL/XCCDF/O
CIL/
CPE/CAPEC/M
AEC/CWSS/CE
E/ARF/SWID/C
ybOX

Operational Enterprise Networks

CVE/CWE/CVSS/CCE/CCSS/OVAL/OCIL/XC
CDF/CPE/CAPEC/MAEC/CWSS/CEE/ARF/S
WID/CybOX

Development & Sustainment Security Management Processes

Trust Management

Enterprise IT Change Management

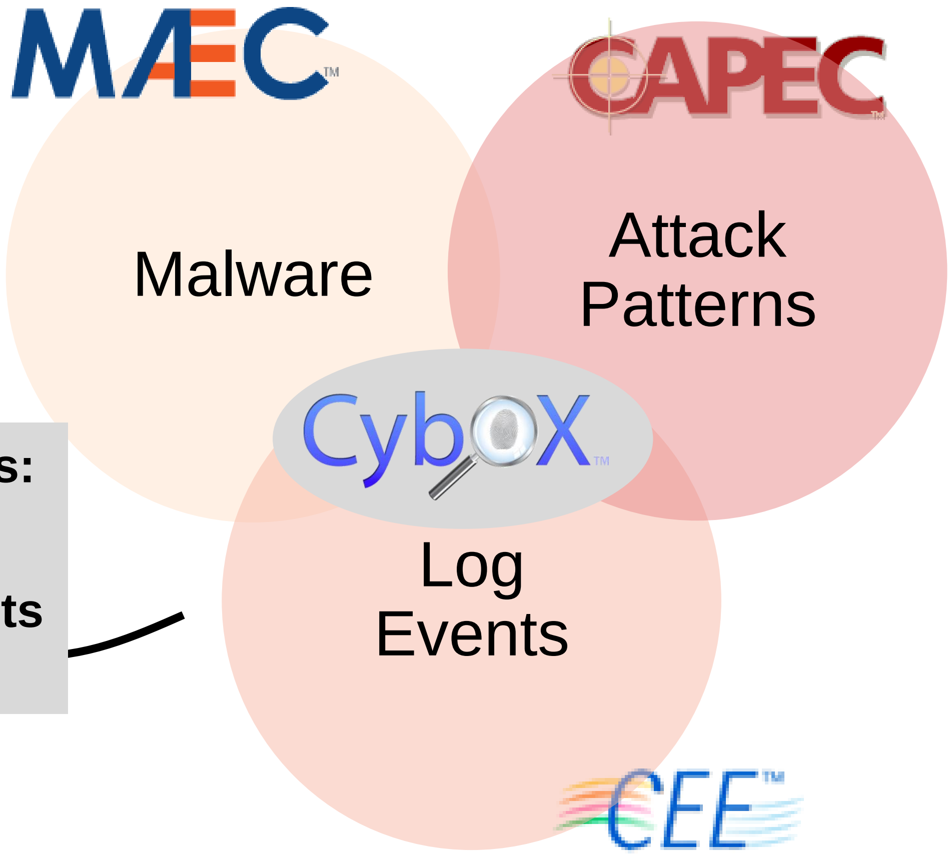
Identity Management

Centralized Reporting

Enterprise IT Asset Management

Evolution of Standardized Representations - Sharing

CVE	Vulnerabilities
CWE	Weaknesses
CAPEC	Attack Patterns
MAEC	Malware Behavior
CybOX	Cyber Observables
?	Threat Indicators



Imports & Extends:

- Object
- Defined Objects
- Actions

CybOX™

- **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)**

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?

What you are looking for

Why should you care about it?

What exactly were they doing?

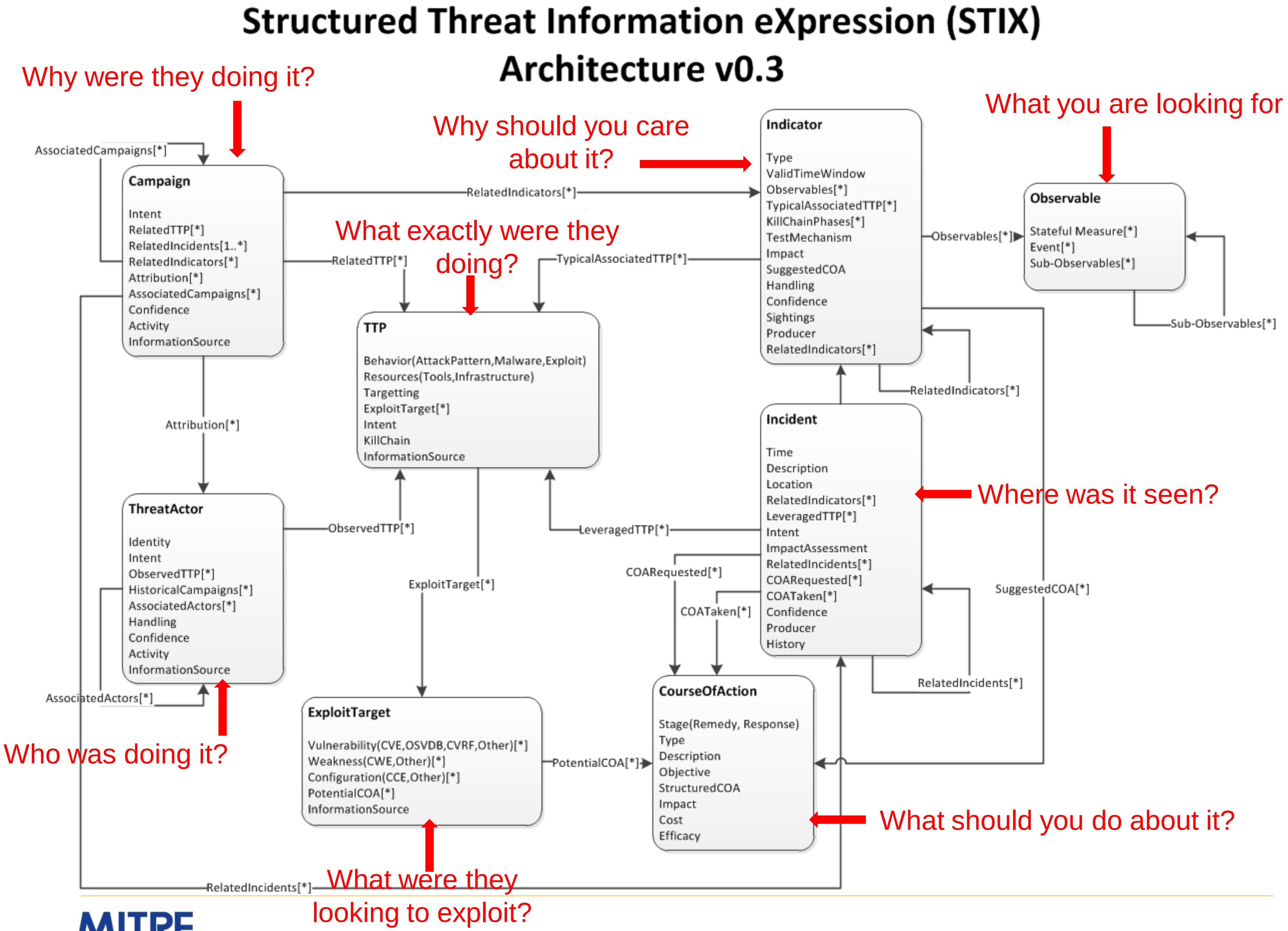
Where was it seen?

Who was doing it?

What should you do about it?

What were they looking to exploit?

STIX Architecture



Structured Threat Information eXpression (STIX) Architecture v0.3

Campaign

Intent
RelatedTTP[*]
RelatedIncidents[1..*]
RelatedIndicators[*]
Attribution[*]
AssociatedCampaigns[*]
Confidence
Activity
InformationSource

Indicator

Type
ValidTimeWindow
Observables[*]
TypicalAssociatedTTP[*]
KillChainPhases[*]
TestMechanism
Impact
SuggestedCOA
Handling
Confidence
Sightings
Producer
RelatedIndicators[*]

Observable

Stateful Measure[*]
Event[*]
Sub-Observables[*]

TTP

Behavior(AttackPattern,Malware,Exploit)
Resources(Tools,Infrastructure)
Targetting
ExploitTarget[*]
Intent
KillChain
InformationSource

ThreatActor

Identity
Intent
ObservedTTP[*]
HistoricalCampaigns[*]
AssociatedActors[*]
Handling
Confidence
Activity
InformationSource

Incident

Time
Description
Location
RelatedIndicators[*]
LeveragedTTP[*]
Intent
ImpactAssessment
RelatedIncidents[*]
COARequested[*]
COATaken[*]
Confidence
Producer
History

ExploitTarget

Vulnerability(CVE,OSVDB,CVRF,Other)[*]
Weakness(CWE,Other)[*]
Configuration(CCE,Other)[*]
PotentialCOA[*]
InformationSource

CourseOfAction

Stage(Remedy, Response)
Type
Description
Objective
StructuredCOA
Impact
Cost
Efficacy

Why were they doing it?

Why should you care about it?

What you are looking for

What exactly were they doing?

Where was it seen?

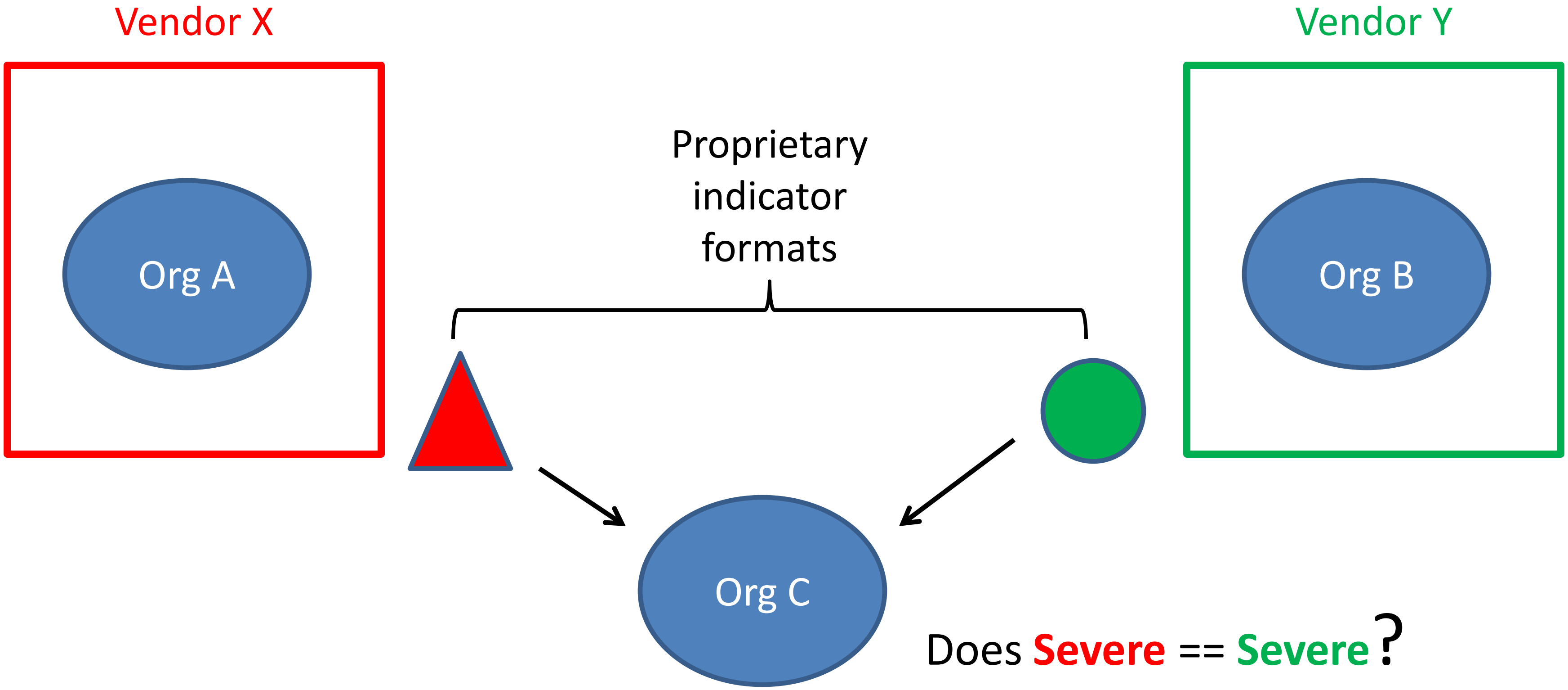
Who was doing it?

What should you do about it?

What were they looking to exploit?

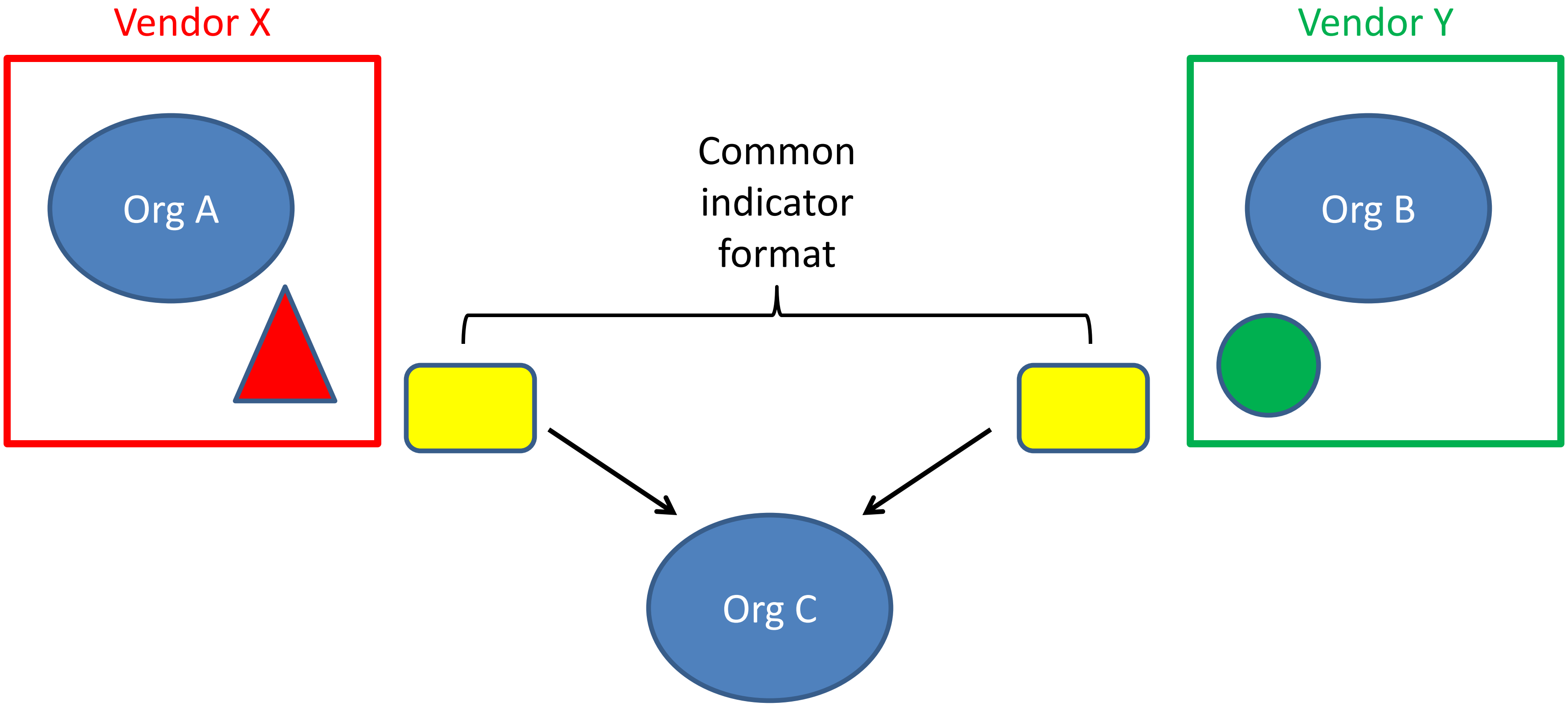
MITRE

Sharing Challenges



- 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

Free Resources and Events

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

SOFTWARE ASSURANCE FORUM

Homeland Security

Commerce

National Defense

BUILDING SECURITY IN

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

Next SwA Working Groups at MITRE, McLean, VA – 27-29 2012

BUILDING SECURITY IN
SOFTWARE ASSURANCE

SOFTWARE ASSURANCE FORUM

"Building Security In"

https://buildsecurityin.us-cert.gov/swa

Homeland Security

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