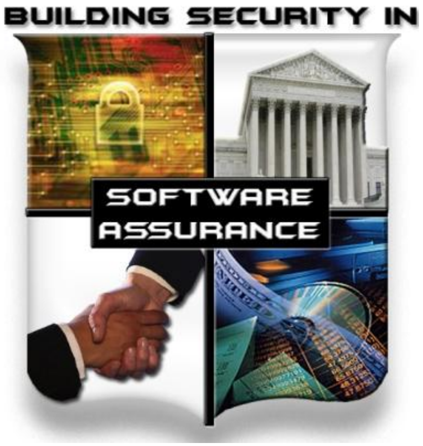
Software Assurance:

Mitigating the Sources of Risk that contribute to Zero-Day Cyber Attacks



Software Supply Chain Practices

Nov 29, 2011

Homeland Security

Joe Jarzombek, PMP, CSSLP
Director for Software Assurance
National Cyber Security Division

Software Supply Chain Practices

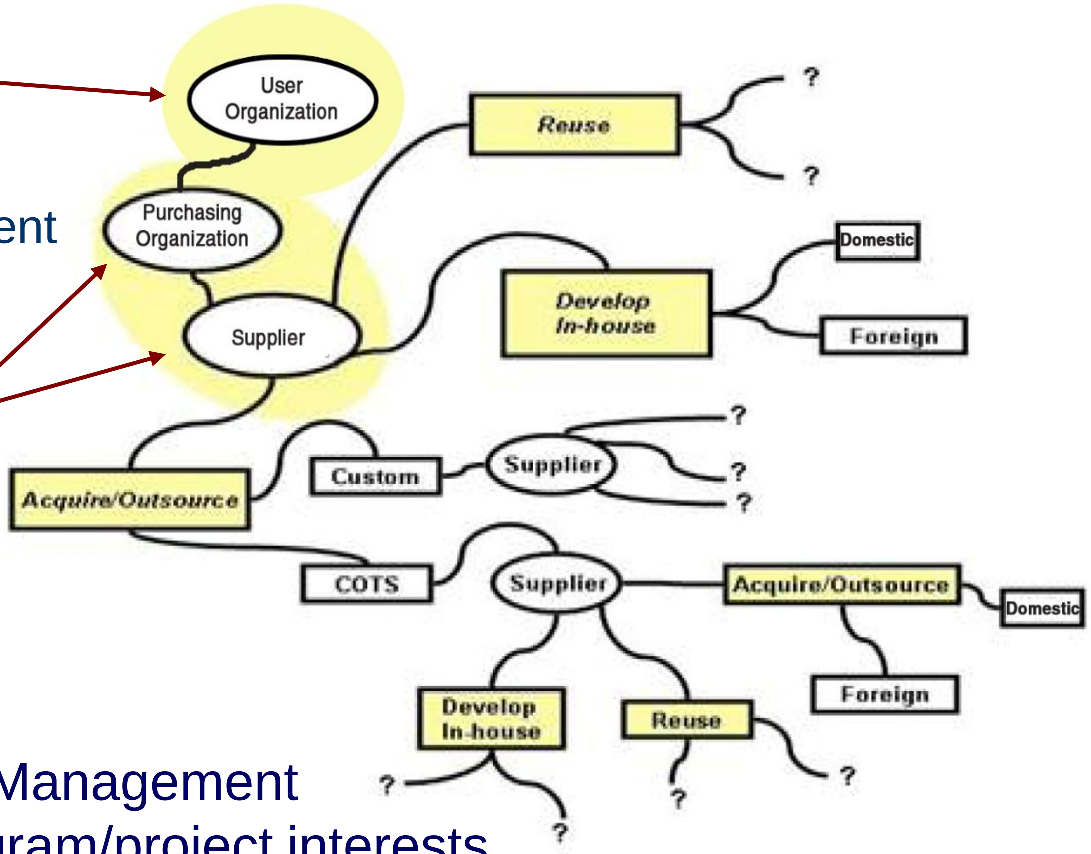
- How can we mitigate or reduce residual security debt being passed or inherited by using organizations?
  - – What constitutes residual security debt in the supply chain?
  - – How would considerations for security processes contribute to determining capabilities of supply chain actors?
  - – How do security-enhanced security practices measurably contribute to reductions in exploitable weaknesses in software products, systems, and services?
  - – Can code behavior be well-understood not to possess malicious constructs or exploitable weaknesses

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

- 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

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

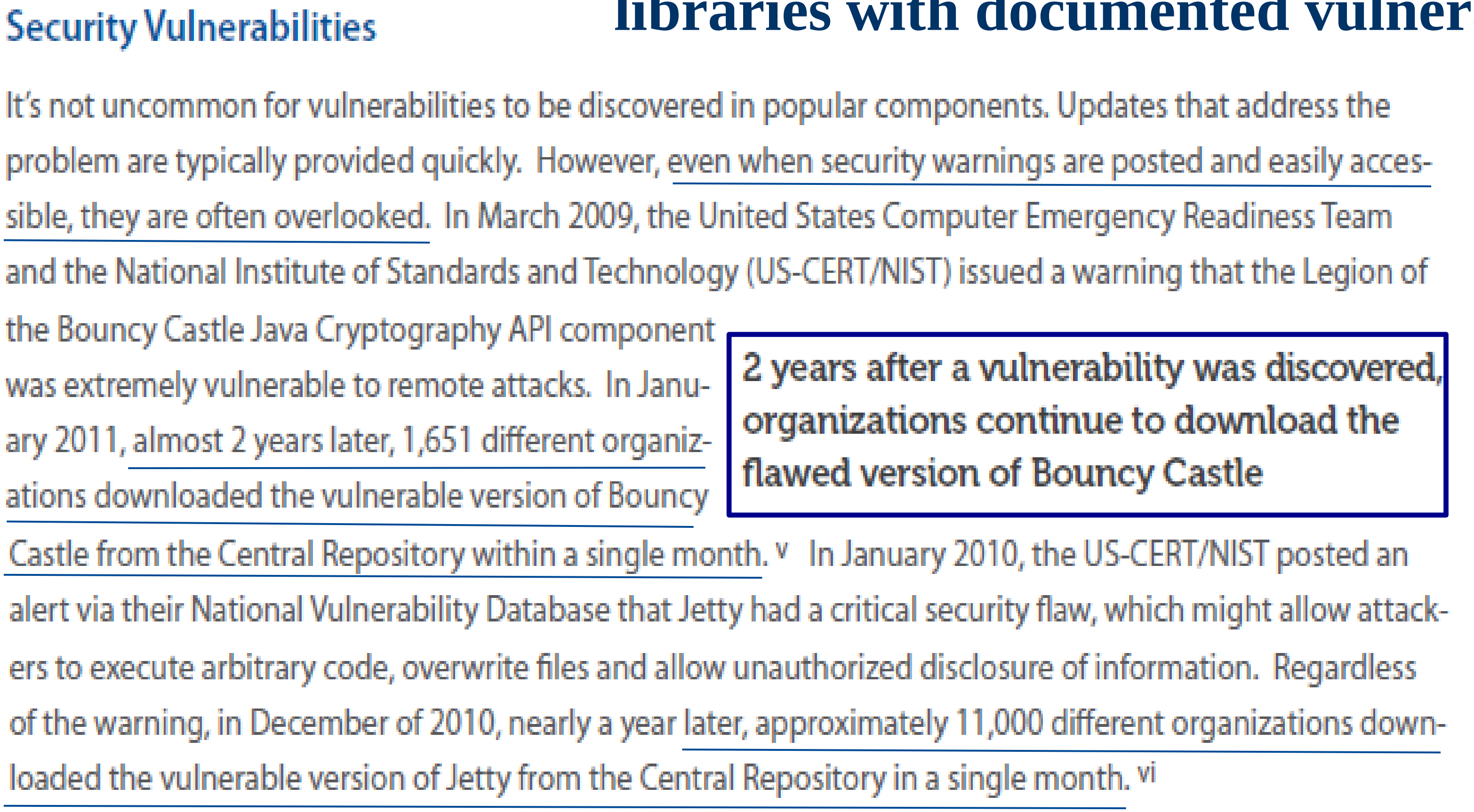
- Enterprise-Level:
  - Regulatory compliance
  - Changing threat environment
  - Business Case
- Program/Project-Level:
  - Cost
  - Schedule
  - Performance



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 security policies with respect to the mission.
3. Determine how much risk the organization can afford and who is accountable for that risk.

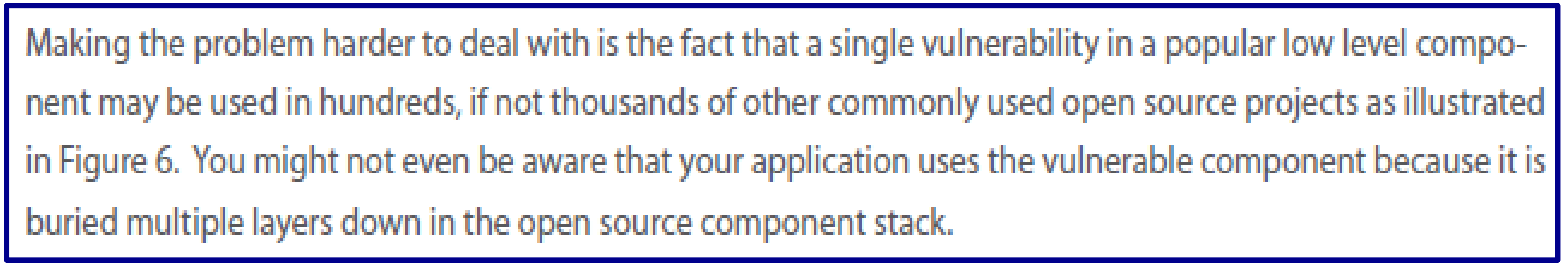
Thousands of downloads from open libraries with documented vulnerabilities



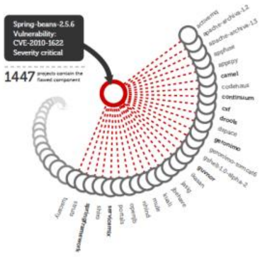
Security Vulnerabilities

It's not uncommon for vulnerabilities to be discovered in popular components. Updates that address the problem are typically provided quickly. However, even when security warnings are posted and easily accessible, they are often overlooked. In March 2009, the United States Computer Emergency Readiness Team and the National Institute of Standards and Technology (US-CERT/NIST) issued a warning that the Legion of the Bouncy Castle Java Cryptography API component was extremely vulnerable to remote attacks. In January 2011, almost 2 years later, 1,651 different organizations downloaded the vulnerable version of Bouncy Castle from the Central Repository within a single month.[v] In January 2010, the US-CERT/NIST posted an alert via their National Vulnerability Database that Jetty had a critical security flaw, which might allow attackers to execute arbitrary code, overwrite files and allow unauthorized disclosure of information. Regardless of the warning, in December of 2010, nearly a year later, approximately 11,000 different organizations downloaded the vulnerable version of Jetty from the Central Repository in a single month.[vi]

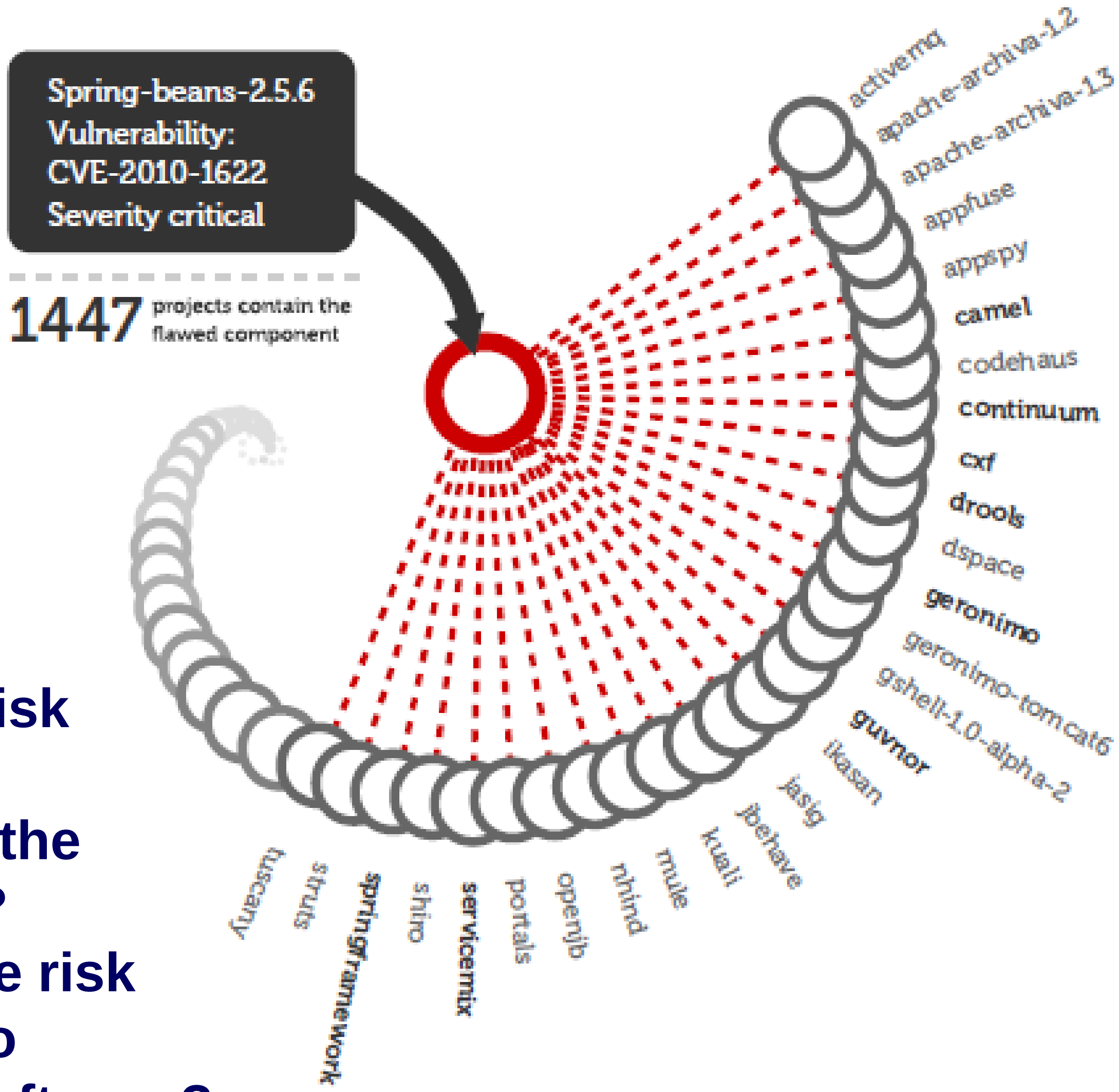
2 years after a vulnerability was discovered, organizations continue to download the flawed version of Bouncy Castle



Making the problem harder to deal with is the fact that a single vulnerability in a popular low level component may be used in hundreds, if not thousands of other commonly used open source projects as illustrated in Figure 6. You might not even be aware that your application uses the vulnerable component because it is buried multiple layers down in the open source component stack.



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



Who makes risk decisions?
Who inherits the residual risk?
Who owns the risk attributable to exploitable software?

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

Software Assurance

The level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidently inserted at anytime during its life cycle and that the software functions as intended. *Derived From: CNSSI-4009*

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

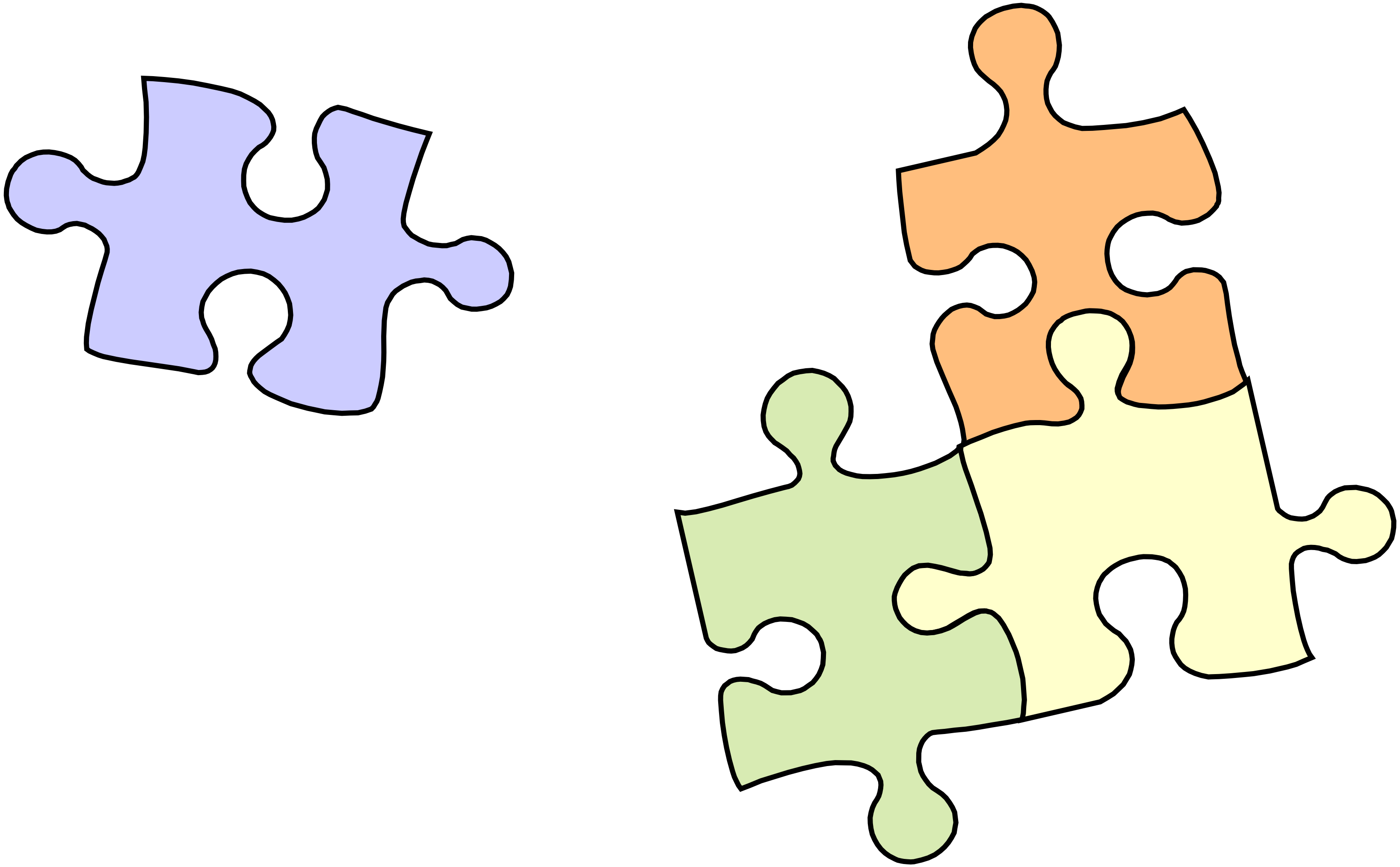
Automation

Languages, tools, enumerations and repositories

throughout the Lifecycle

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

Automation is *one piece*



of the SwA puzzle.

What is the context?

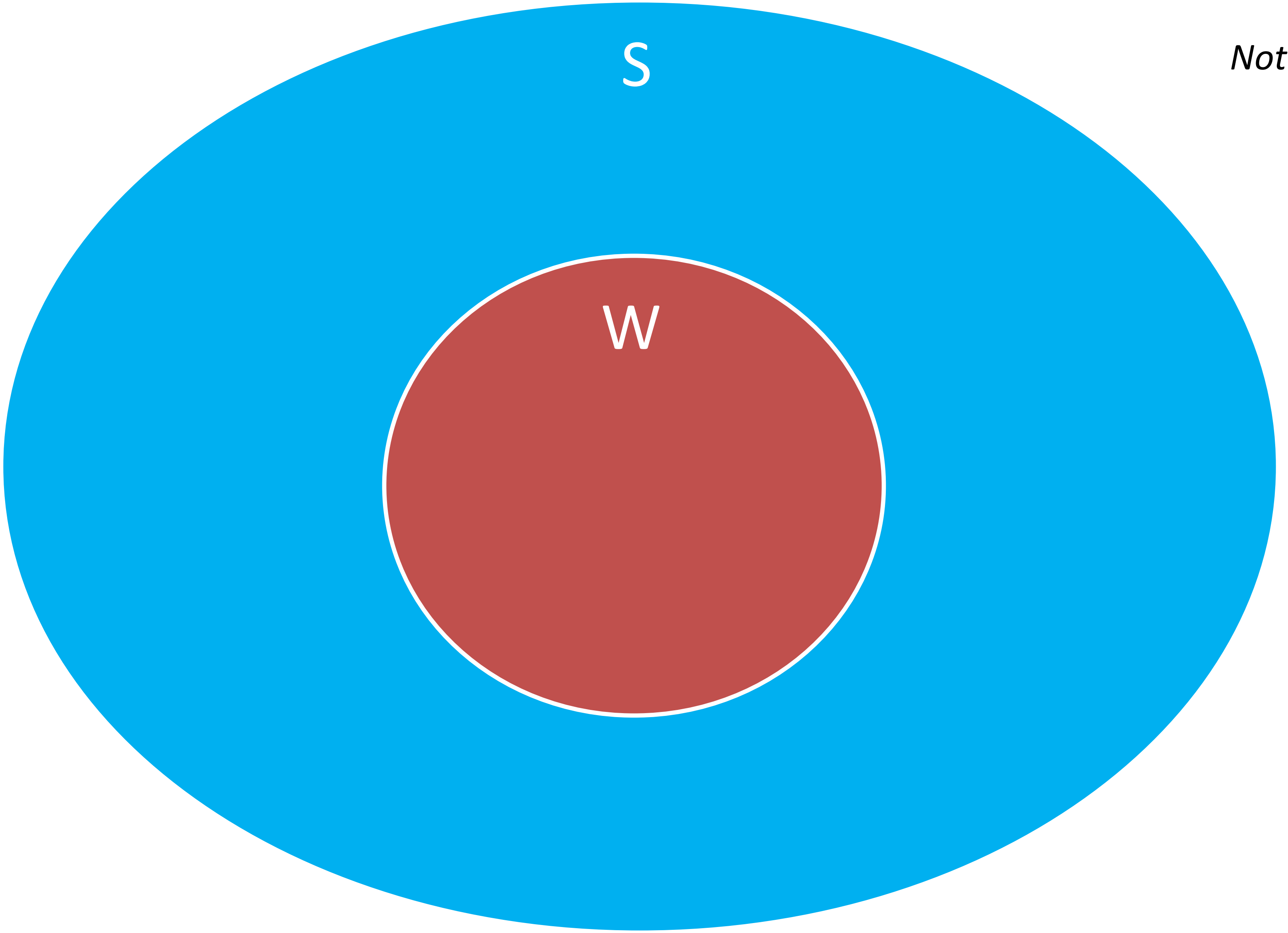
Where can automation help - *today*?

What problems are we trying to solve?

Where do we start?

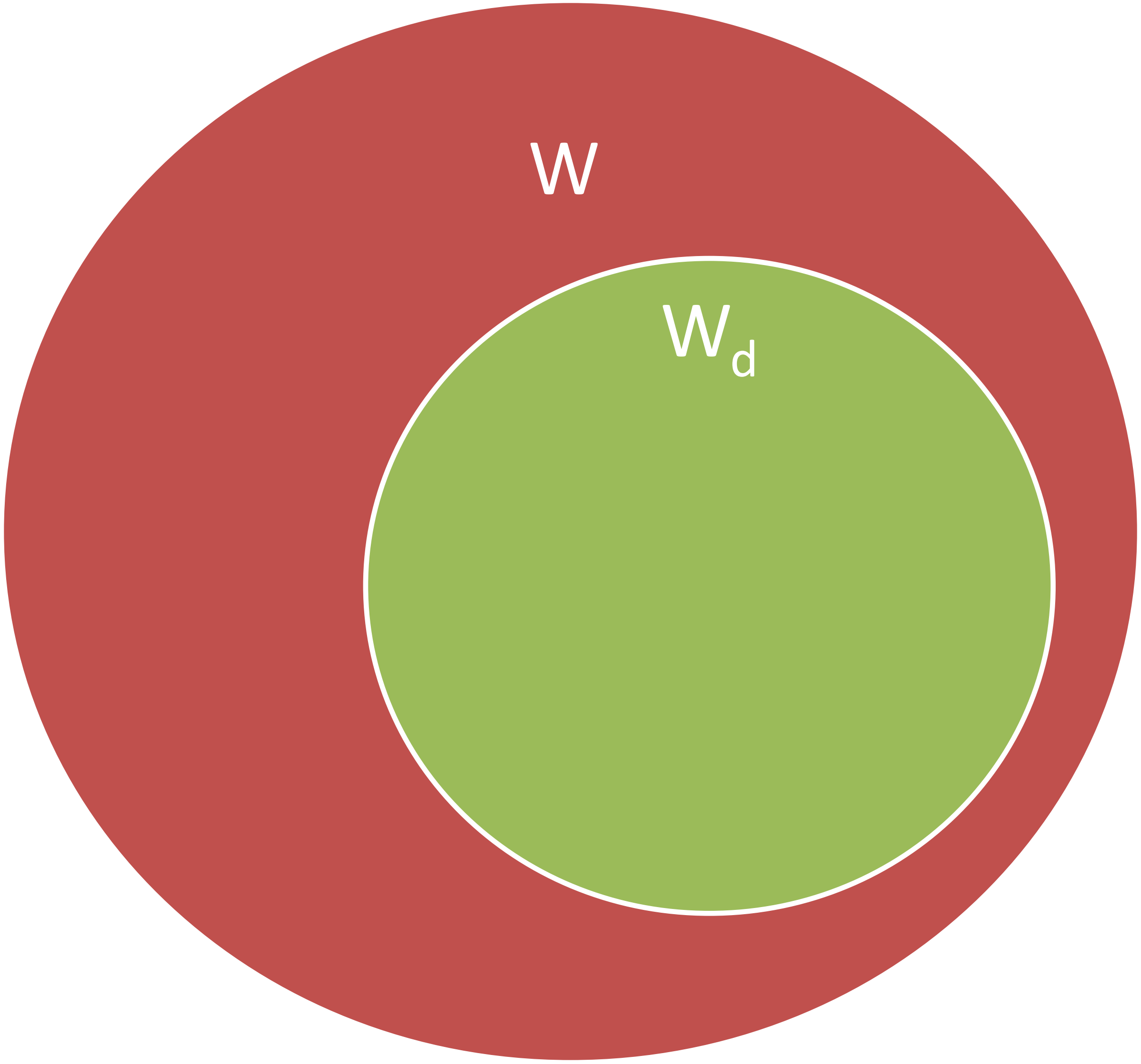
S: The set of all software in existence at some point in time

Notional



W: The set of all instances of software weaknesses in S

Notional



W_d: The set of all *discovered* software weaknesses in W

There are many definitions of “weakness” -- *in this context*

A *(software) weakness* is a property of software/ systems that, under the right conditions, may permit unintended / unauthorized behavior.

*Common Weakness Enumeration (CWE) http://cwe.mitre.org/

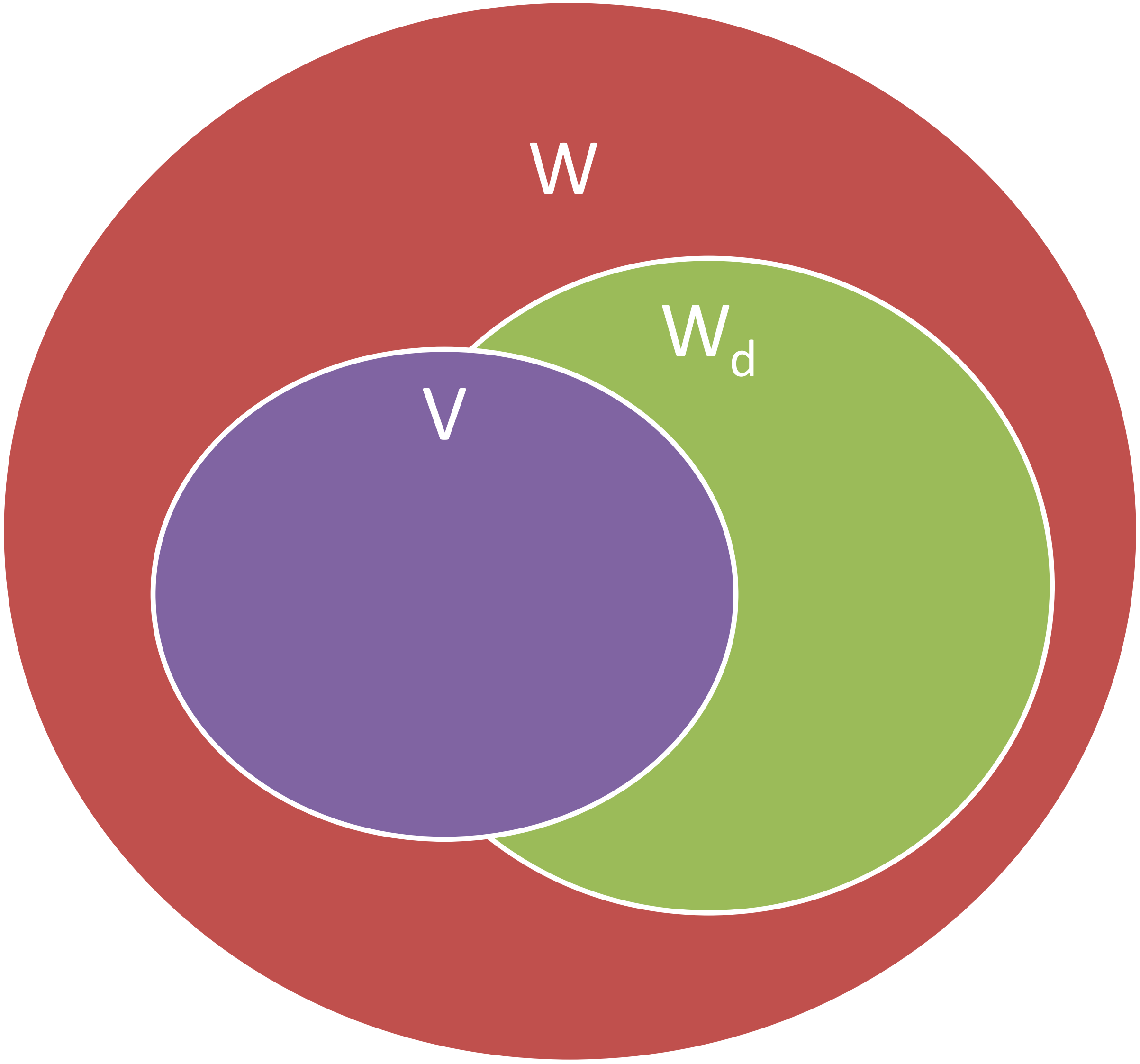
There are many definitions of “vulnerability” -- *in this context*:

A *(software) vulnerability* is a collection of one or more weaknesses that contain the right conditions to permit unauthorized parties to force the software to perform unintended behavior (a.k.a. “is exploitable”)

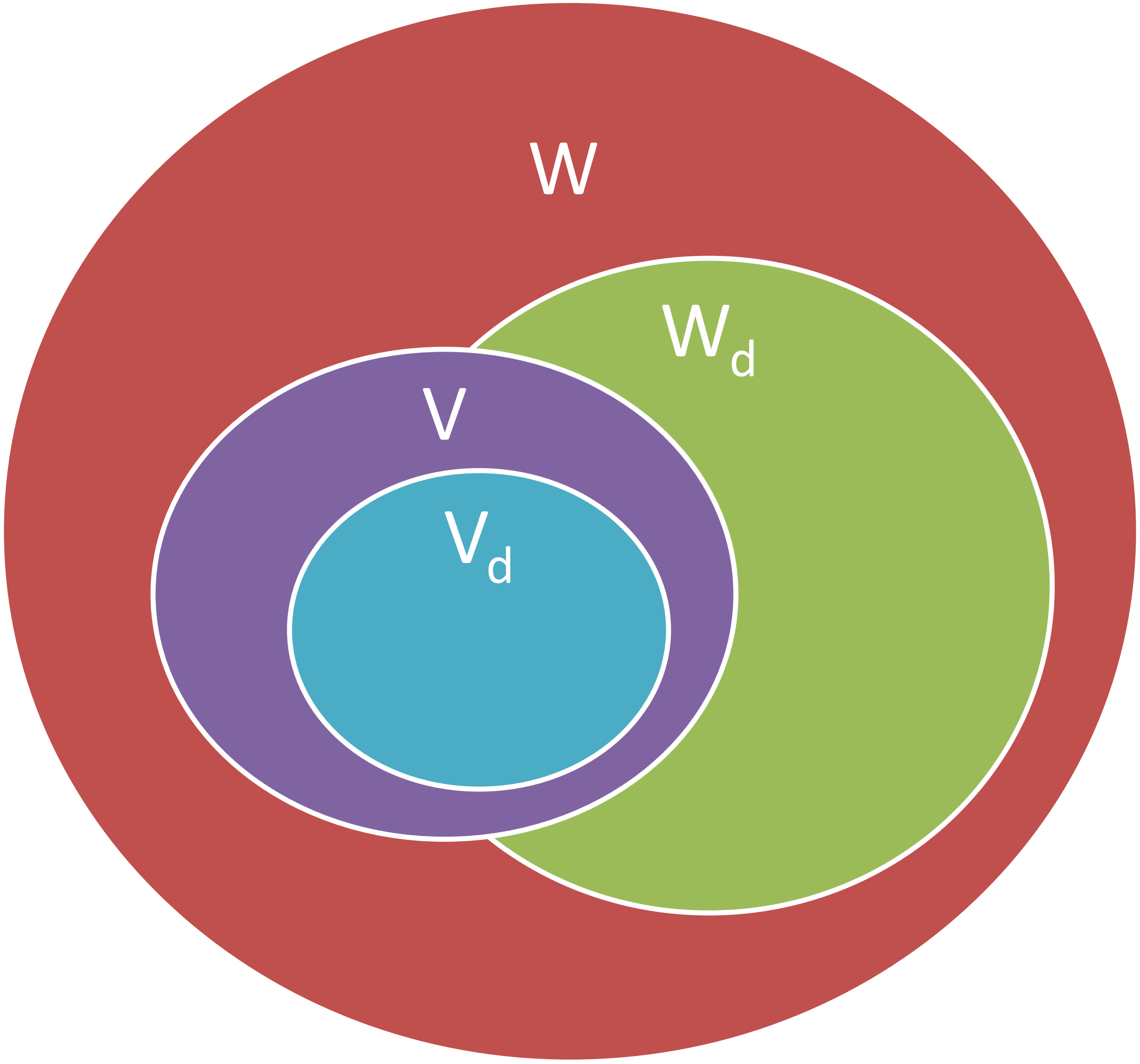
*Common Vulnerabilities and Exposures (CVE) http://cwe.mitre.org/

* Part of ITU-T Cyber Information Exchange (CYBEX) series 1500

Notional

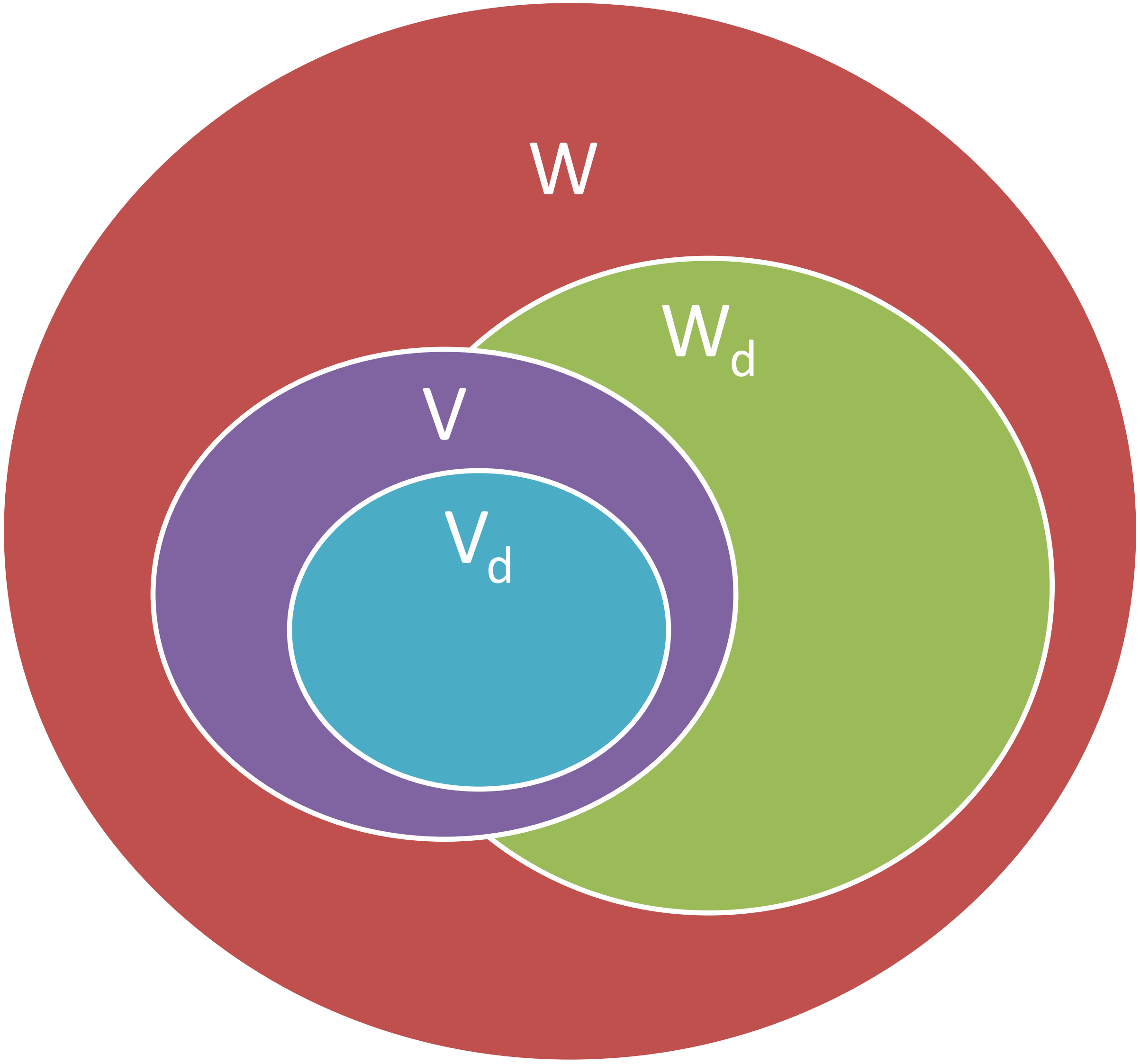


V: The set of all vulnerabilities in W



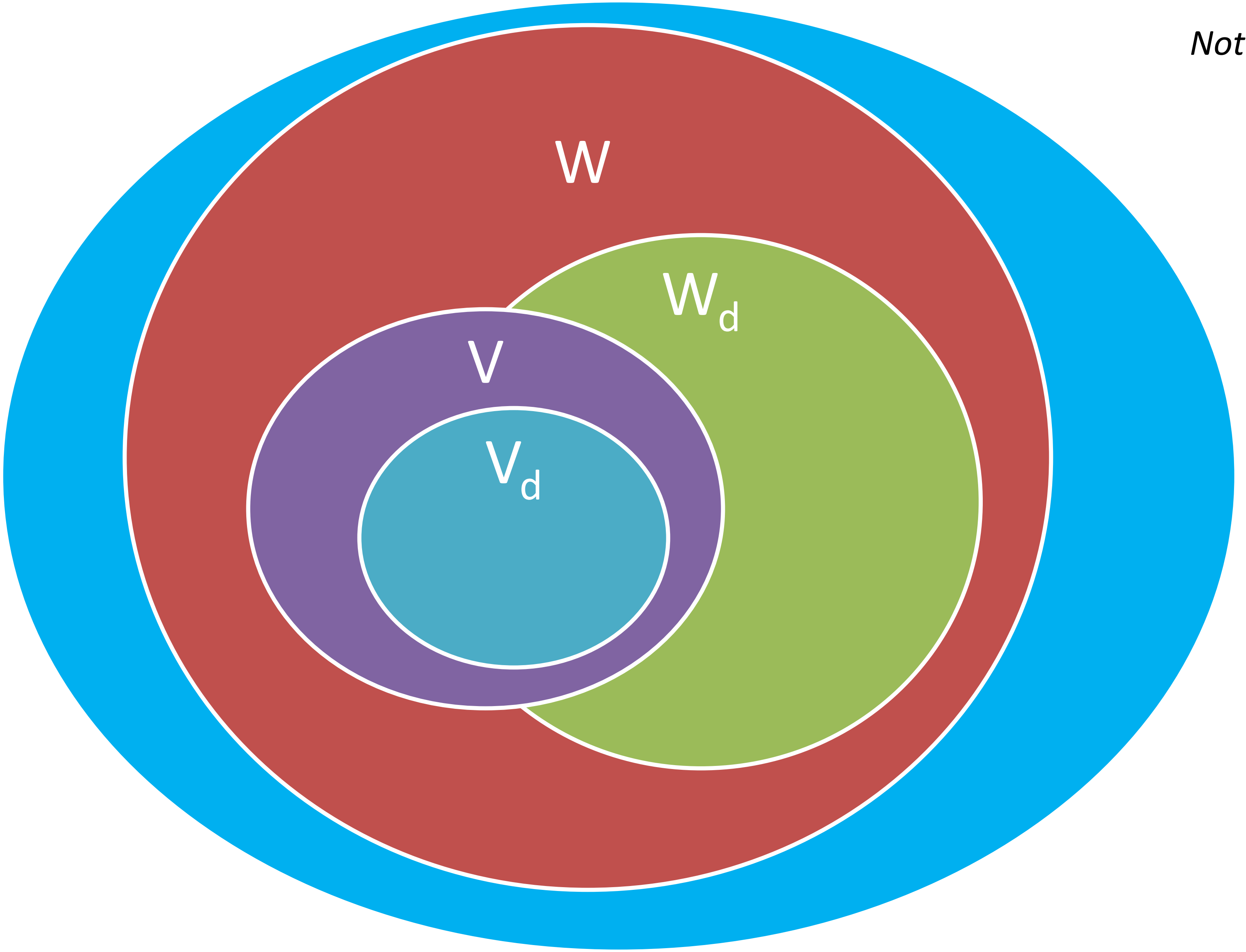
V_d: The set of all *discovered* vulnerabilities in V

Notional



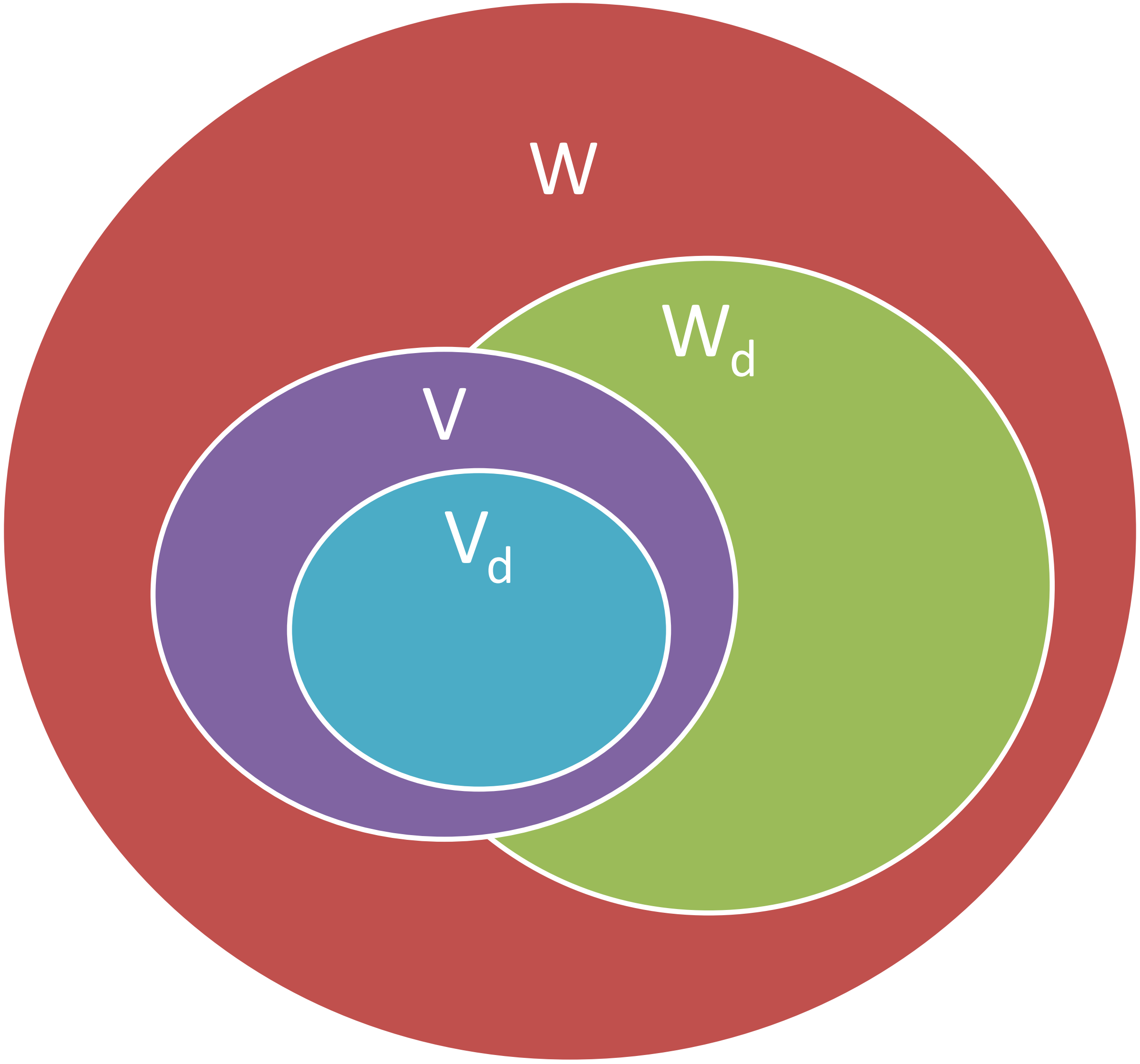
What does the future hold?

Notional



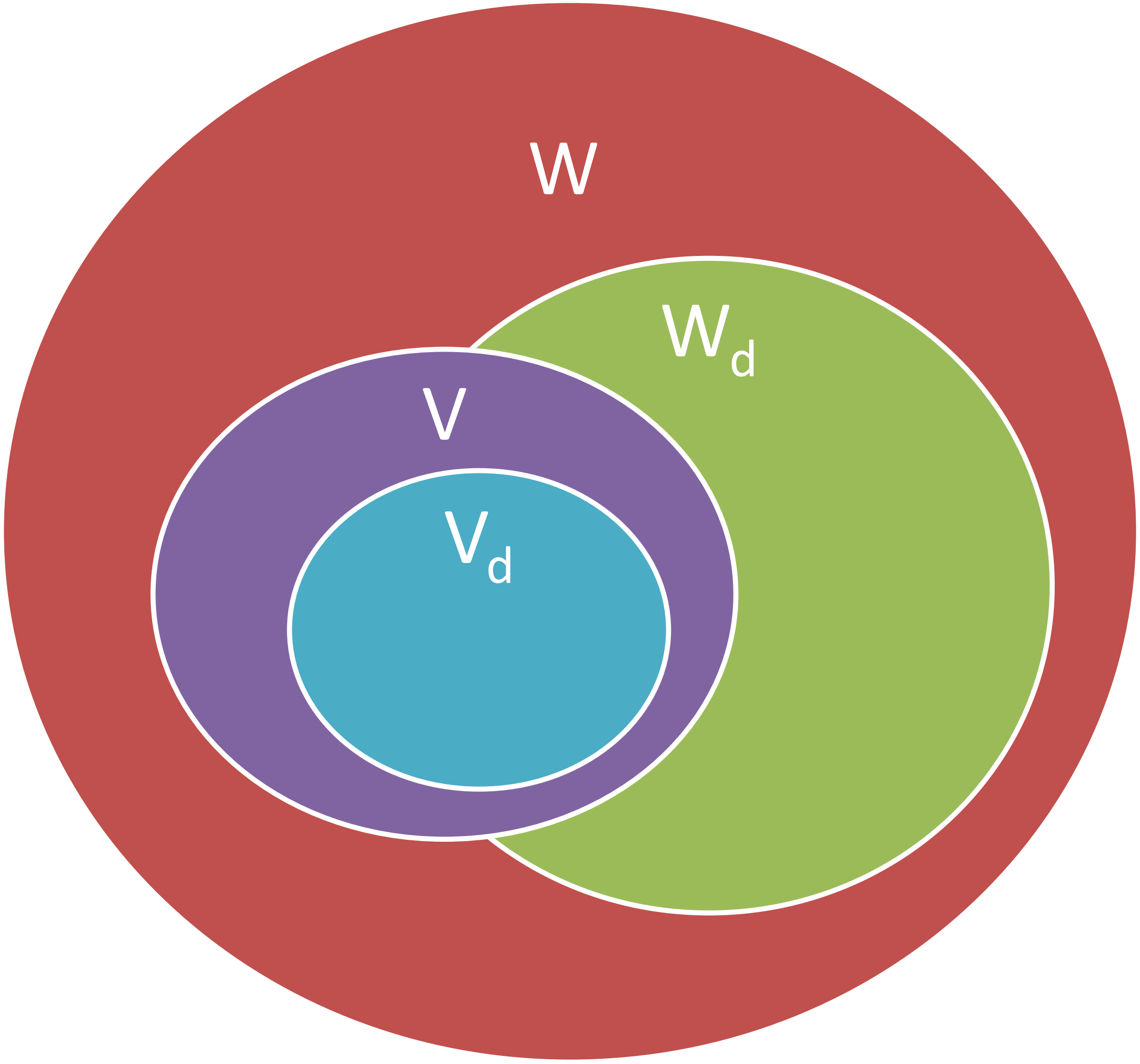
We know it's *not* this, at least not in the near-term

Notional



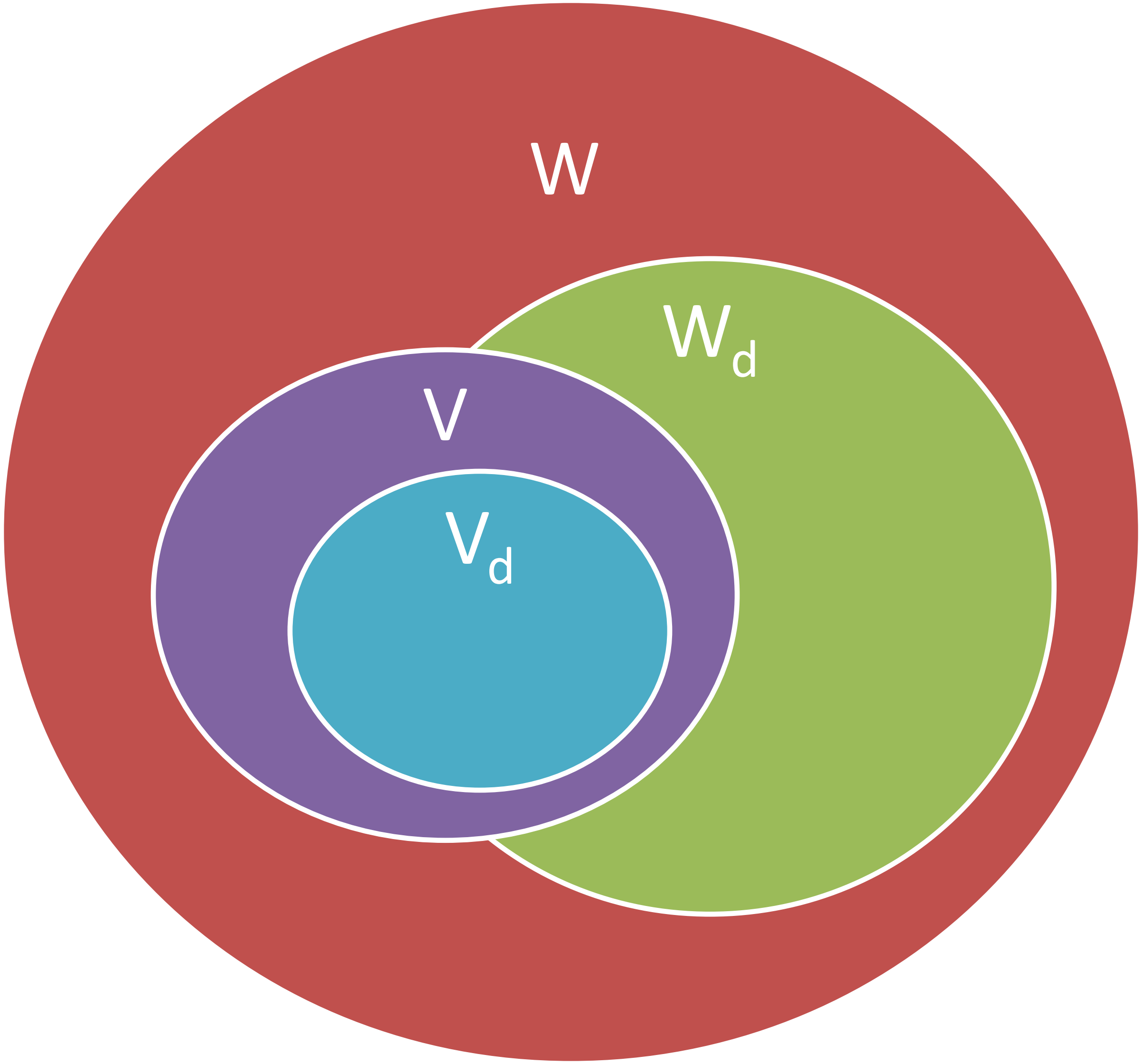
Maybe the problem grows unbounded?

Notional



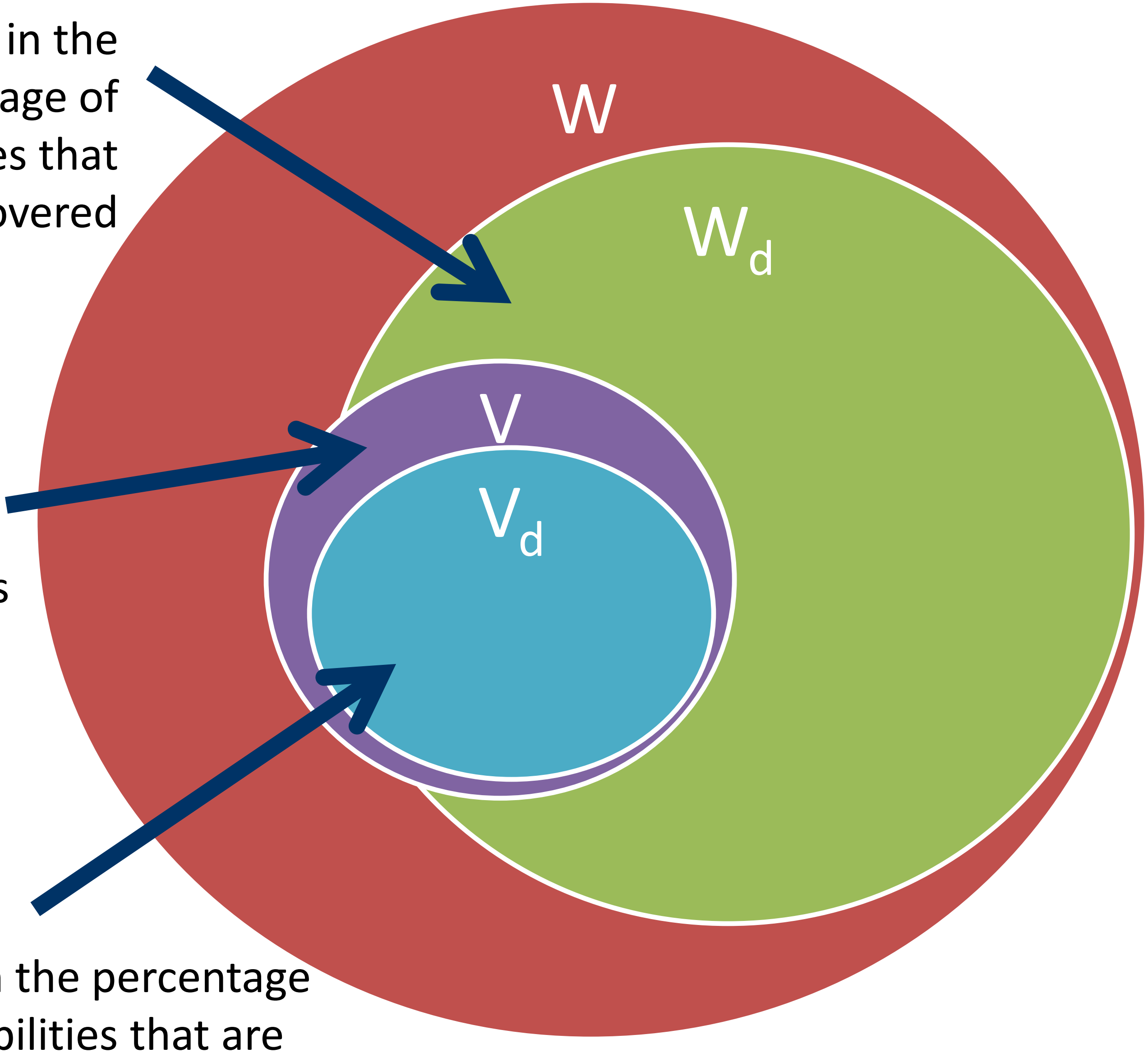
Maybe just some things get worse?

Notional



One reasonable near-term goal

Notional



Increase in the percentage of weaknesses that are discovered

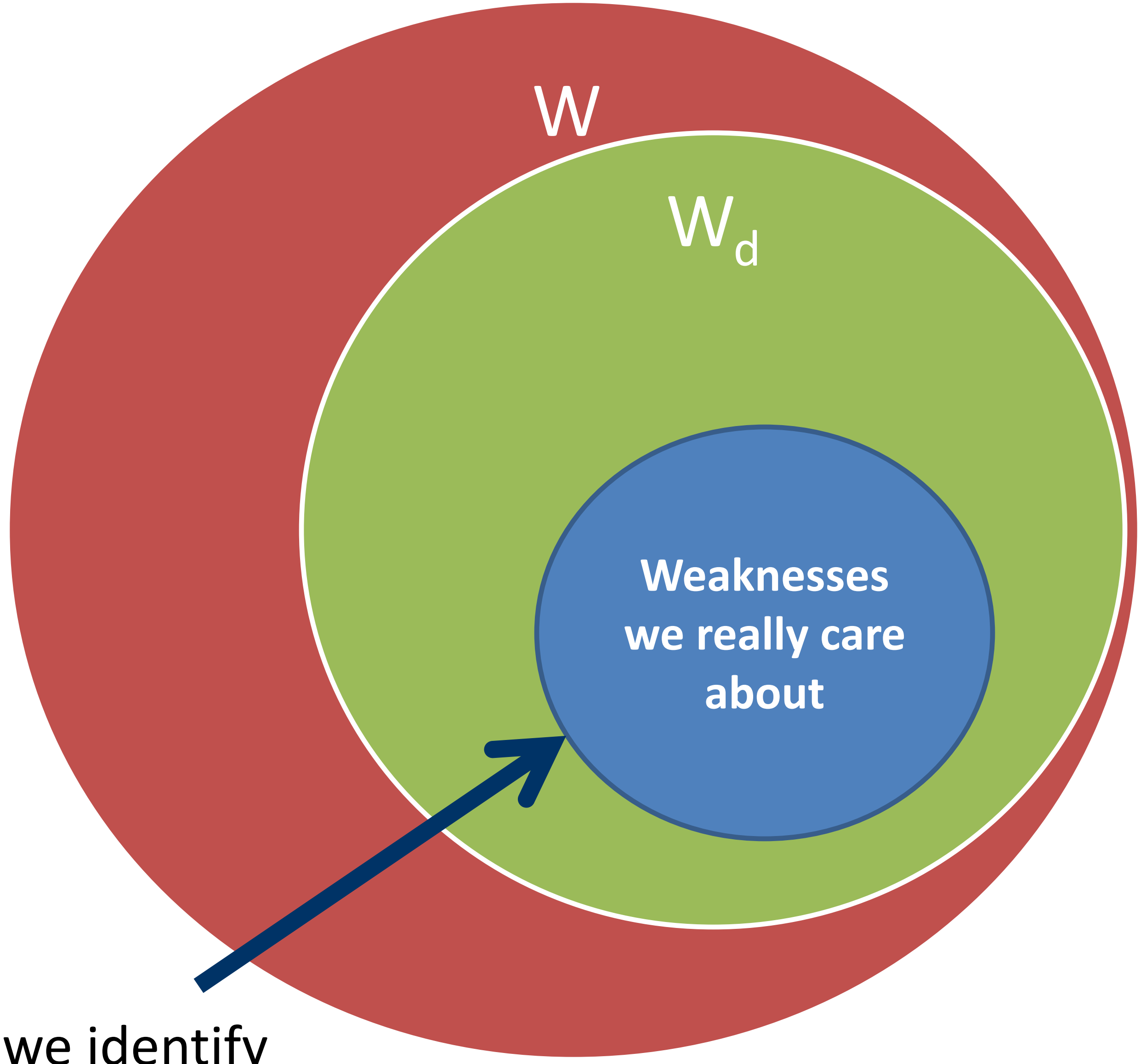
Decreased number of vulnerabilities

Increase in the percentage of vulnerabilities that are discovered

Is this really better? Yes

For the software we're responsible for

Notional

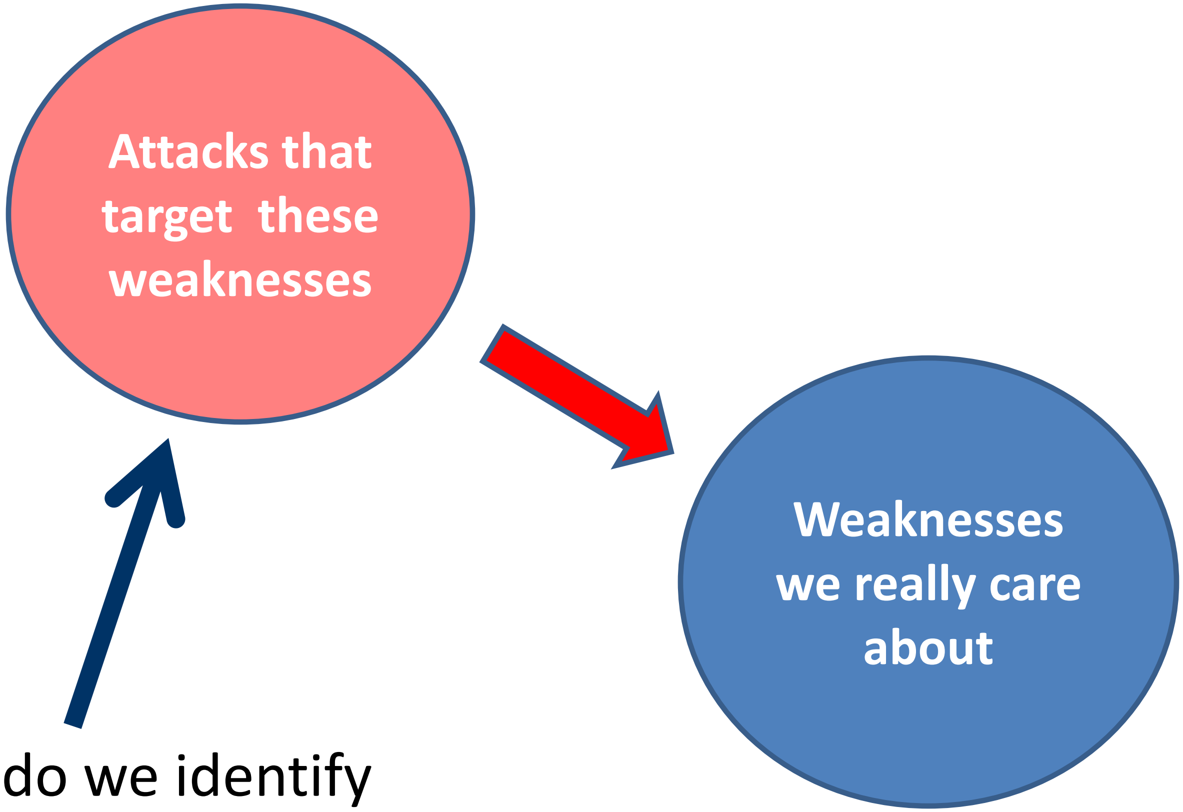


How do we identify these?

which weaknesses are most important?

For the software we’re responsible for

Notional


Attacks that target these weaknesses
Weaknesses we really care about

How do we identify these?

how can those weaknesses be attacked?

Weaknesses and attacks relevant here

and here

and here

and here

May be different than those relevant here

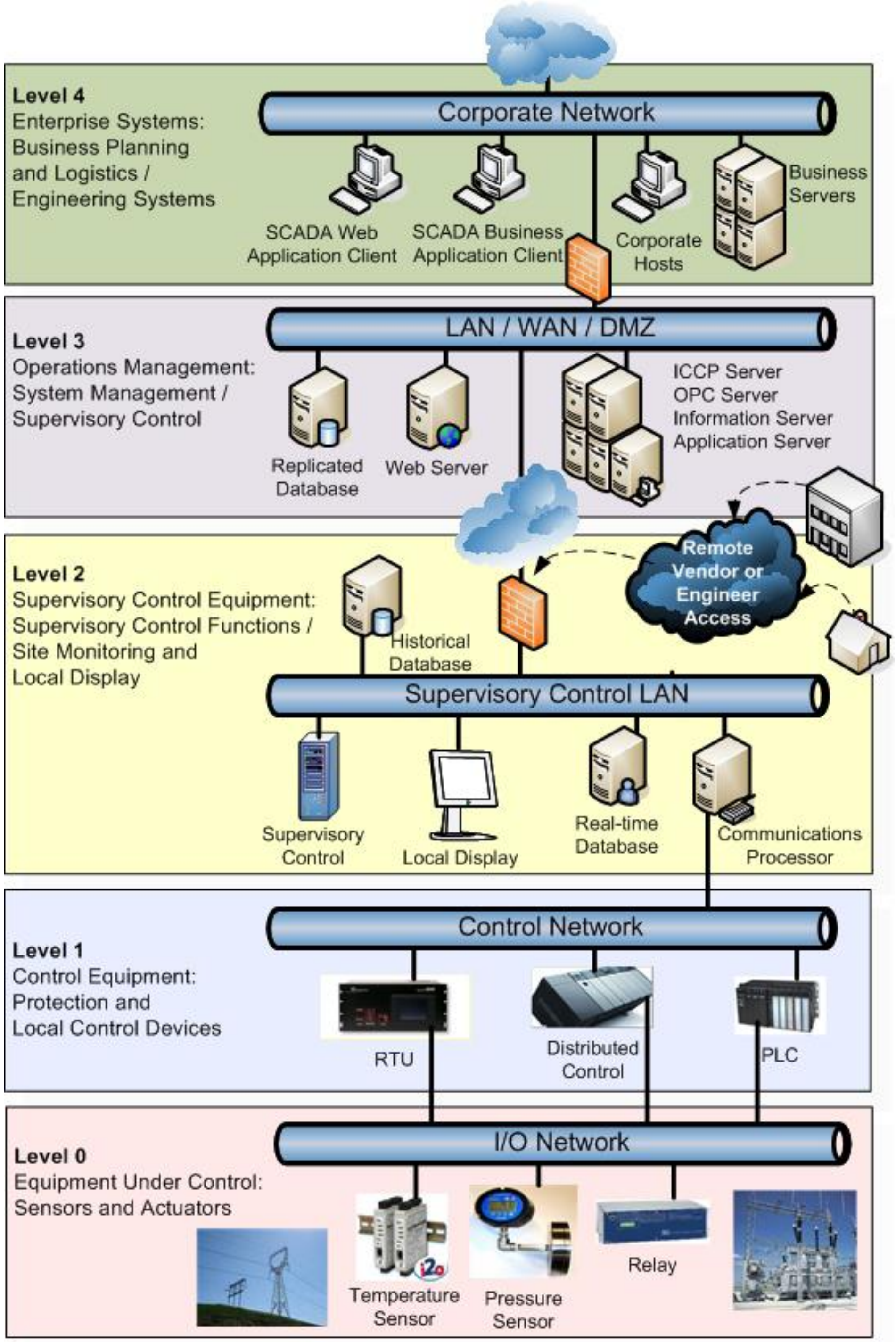
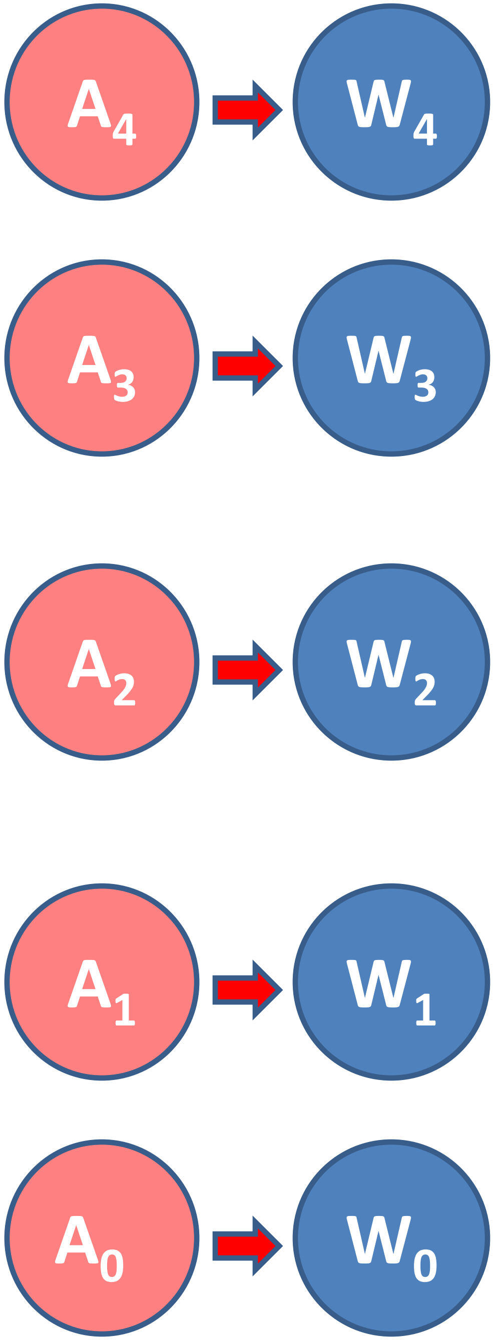

Level 4
Enterprise Systems: Business Planning and Logistics / Engineering Systems
Corporate Network
SCADA Web Application Client
SCADA Business Application Client
Corporate Hosts
Business Servers
Level 3
Operations Management: System Management / Supervisory Control
LAN / WAN / DMZ
Replicated Database
Web Server
ICCP Server
OPC Server
Information Server
Application Server
Level 2
Supervisory Control Equipment: Supervisory Control Functions / Site Monitoring and Local Display
Historical Database
Remote Vendor or Engineer Access
Supervisory Control LAN
Supervisory Control
Local Display
Real-time Database
Communications Processor
Level 1
Control Equipment: Protection and Local Control Devices
Control Network
RTU
Distributed Control
PLC
Level 0
Equipment Under Control: Sensors and Actuators
I/O Network
Temperature Sensor
Pressure Sensor
Relay

Diagram from: ***Vulnerability Analysis of Energy Delivery Control Systems***, INL, September 2011



$A_4 \rightarrow W_4$

$A_3 \rightarrow W_3$

$A_2 \rightarrow W_2$

$A_1 \rightarrow W_1$

$A_0 \rightarrow W_0$

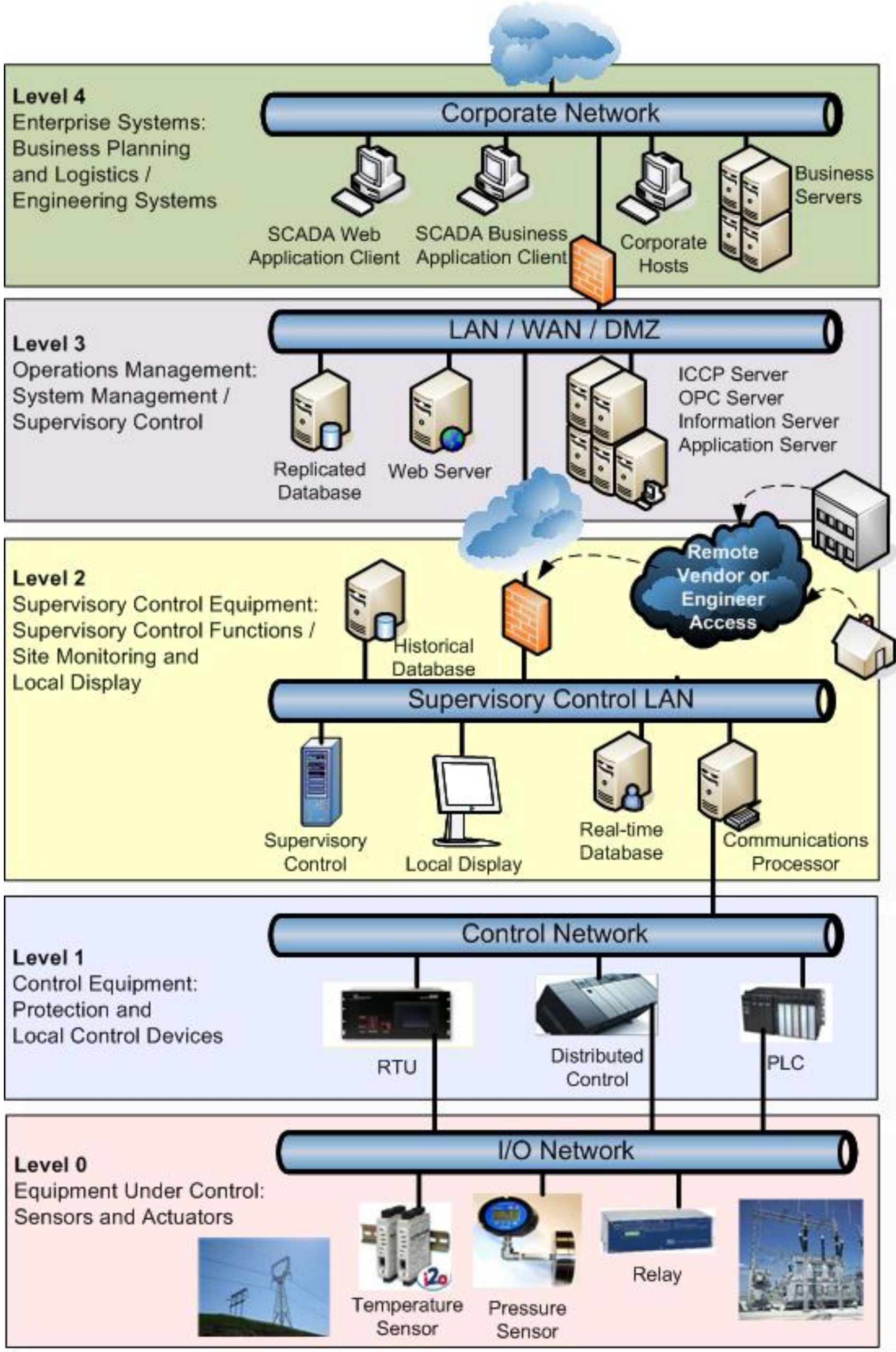
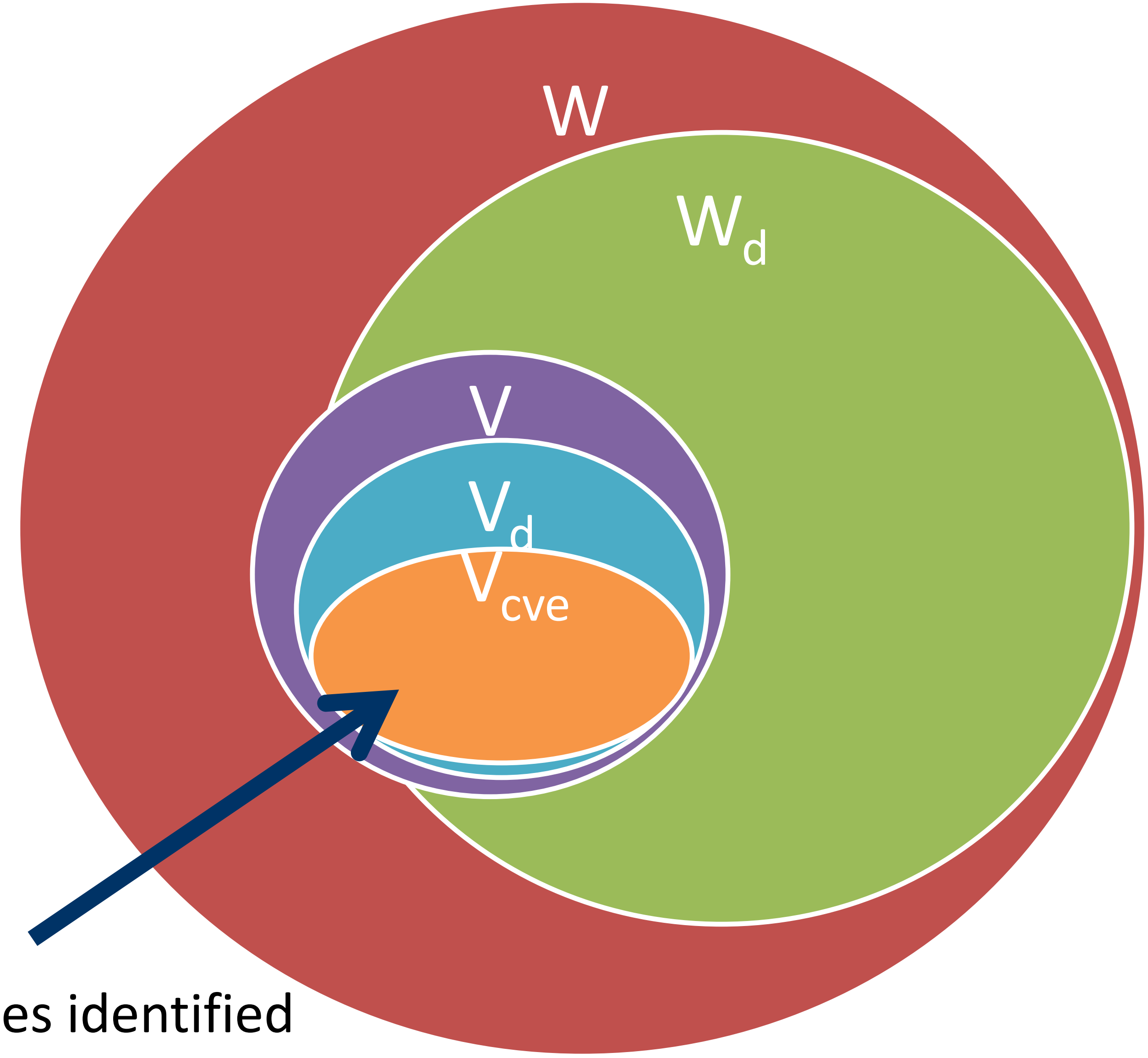

Level 4
Enterprise Systems: Business Planning and Logistics / Engineering Systems
Corporate Network
SCADA Web Application Client
SCADA Business Application Client
Corporate Hosts
Business Servers
Level 3
Operations Management: System Management / Supervisory Control
LAN / WAN / DMZ
Replicated Database
Web Server
ICCP Server
OPC Server
Information Server
Application Server
Remote Vendor or Engineer Access
Level 2
Supervisory Control Equipment: Supervisory Control Functions / Site Monitoring and Local Display
Historical Database
Supervisory Control LAN
Supervisory Control
Local Display
Real-time Database
Communications Processor
Level 1
Control Equipment: Protection and Local Control Devices
Control Network
RTU
Distributed Control
PLC
Level 0
Equipment Under Control: Sensors and Actuators
I/O Network
Temperature Sensor
Pressure Sensor
Relay

Diagram from: ***Vulnerability Analysis of Energy Delivery Control Systems***, INL, September 2011

For the software we're responsible for

Notional



Vulnerabilities identified with a CVE are a good starting point

where should we start?

Dictionary of publicly-disclosed vulnerabilities with unique identifiers

- CVE ID
- Status
- Description
- References

Note: Each CVE entry is the result of expert analysis to verify, de-conflict and de-duplicate public vulnerability disclosures

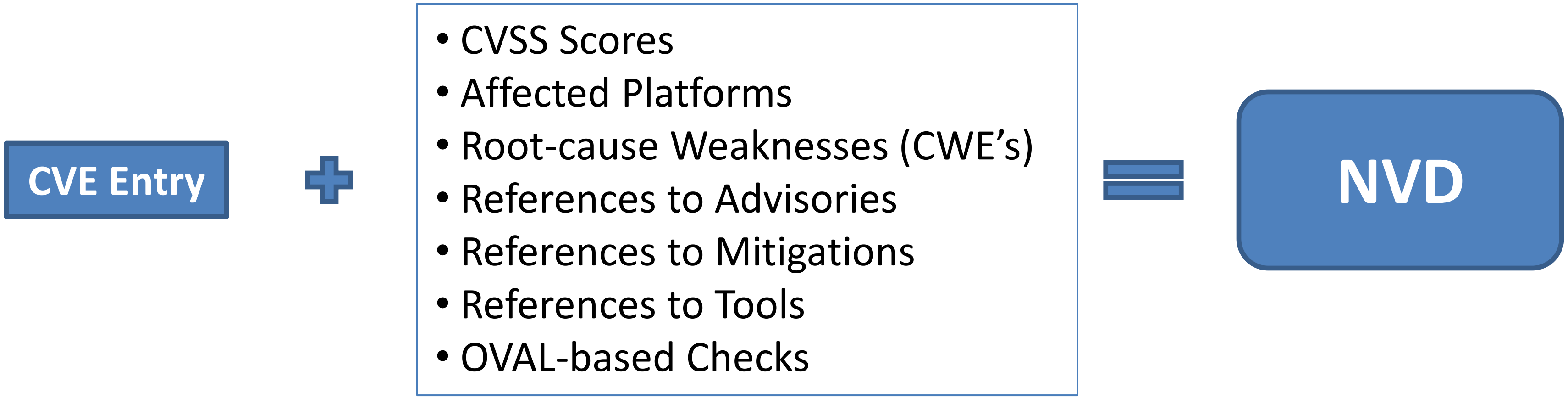
CVE entries feed into *NVD*

```
assert(CVE != Bug_Database);
```

Over 48,258 entries
(as of 28 Nov 2011)

Common Vulnerabilities and Exposures (CVE)

National Vulnerability Database (NVD)



CVE Entry

- CVSS Scores
- Affected Platforms
- Root-cause Weaknesses (CWE’s)
- References to Advisories
- References to Mitigations
- References to Tools
- OVAL-based Checks

NVD

U.S. government repository of standards-based vulnerability management data

website: **nvd.nist.gov**

Dictionary of software weakness *types*

- CWE ID
- Name
- Description
- Alternate Names
- Applicable Platforms
- Applicable Languages
- **Technical Impacts**
- Potential Mitigations
- **Observed Instances (CVE's)**
- **Related Attack Patterns (CAPEC's)**
- Examples

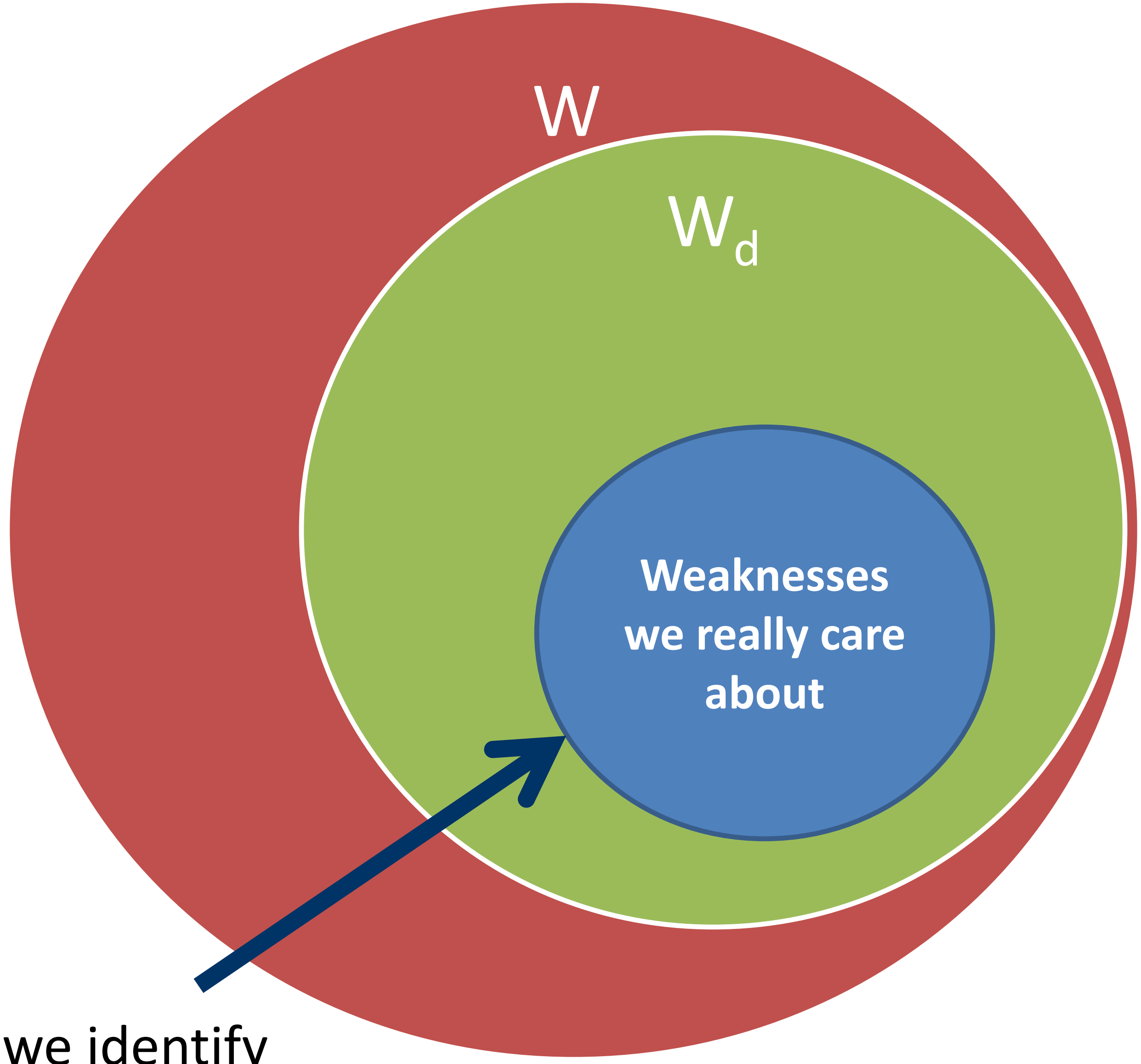
Plus much, much more

860+ entries in a tree-structure

Common Weakness Enumeration (CWE)

For the software we're responsible for

Notional



How do we identify these?

which weaknesses are most important?

Prioritizing weaknesses to be mitigated

OWASP
The Open Web Application Security Project
http://www.owasp.org

OWASP Top 10

CWE & SANS Institute
TOP 25
MOST DANGEROUS
SOFTWARE
ERRORS

CWE/SANS Top 25

Lists are a good start but they are designed to be broadly applicable

We would like a way to specify priorities based on business/mission risk

Common Weakness Risk Analysis Framework (CWRAF)

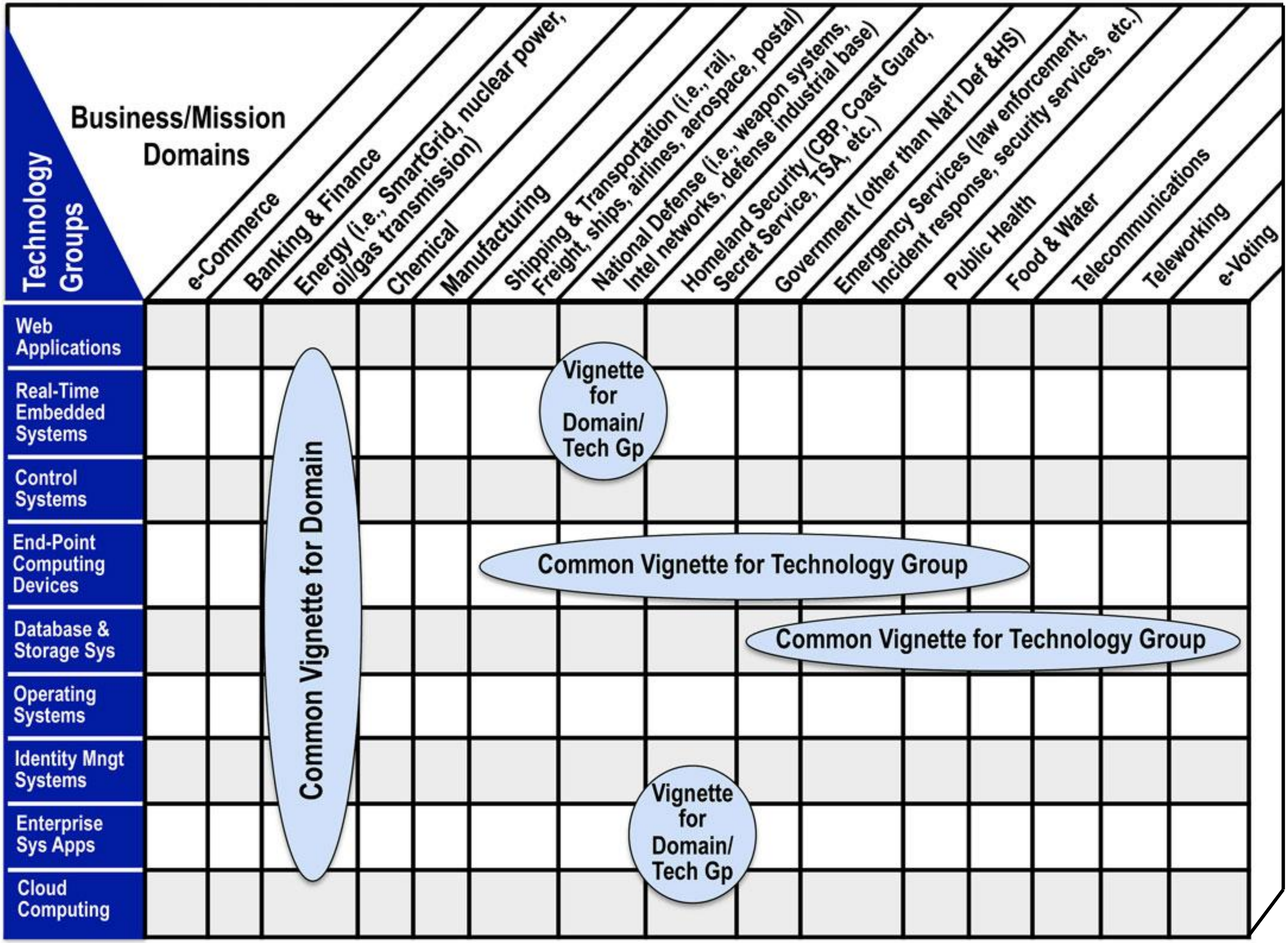
*How do I **identify** which of the 800+ CWE's are most important for my specific business domain, technologies and environment?*

Common Weakness Scoring System (CWSS)

*How do I **rank** the CWE's I care about according to my specific business domain, technologies and environment?*

How do I identify and score weaknesses important to my organization?

Leveraging Vignettes in Cyber Security Standardization for Key ICT Applications in various Domains



Business/Mission Domains

Technology Groups

e-Commerce

Banking & Finance

Energy (i.e., SmartGrid, nuclear power, oil/gas transmission)

Chemical

Manufacturing

Shipping & Transportation (i.e., rail, Freight, ships, airlines, aerospace, postal)

National Defense (i.e., weapon systems, Intel networks, defense industrial base)

Homeland Security (CBP, Coast Guard, Secret Service, TSA, etc.)

Government (other than Nat'l Def &HS)

Emergency Services (law enforcement, Incident response, security services, etc.)

Public Health

Food & Water

Telecommunications

Teleworking

e-Voting

Web Applications

Real-Time Embedded Systems

Control Systems

End-Point Computing Devices

Database & Storage Sys

Operating Systems

Identity Mngt Systems

Enterprise Sys Apps

Cloud Computing

Common Vignette for Domain

Vignette for Domain/ Tech Gp

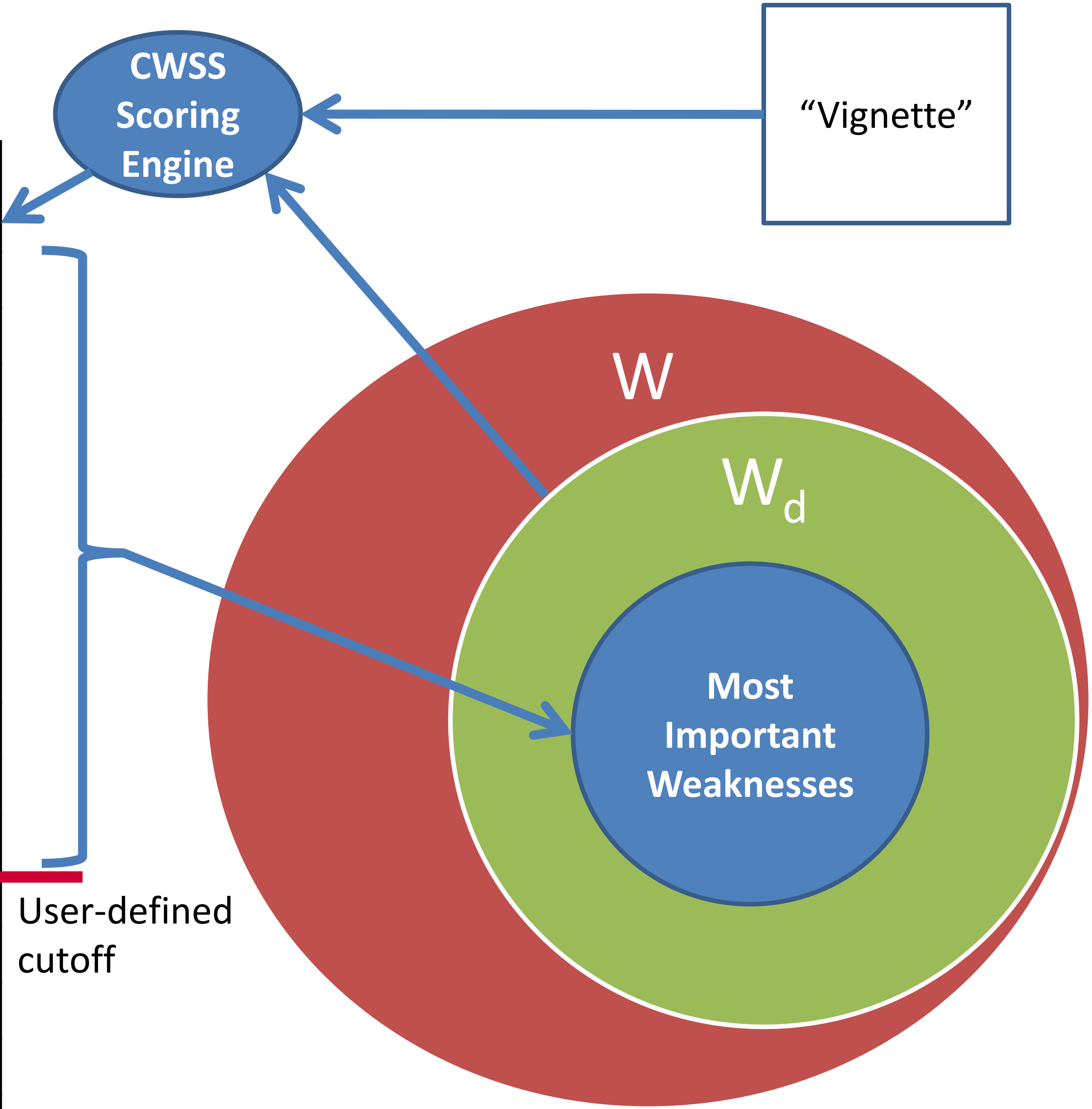
Common Vignette for Technology Group

Common Vignette for Technology Group

Vignette for Domain/ Tech Gp

Common Weakness Risk Assessment Framework uses Vignettes with Archetypes to identify top CWEs in respective Domain/Technology Groups

CWSS Score	CWE
97	CWE-79
95	CWE-78
94	CWE-22
94	CWE-434
94	CWE-798
93	CWE-120
93	CWE-250
92	CWE-770
91	CWE-829
91	CWE-190
91	CWE-494
90	CWE-134
90	CWE-772
90	CWE-476
90	CWE-131
…	



CWRAF/CWSS in a Nutshell

Common Weakness Risk Analysis Framework (CWRAF) and Common Weakness Scoring System (CWSS)

Organizations that have declared plans to work on CWRAF Vignettes and Technical Scorecards to help evolve CWRAF to meet their customer's and the community's needs for a scoring system for software errors.

Trustwave® SpiderLabs®

DTCC®

CISQ

EC-Council

OWASP
The Open Web Application Security Project

SAIC®

Common Weakness Risk Analysis Framework (CWRAF) and Common Weakness Scoring System (CWSS)

Organizations that have declared plans to support CWSS in their future offerings and are working to help evolve CWSS to meet their customer's and the community's needs for a scoring system for software errors.

CENZIC

GRAMMATECH

Klocwork

VERACODE

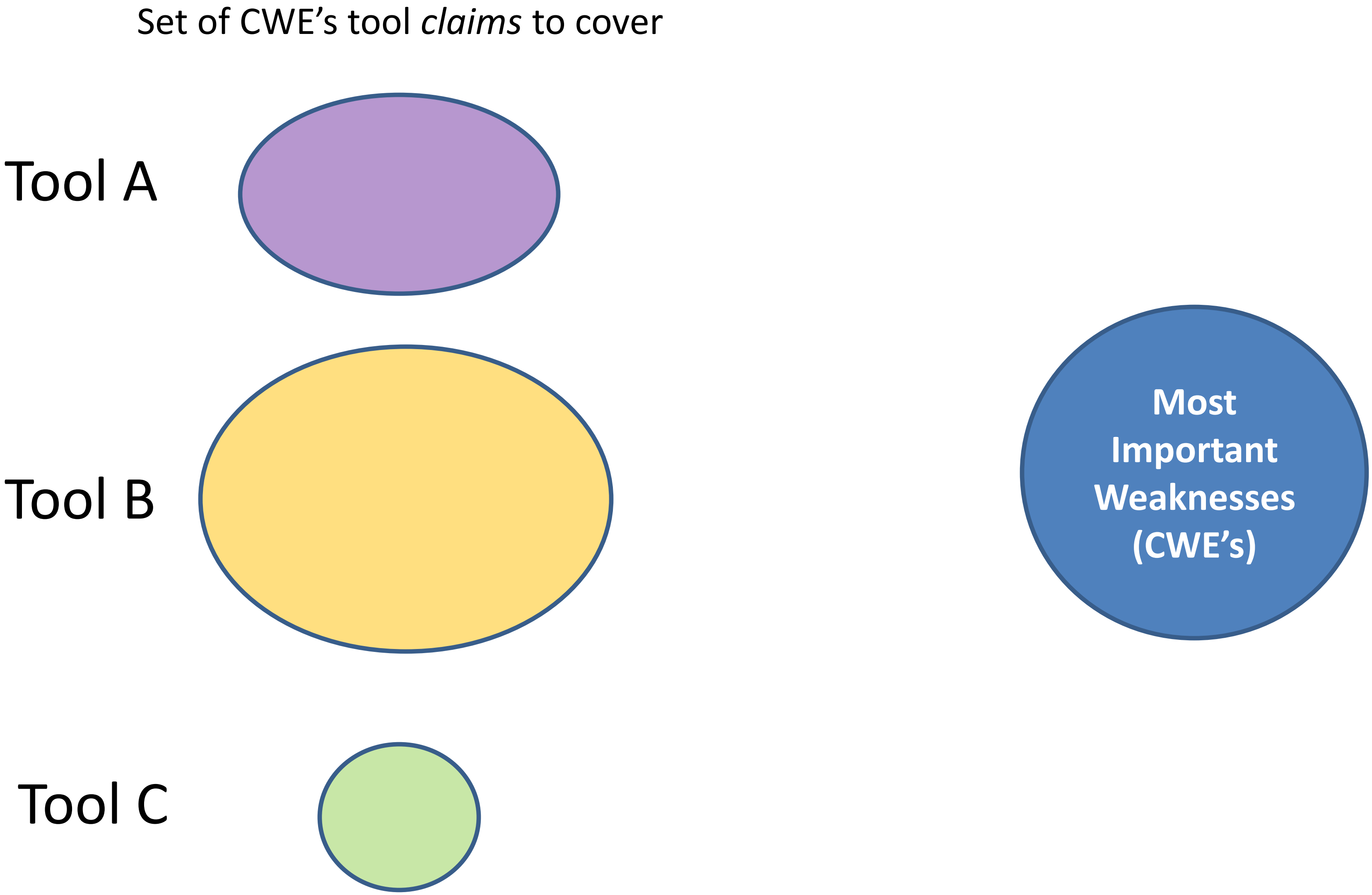
SANS

FORTIFY SOFTWARE

CAST

coverity®

CWE Coverage Claims Representation (CCR)



Set of CWE’s tool *claims* to cover

Tool A

Tool B

Tool C

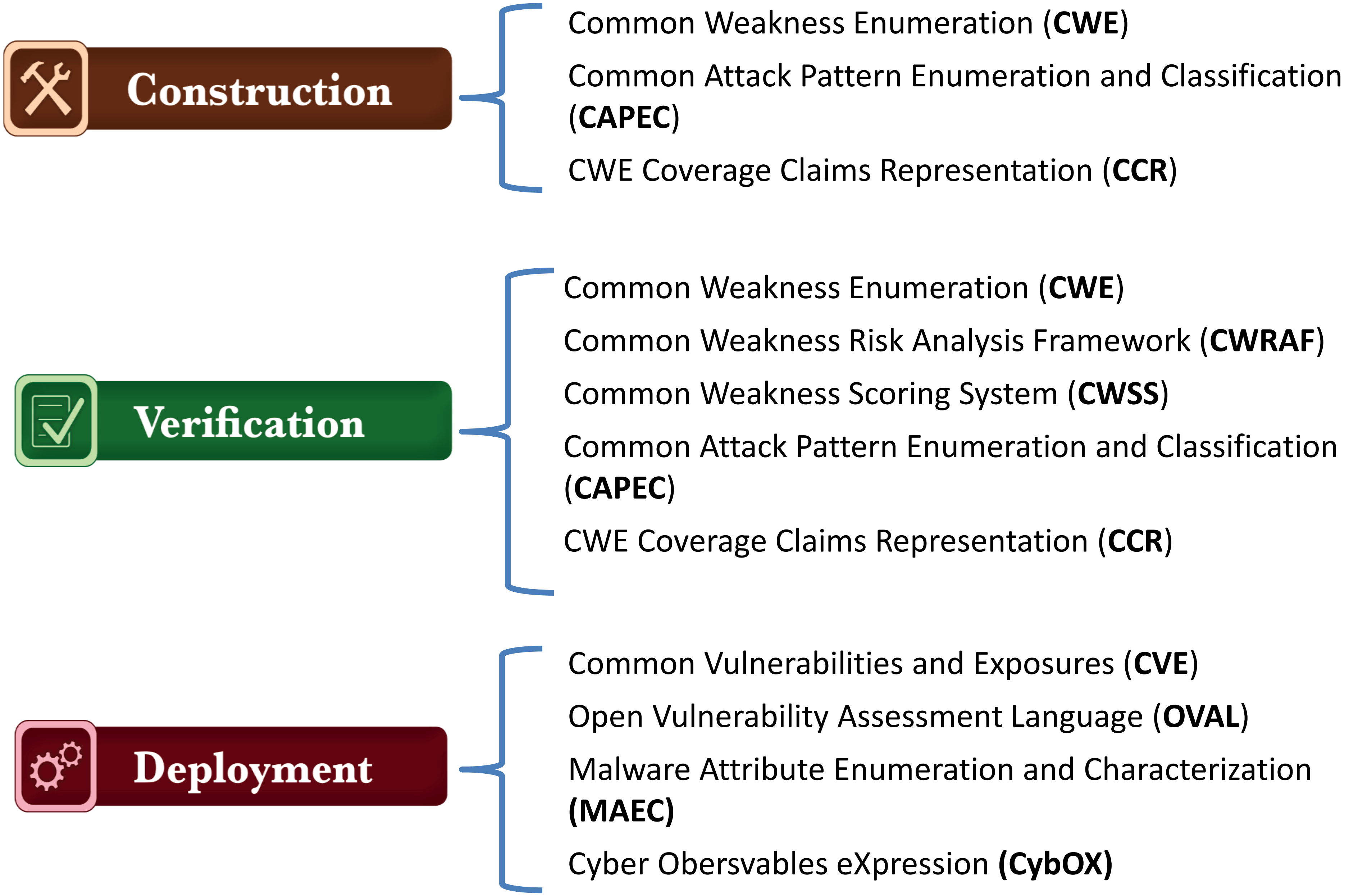
Most Important Weaknesses (CWE’s)

Which static analysis tools find the CWE’s I care about?

CWRAF/CWSS Provides Risk Prioritization for CWE throughout Software Life Cycle

- Enables education and training to provide specific practices for eliminating software fault patterns;
- Enables developers to mitigate top risks attributable to exploitable software;
- Enables testing organizations to use suite of test tools & methods (with CWE Coverage Claims Representation) that cover applicable concerns;
- Enables users and operation organizations to deploy and use software that is more resilient and secure;
- Enables procurement organizations to specify software security expectations through acquisition of software, hosted applications and services.

automation can help…



Construction

- Common Weakness Enumeration (**CWE**)
- Common Attack Pattern Enumeration and Classification (**CAPEC**)
- CWE Coverage Claims Representation (**CCR**)

Verification

- Common Weakness Enumeration (**CWE**)
- Common Weakness Risk Analysis Framework (**CWRAF**)
- Common Weakness Scoring System (**CWSS**)
- Common Attack Pattern Enumeration and Classification (**CAPEC**)
- CWE Coverage Claims Representation (**CCR**)

Deployment

- Common Vulnerabilities and Exposures (**CVE**)
- Open Vulnerability Assessment Language (**OVAL**)
- Malware Attribute Enumeration and Characterization **(MAEC)**
- Cyber Obersvables eXpression **(CybOX)**

Software Supply Chain Practices

- How can we mitigate or reduce residual security debt being passed or inherited by using organizations?
  - What constitutes residual security debt in the supply chain?
  - How would considerations for security processes contribute to determining capabilities of supply chain actors?
  - How do security-enhanced security practices measurably contribute to reductions in exploitable weaknesses in software products, systems, and services?
  - Can code behavior be well-understood not to possess malicious constructs or exploitable weaknesses

**SwA Working Group sessions 28 Nov – 2 Dec 2011
at MITRE, McLean, VA**

SOFTWARE ASSURANCE FORUM

"Building Security In"

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

Joe Jarzombek, PMP, CSSLP
Director for Software Assurance
National Cyber Security Division
Department of Homeland Security
Joe.Jarzombek@dhs.gov
(703) 235-3673
LinkedIn SwA Mega-Community

SOFTWARE ASSURANCE FORUM

Homeland Security

Commerce

National Defense

BUILDING SECURITY IN

Public/Private Collaboration Efforts for Software Supply Chain Risk Management

Next SwA Working Sessions 28 Nov – 2 Dec 2011 at MITRE, McLean, VA

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.

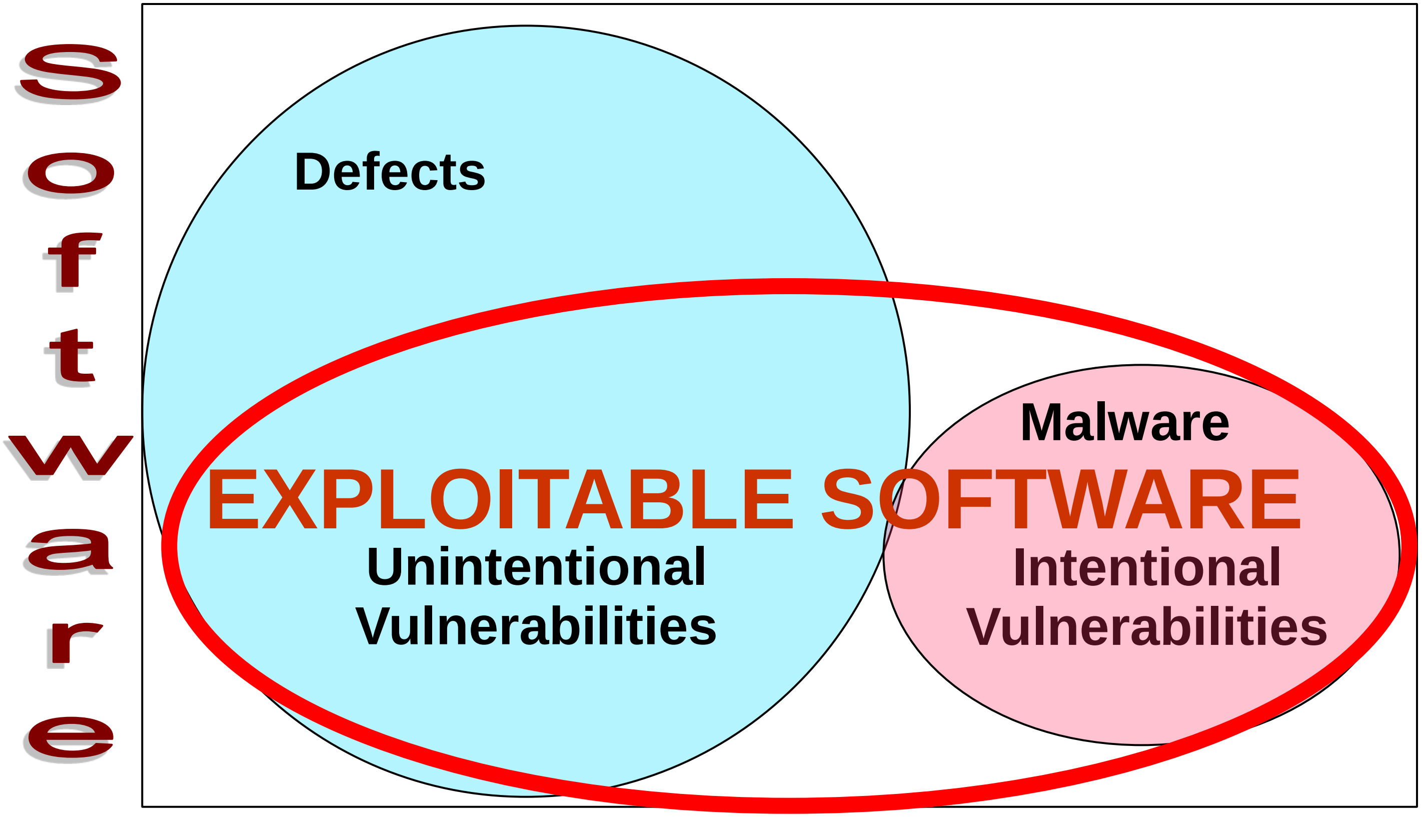
- National Defense Authorization Act (NDAA) -- which included a focus on software security (in Section 932, Strategy on Computer Software Assurance) -- serves as first cybersecurity law of 2011 and requires the U.S. Dept of Defense to develop a strategy for ensuring the security of software applications.

- 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 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

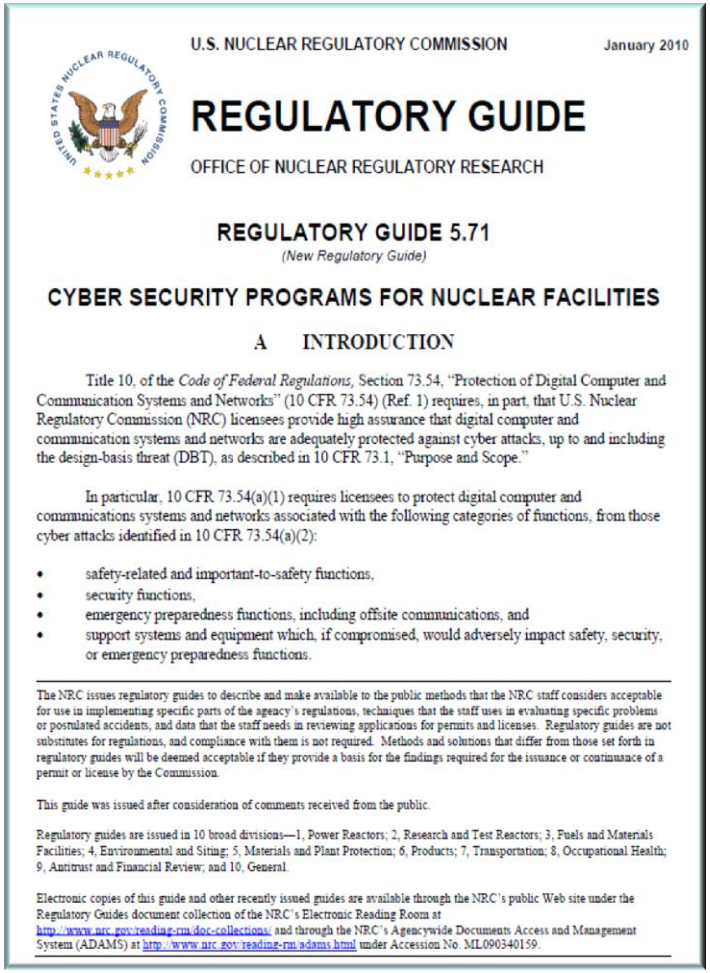
*Intentional vulnerabilities: spyware & malicious logic deliberately imbedded (might not be considered defects)

Homeland Security

Note: Chart is not to scale – notional representation -- for discussions

NRC Regulatory Guidance on Cyber Security

- NRC Regulatory Guide 5.71, “Cyber Security Programs for Nuclear Facilities,” Section C.12 in Appendix C, “System and Service Acquisition”
  - Directly relates to current NRC guidance on cyber security in the supply chain and SDLC of an ICS regulated by the agency.
  - Section C.12.2 “Supply Chain Protection” control drill down to the vendor level with requirements accountability for the RG 5.71 control baseline (Appendices B&C).
  - Section C.12.3 “Trustworthiness” requires developers employ software quality and validation methods to minimize flawed or malformed software; requires all tools to undergo commercial certification process
  - Section C.12.5 “Developer Security Testing”



U.S. NUCLEAR REGULATORY COMMISSION January 2010

REGULATORY GUIDE

OFFICE OF NUCLEAR REGULATORY RESEARCH

REGULATORY GUIDE 5.71

(New Regulatory Guide)

CYBER SECURITY PROGRAMS FOR NUCLEAR FACILITIES

A INTRODUCTION

Title 10, of the *Code of Federal Regulations*, Section 73.54, “Protection of Digital Computer and Communication Systems and Networks” (10 CFR 73.54) (Ref. 1) requires, in part, that U.S. Nuclear Regulatory Commission (NRC) licensees provide high assurance that digital computer and communication systems and networks are adequately protected against cyber attacks, up to and including the design-basis threat (DBT), as described in 10 CFR 73.1, “Purpose and Scope.”

In particular, 10 CFR 73.54(a)(1) requires licensees to protect digital computer and communications systems and networks associated with the following categories of functions, from those cyber attacks identified in 10 CFR 73.54(a)(2):

- safety-related and important-to-safety functions,
- security functions,
- emergency preparedness functions, including offsite communications, and
- support systems and equipment which, if compromised, would adversely impact safety, security, or emergency preparedness functions.

The NRC issues regulatory guides to describe and make available to the public methods that the NRC staff considers acceptable for use in implementing specific parts of the agency’s regulations, techniques that the staff uses in evaluating specific problems or postulated accidents, and data that the staff needs in reviewing applications for permits and licenses. Regulatory guides are not substitutes for regulations, and compliance with them is not required. Methods and solutions that differ from those set forth in regulatory guides will be deemed acceptable if they provide a basis for the findings required for the issuance or continuance of a permit or license by the Commission.

This guide was issued after consideration of comments received from the public.

Regulatory guides are issued in 10 broad divisions—1, Power Reactors; 2, Research and Test Reactors; 3, Fuels and Materials Facilities; 4, Environmental and Siting; 5, Materials and Plant Protection; 6, Products; 7, Transportation; 8, Occupational Health; 9, Antitrust and Financial Review; and 10, General.

Electronic copies of this guide and other recently issued guides are available through the NRC’s public Web site under the Regulatory Guides document collection of the NRC’s Electronic Reading Room at http://www.nrc.gov/reading-rm/doc-collections/ and through the NRC’s Agencywide Documents Access and Management System (ADAMS) at http://www.nrc.gov/reading-rm/adams.html under Accession No. ML090340159.

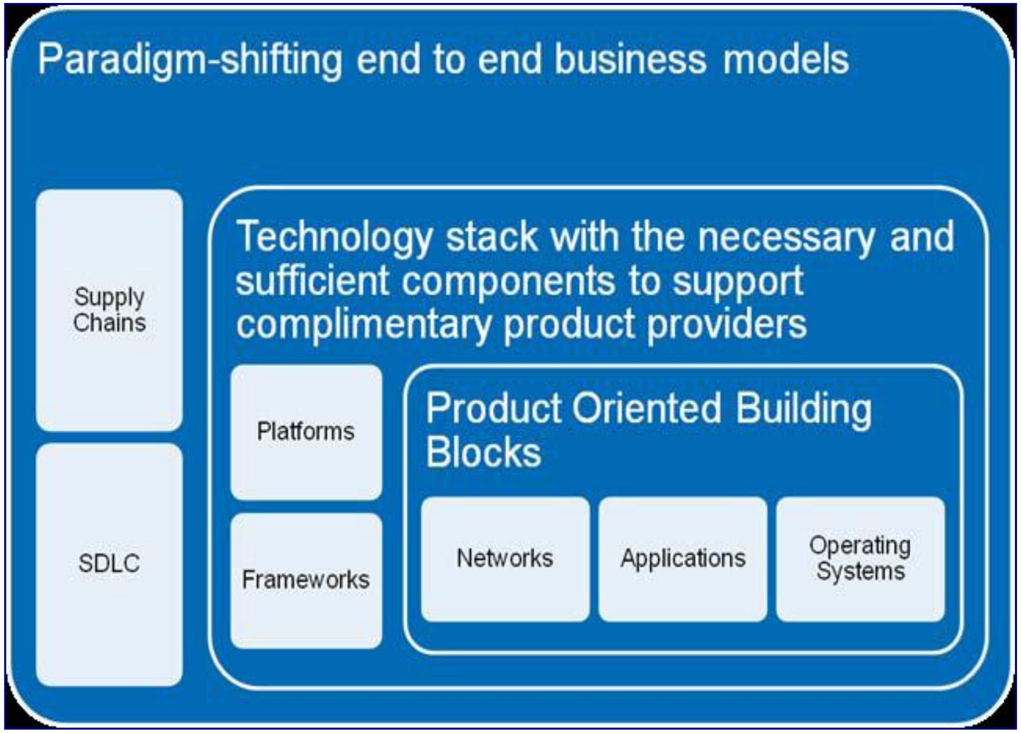
See NRC Regulatory Guidance on Cyber Security http://pbadupws.nrc.gov/docs/ML0903/ML090340159.pdf

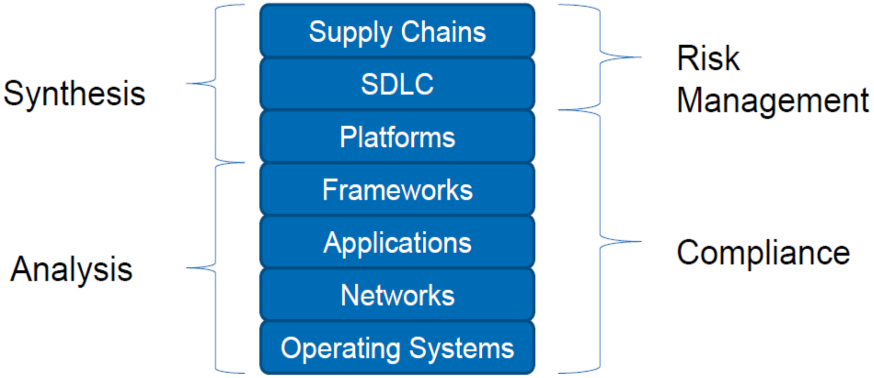
IT/software security risk landscape is a convergence between "defense in depth" and "defense in breadth"

Enterprise Risk Management and Governance are security motivators

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

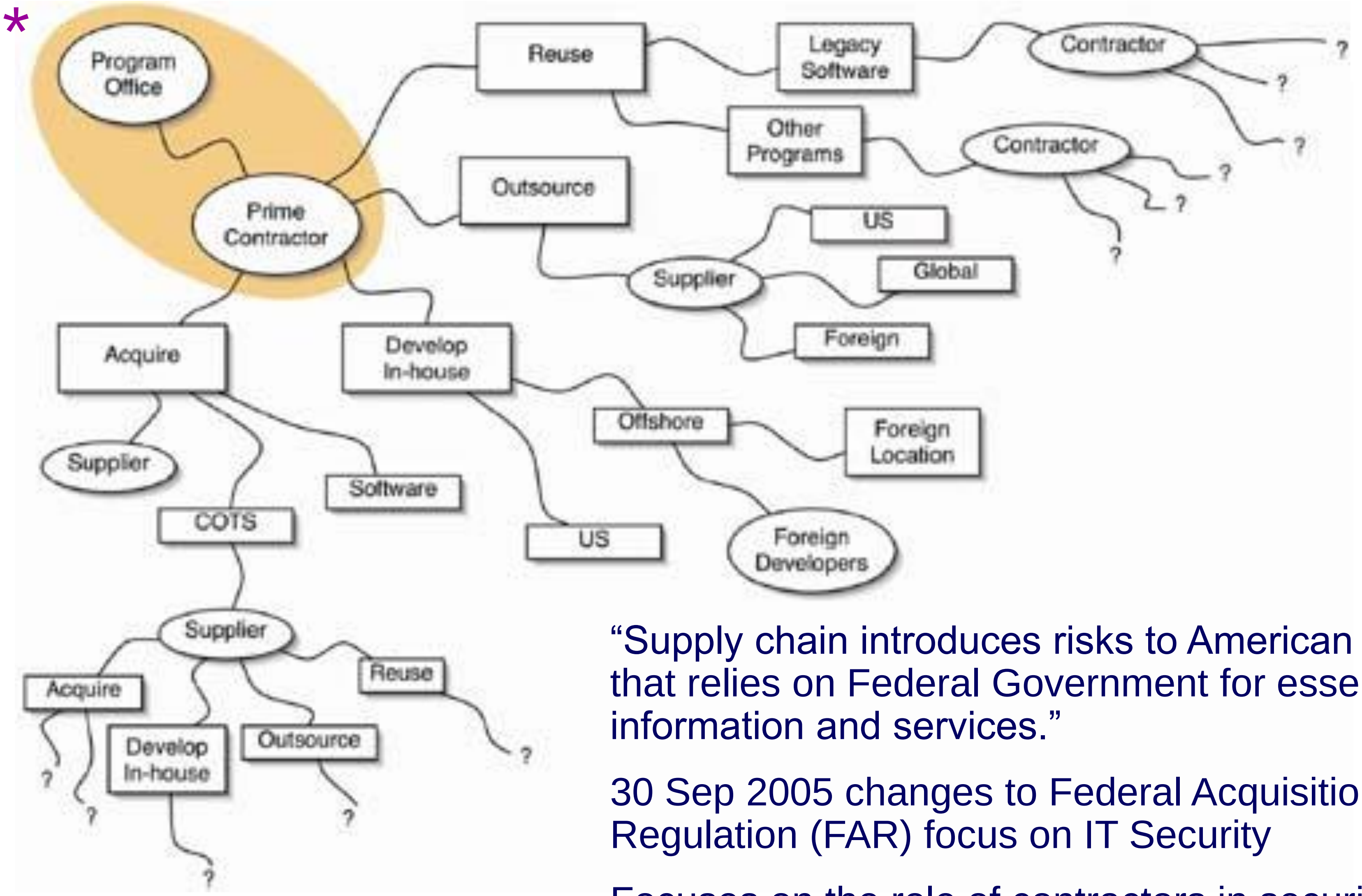
> "In the digital age, sovereignty is demarcated not by territorial frontiers but by supply chains."
>
> – Dan Geer, CISO In-Q-Tel


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"

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

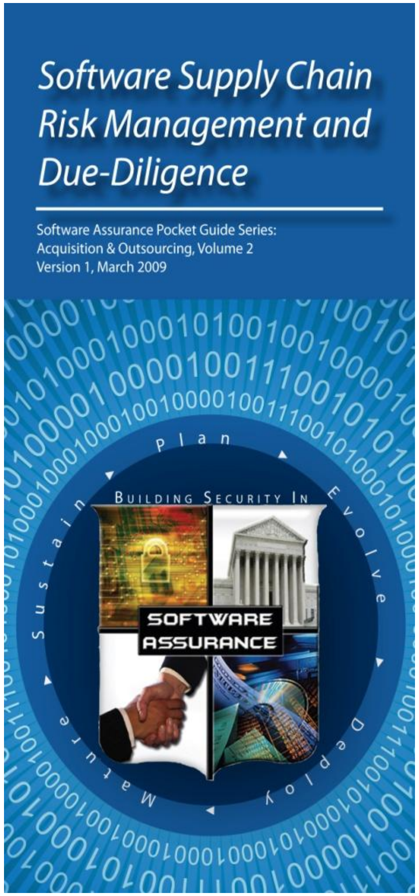
- Integrating Security into the Software Development Life Cycle
- ⇨ Key Practices for Mitigating the Most Egregious Exploitable Software Weaknesses
- ⇨ Software Security Testing
- ⇨ Requirements and Analysis for Secure Software
- ⇨ Architecture and Design Considerations for Secure Software
- ⇨ Secure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

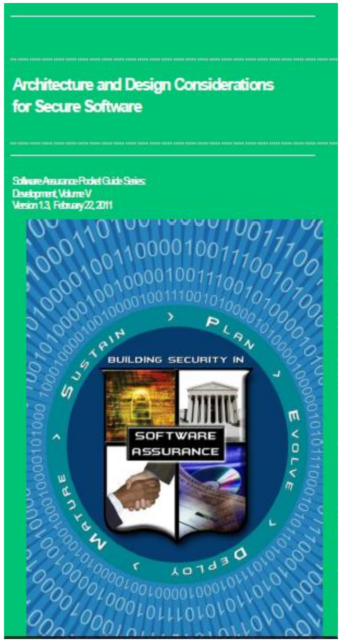
SwA Pocket Guides and SwA-related documents are collaboratively developed with peer review; they are subject to update and are freely available for download via the DHS Software Assurance Community Resources and Information Clearinghouse at https://buildsecurityin.us-cert.gov/swa (see SwA Resources)

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

Apply Key Principles & Practices

Table 1- Adapted from "Enhancing the Development Life Cycle to Produce Secure Software"

General Principle	Key Practices	Principle Design Conformance
Minimize the number of high-consequence targets	Principle of least privilege	Minimizes the number of actors in the system granted high levels of privilege, and the amount of time any actor holds onto its privileges.
	Separation of privileges, duties, and roles	Ensures that no single entity (human or software) should have all the privileges required to modify, delete, or destroy the system, components and resources.
	Separation of domains	This practice makes separation of roles and privileges easier to implement.
Don't expose vulnerable or high-consequence components	Keep program data, executables, and configuration data separated	Reduces the likelihood that an attacker who gains access to program data will easily locate and gain access to program executables or control/configuration data.
	Segregate trusted entities from untrusted entities	Reduces the exposure of the software's high-consequence functions from its high-risk functions, which can be susceptible to attacks.
	Minimize the number of entry and exit points	Reduces the attack surface.
	Assume environment data is not trustworthy	Reduces the exposure of the software to potentially malicious execution environment components or attacker-intercepted and modified environment data.
	Use only trusted interfaces to environment resources	This practice reduces the exposure of the data passed between the software and its environment.
Deny attackers the means to compromise	Simplify the design	This practice minimizes the number of attacker-exploitable vulnerabilities and weaknesses in the system.
	Hold ***all*** actors accountable	This practice ensures that all attacker actions are observed and recorded, contributing to the ability to recognize and isolate/block the source of attack patterns.
	Timing, synchronization, and sequencing should be simplified to avoid issues.	Modeling and documenting timing, synchronization, and sequencing issues will reduce the likelihood of race conditions, order dependencies, synchronization problems, and deadlocks.
	Make secure states easy to enter and vulnerable states difficult to enter	This practice reduces the likelihood that the software will be allowed to inadvertently enter a vulnerable state.
	Design for controllability	This practice makes it easier to detect attack paths, and disengage the software from its interactions with attackers. Caution should be taken when using this approach since it can open a whole range of new attack vectors.
	Design for secure failure	Reduces the likelihood that a failure in the software will leave it vulnerable to attack.

Conduct Design Reviews

Verification activities during the design phase includes:

- Structured inspections, conducted on parts or views of the high-level design throughout the design phase;
- Independent verification and validation (IV&V) reviews;
- A preliminary design review conducted at the end of the architecture design phase and before entry into the detailed design phase; and
- A critical design review added at designated points in the software development lifecycle.

Design Verification–The design should be verified considering the following criteria:

- The design is correct and consistent with and traceable to requirements;
- The design implements the proper sequence of events, inputs, outputs, interfaces, logic flow, allocation of timing and sizing budgets, error definition, isolation, and recovery;
- The selected design can be derived from requirements; and
- The design implements safety, security, and other critical requirements correctly as shown by suitably rigorous methods.

Leverage Attack Patterns

CAPEC – Common Attack Pattern Enumeration and Classification: *capec.mitre.org*

Attack Patterns:

- Blueprint for creating a specific type of attack
- Abstracted common attack approaches from the set of known exploits
- Capture the attacker's perspective to aid software developers, acquirers and operators in improving the assurance profile of their software

Use Attack Patterns to:

- Guide definition of appropriate policies
- Guide creation of appropriate security requirements (positive and negative)
- Provide context for architectural risk analysis
- Guide risk-driven secure code review
- Provide context for appropriate security testing
- Provide a bridge between secure development and secure operations

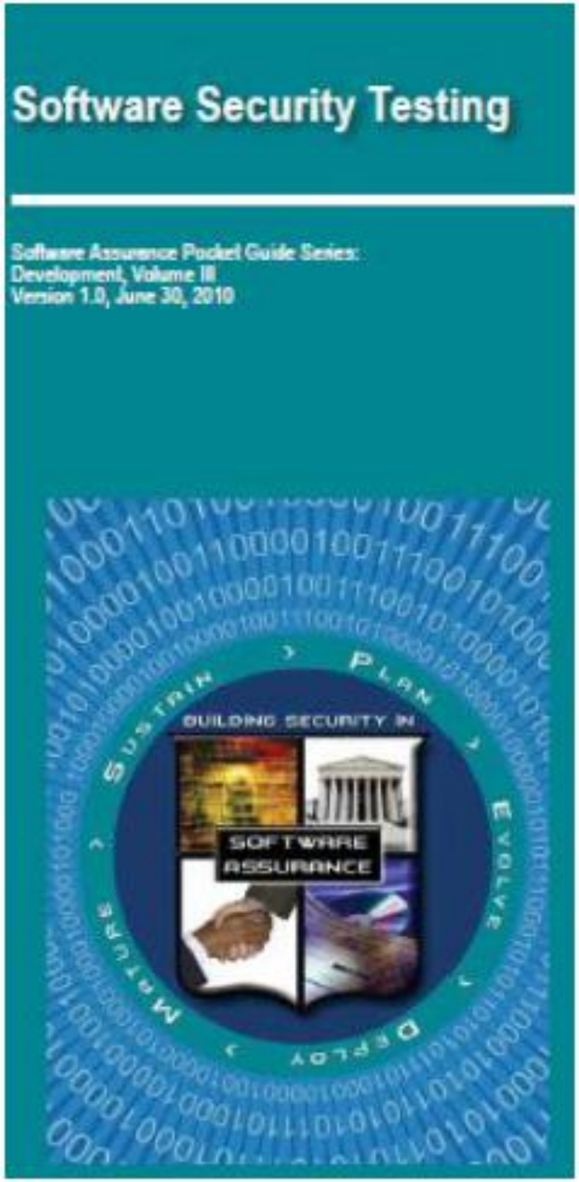
Ask Questions

Ask the designers/developers of the software:

- How does software assure that state information has not been tampered with?
- How does the software prevent attackers from repeatedly triggering a synchronized critical section?
- Are misuse cases utilized during the design process? If so, what modeling techniques are used to define and handle misuse cases?
- How does the software perform access control checks?
- How does the application ensure that only authorized personnel are accessing sensitive information?
- How does the software assure that data transmitted over a communications channel cannot be intercepted and/or modified?
- Are formal methods incorporated into the architectural design to improve software security? If so, what methods are used and why?
- How does the server-side security check for bypasses in the client-side validation? What techniques are used to prevent this?
- How does the server ensure that the Session ID has not been stolen or manipulated?
- How does team prevent weaknesses in initialization of variables and objects?

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

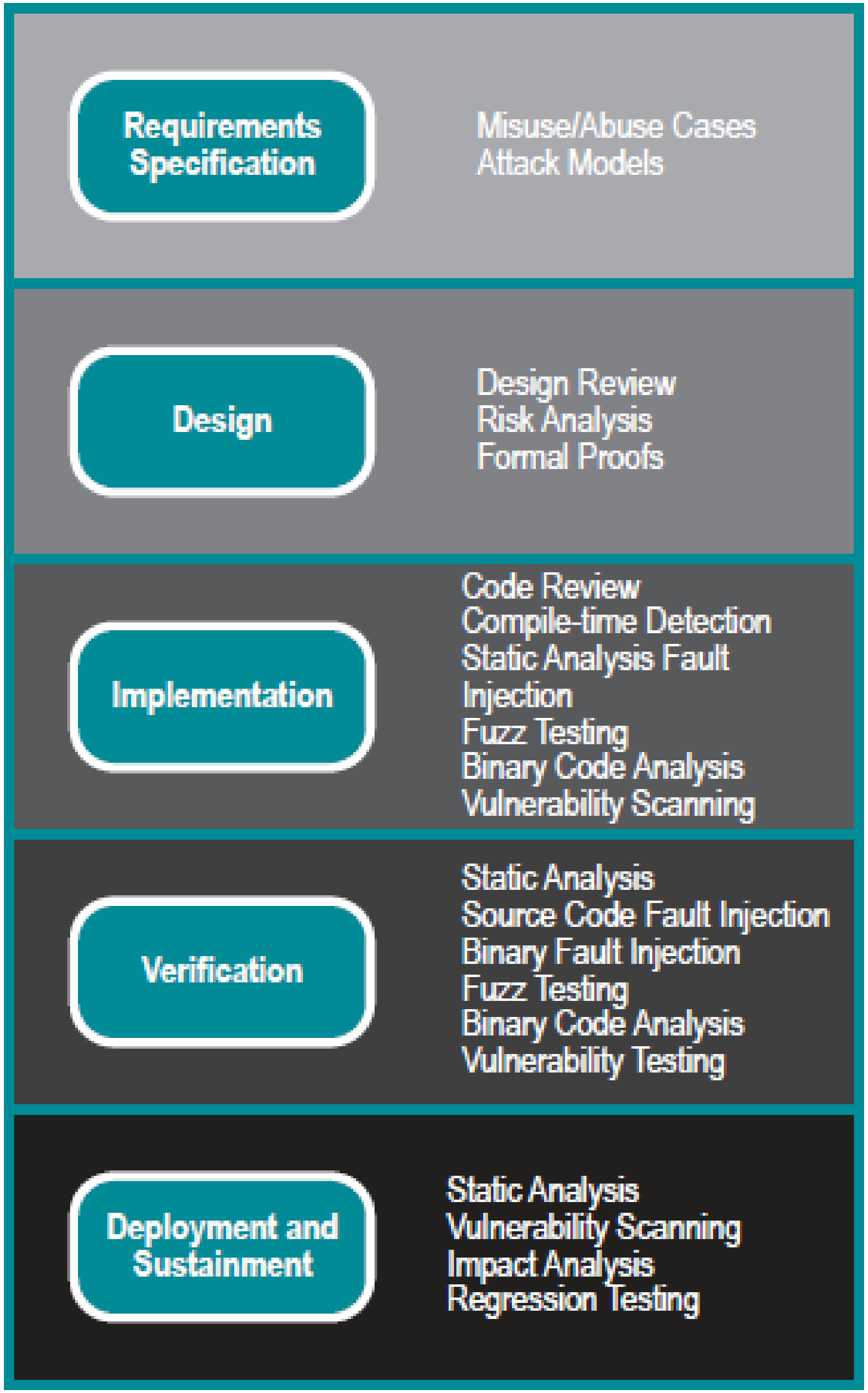
Software Assurance Pocket Guide Series:
Development, Volume III
Version 1.0, June 30, 2010

Software Security Test Techniques throughout the SDLC

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

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

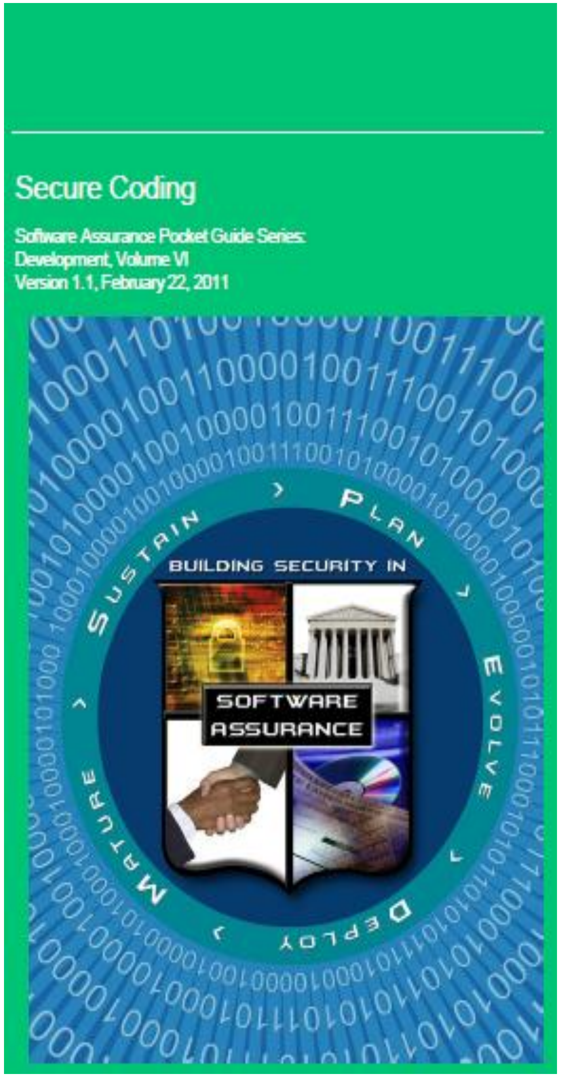
Homeland
Security



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

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

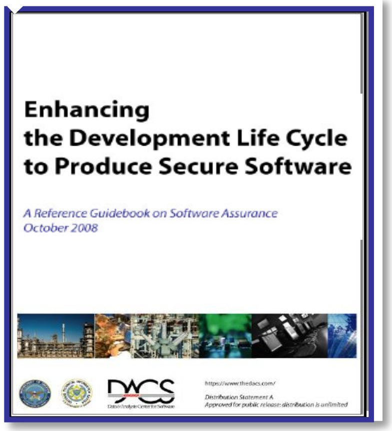
Software Assurance Pocket Guide Series:
Development, Volume VI
Version 1.1, February 22, 2011

Are compiler warnings disabled in code being delivered?

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

Many SwA Resources Focus On Development

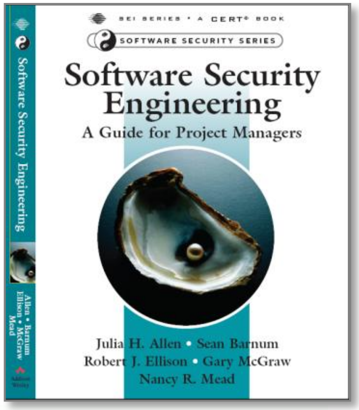

Enhancing the Development Life Cycle to Produce Secure Software

A Reference Guidebook on Software Assurance October 2008

DACS

https://www.thedacs.com/

Distribution Statement A Approved for public release: distribution is unlimited

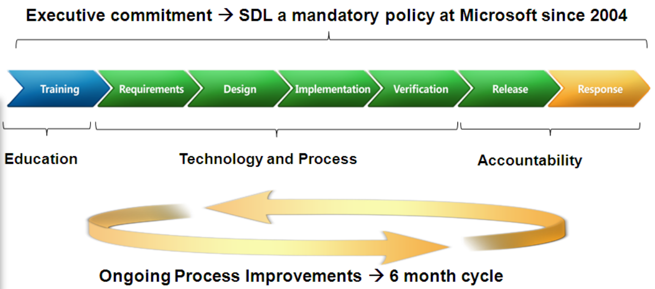

SEI SERIES • A CERT® BOOK

SOFTWARE SECURITY SERIES

Software Security Engineering

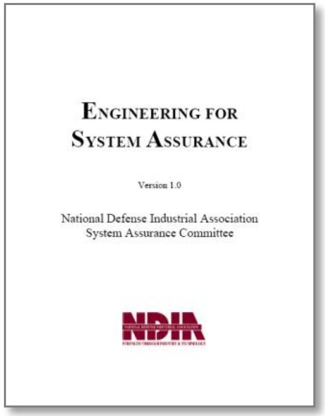
A Guide for Project Managers

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

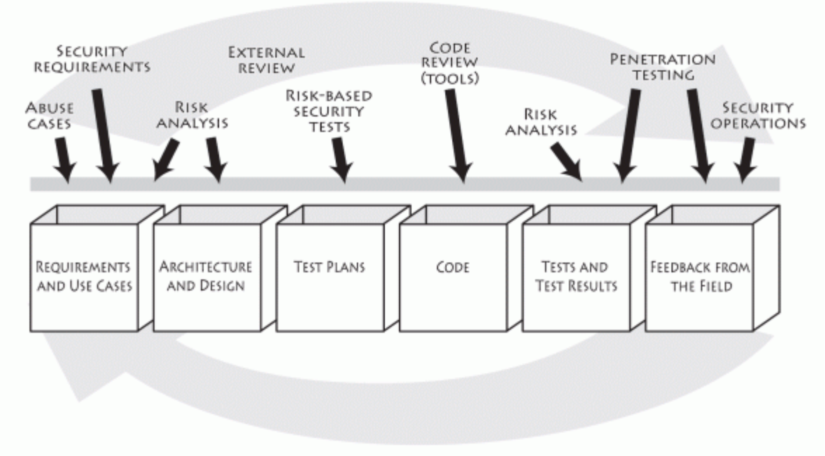
http://www.microsoft.com/sdl


ENGINEERING FOR SYSTEM ASSURANCE

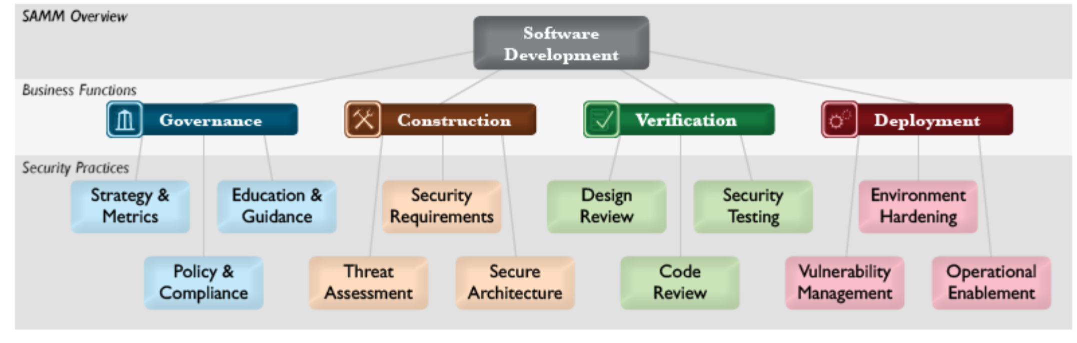
Version 1.0

National Defense Industrial Association
System Assurance Committee

NDIA



Assurance for CMMI ®



Ask Questions

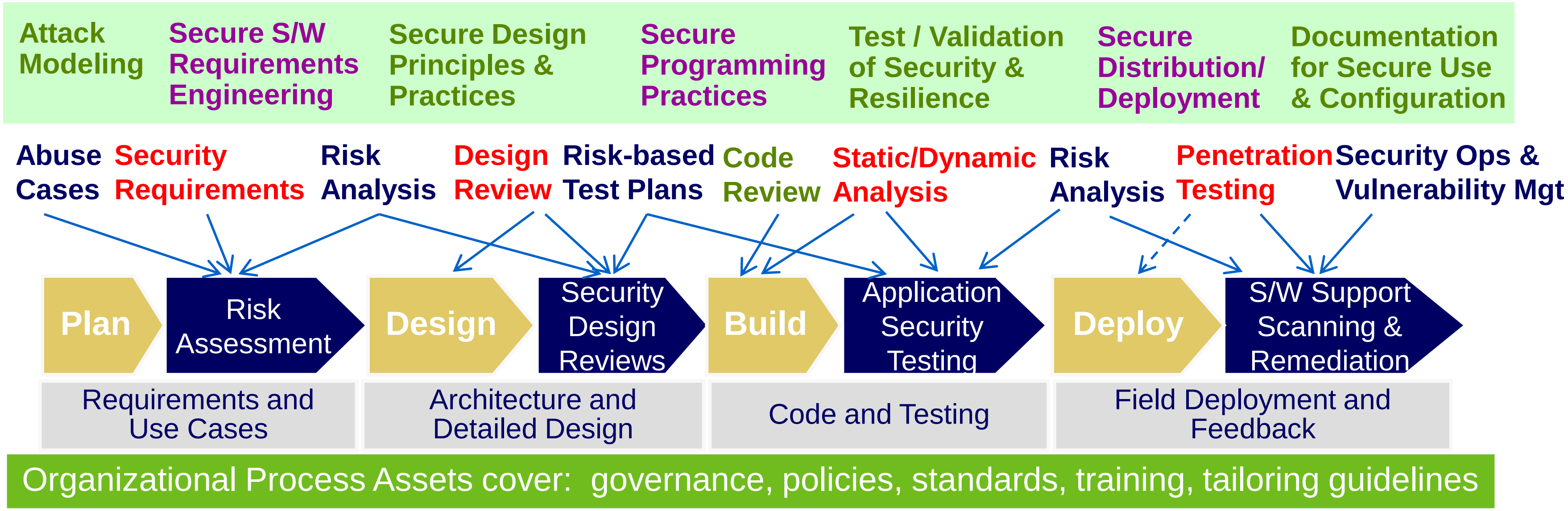
Ask the designers/developers of the software:

- What input and output protection mechanisms are implemented within the application?
- What measures are taken to ensure that users only have access to that which they are allowed to view?
- How are user passwords stored?
- How are connection strings stored?
- What cryptographic functions are implemented within the application, and what considerations went into selecting this over others?
- What tasks are included in each phase of the software development life cycle to identify and address security concerns?
- Who is involved in the code review process and how is it reviewed?
- If any, which third-party libraries are used, and how is it checked to ensure it does not contain malicious code or vulnerabilities?
- What type of auditing is done to keep track of changes made to database records?
- What language is used and why?
- What are the strengths and weaknesses of the language?
- Describe the security aspects of the development process. Provide your development policy and application guides.

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



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

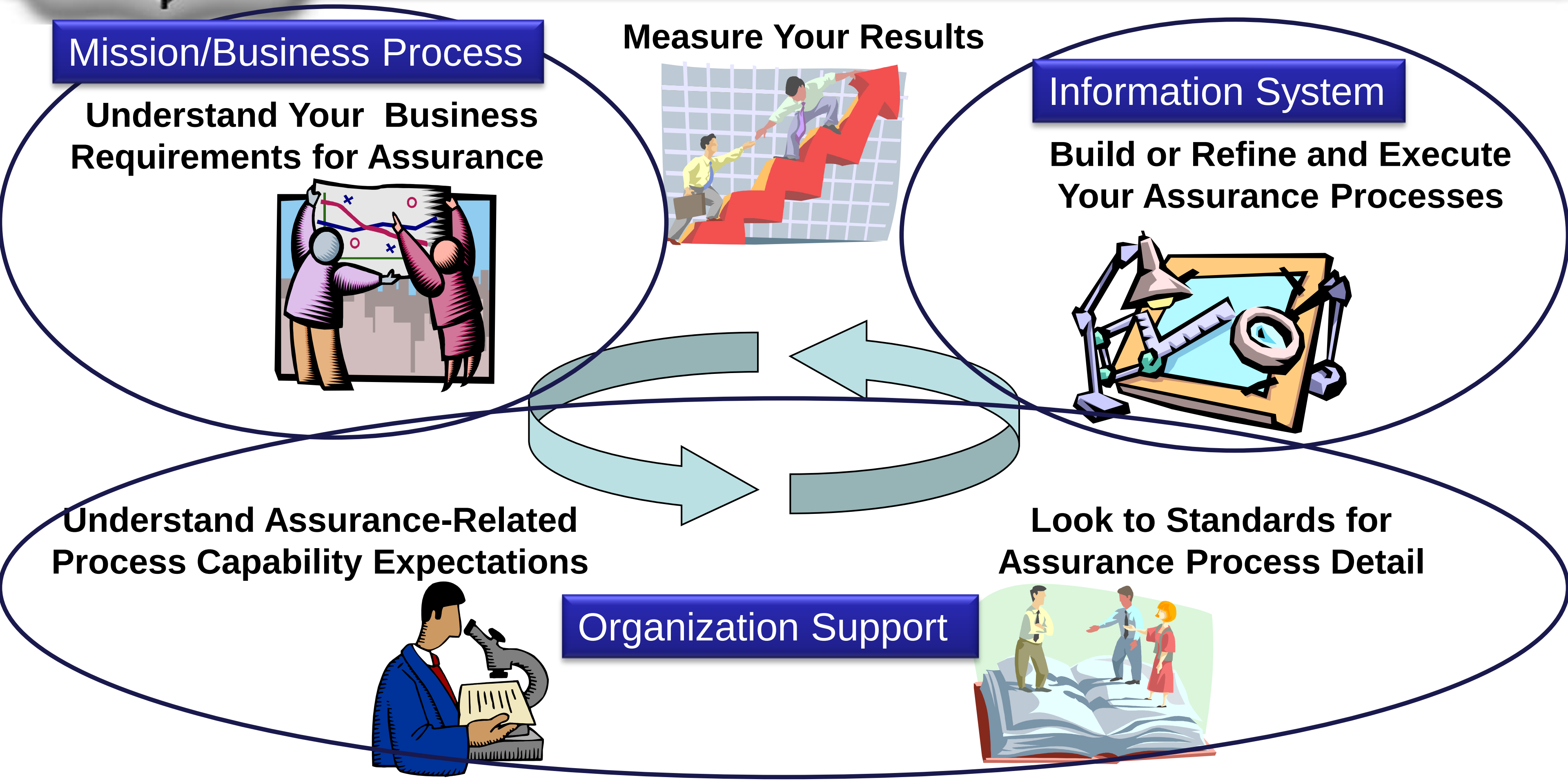
Requirements and Use Cases | Architecture and Detailed Design | Code and Testing | Field Deployment and Feedback

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.

* 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)

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

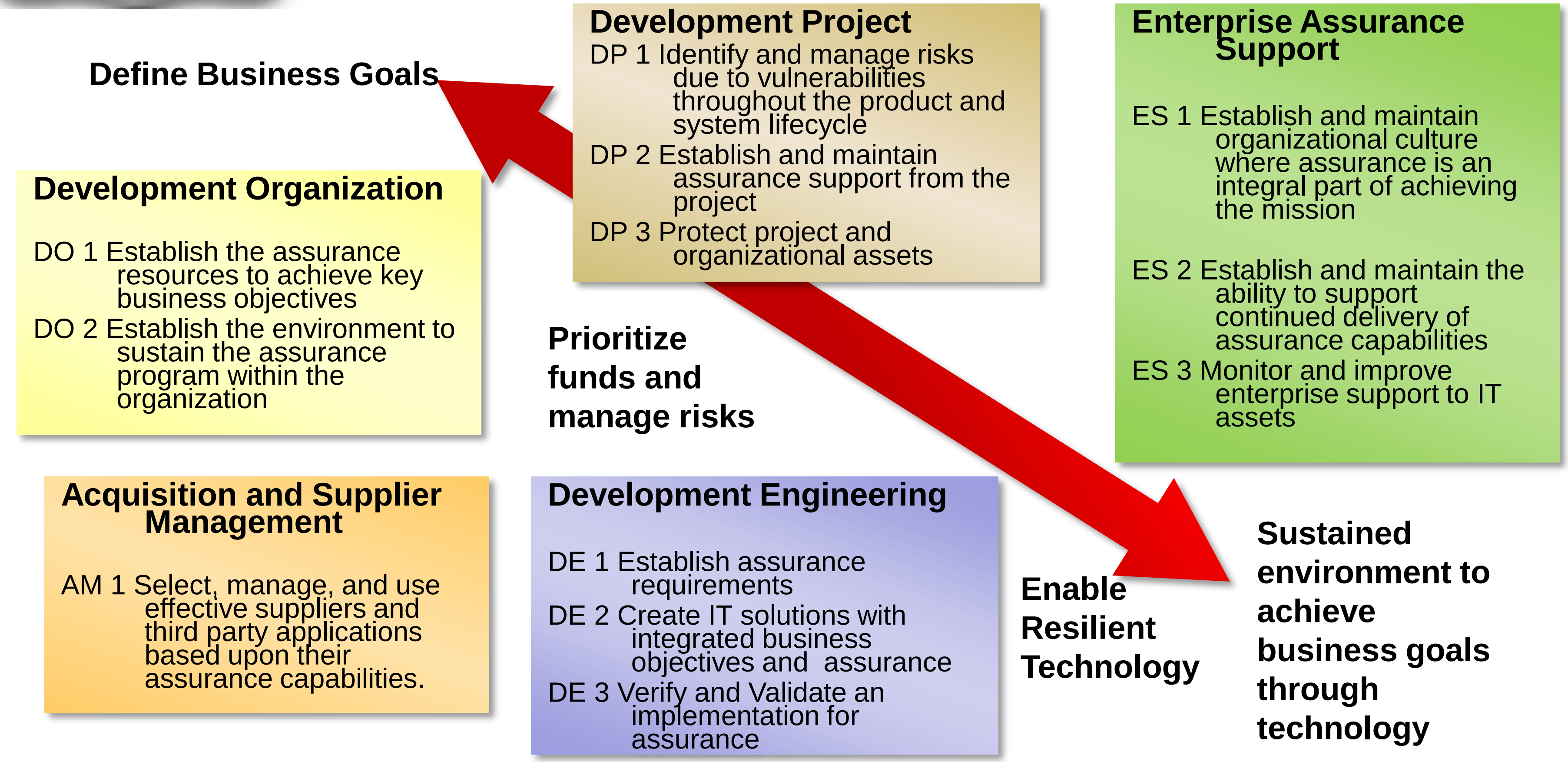
Understand Assurance-Related Process Capability Expectations

Look to Standards for Assurance Process Detail

Organization Support

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

The Assurance PRM Is A Holistic Framework



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

The DHS SwA Processes and Practices Working Group has synthesized the contributions of leading government and industry experts into a set of high-level goals and supporting practices (an evolution of the SwA community's Assurance Process Reference Model)

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

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

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

- NIST IR 7622: DRAFT Piloting Supply Chain Risk Management Practices for Federal Information Systems
- NDIA System Assurance Guidebook
- Microsoft Security Development Lifecycle
- SAFECode

Process Reference Model for Assurance – Goals and Practices September 2010

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

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

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

It can be used by acquirers, suppliers and integrators as a to tool to discuss areas of strength and weakness

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

SwA Community Assurance Process Reference Model – Self Assessment

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

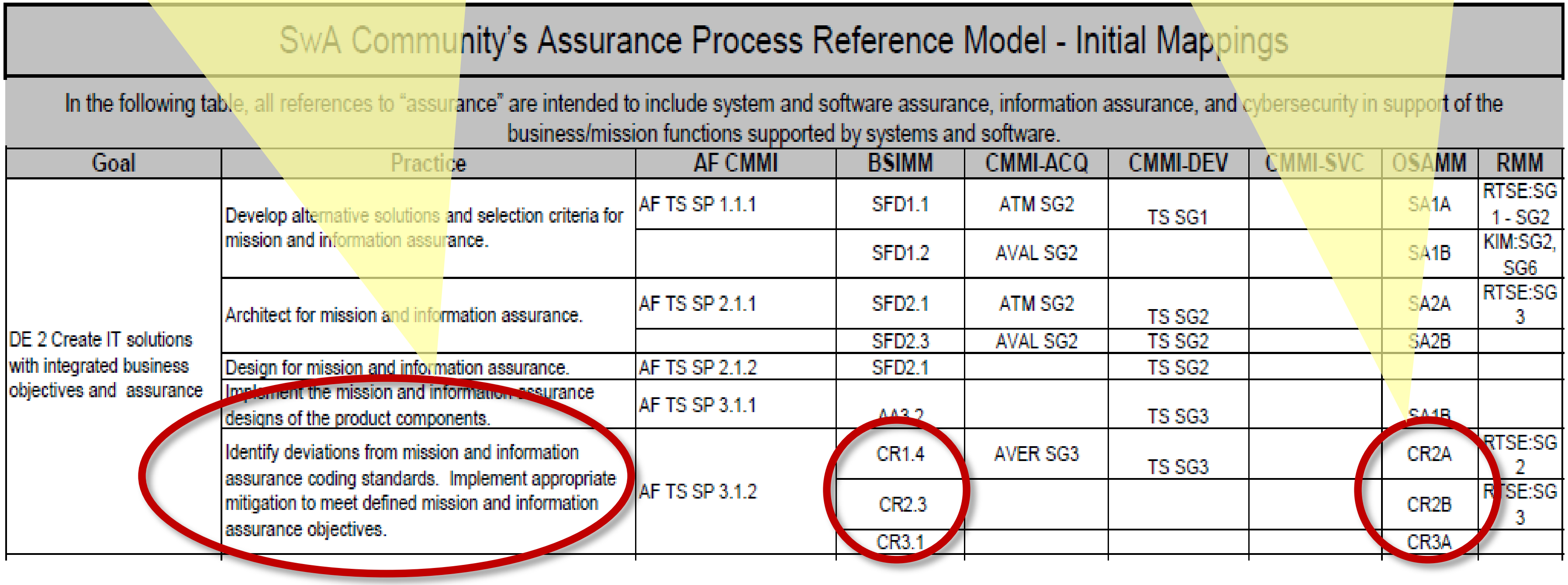
Goal	Practice	Practice Implementation Level	Notes
Development – Engineering			
DE 1 Establish assurance requirements	Understand the operating environment and define the operating constraints for mission and information assurance within the environments of system development.		
	Develop customer mission and information assurance requirements		
	Define product and product component assurance requirements		
	Identify operational concepts and associated scenarios for intended and unintended use and associated assurance considerations		
	Identify appropriate controls for integrity and availability of the system to in support of organizational objectives		
	Analyze assurance requirements		
	Balance assurance needs against cost benefits		
	Obtain Agreement of risk for assurance level		

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

It can be used as a navigation tool to guide SwA implementation efforts

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

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



SwA Community's Assurance Process Reference Model - Initial Mappings

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

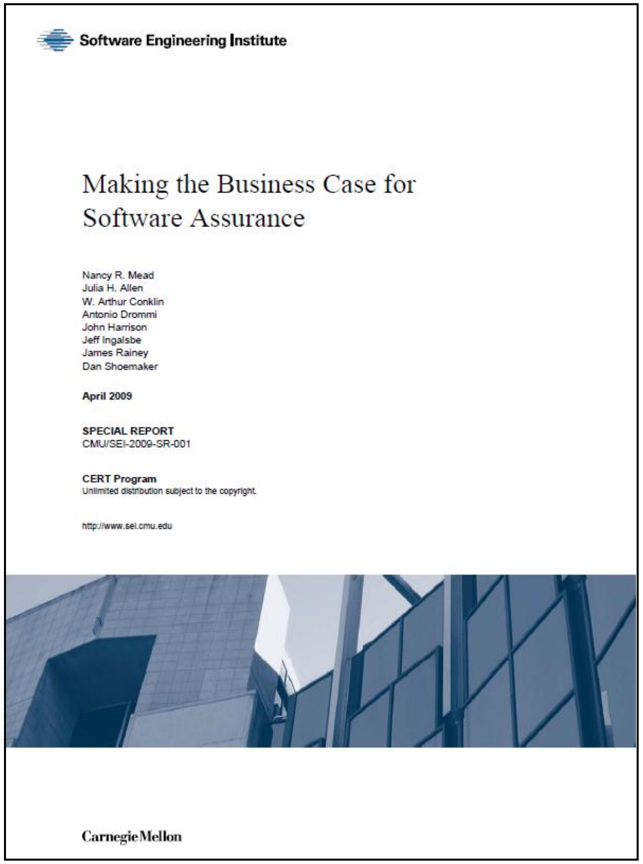
Goal	Practice	AF CMMI	BSIMM	CMMI-ACQ	CMMI-DEV	CMMI-SVC	OSAMM	RMM
DE 2 Create IT solutions with integrated business objectives and assurance	Develop alternative solutions and selection criteria for mission and information assurance.	AF TS SP 1.1.1	SFD1.1	ATM SG2	TS SG1		SA1A	RTSE:SG 1 - SG2
			SFD1.2	AVAL SG2			SA1B	KIM:SG2, SG6
	Architect for mission and information assurance.	AF TS SP 2.1.1	SFD2.1	ATM SG2	TS SG2		SA2A	RTSE:SG 3
			SFD2.3	AVAL SG2	TS SG2		SA2B	
	Design for mission and information assurance.	AF TS SP 2.1.2	SFD2.1		TS SG2			
	Implement the mission and information assurance designs of the product components.	AF TS SP 3.1.1	AA3.2		TS SG3		SA1B	
	Identify deviations from mission and information assurance coding standards. Implement appropriate mitigation to meet defined mission and information assurance objectives.	AF TS SP 3.1.2	CR1.4	AVER SG3	TS SG3		CR2A	RTSE:SG 2
			CR2.3				CR2B	RTSE:SG 3
			CR3.1				CR3A	

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

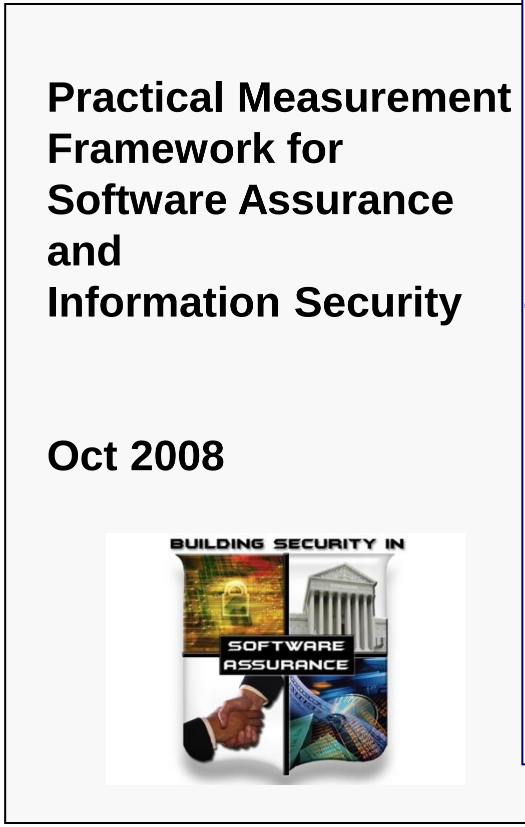
CERT Program
Unlimited distribution subject to the copyright.

http://www.sei.cmu.edu

CarnegieMellon

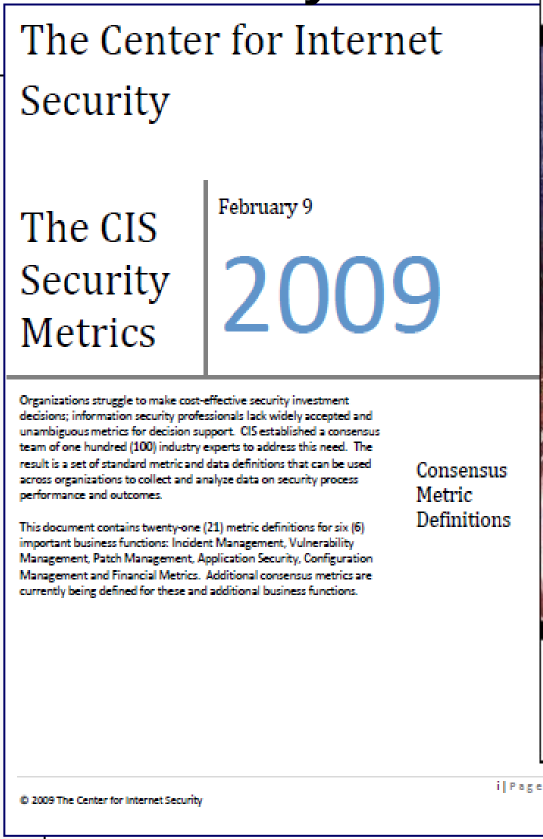
Security Measurement Resources

Oct 08 → Feb 09 → May 09 →



Practical Measurement Framework for Software Assurance and Information Security

Oct 2008



The Center for Internet Security

The CIS Security Metrics

February 9

2009

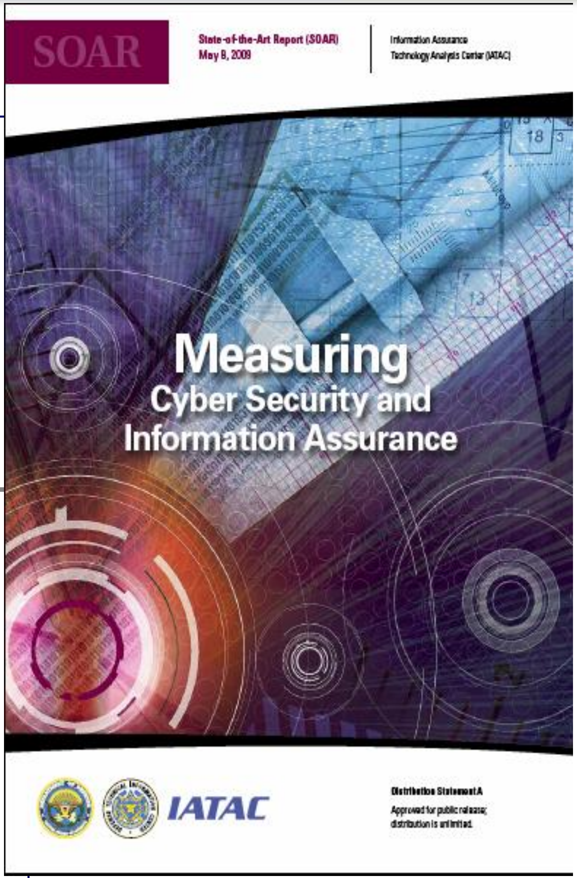
Organizations struggle to make cost-effective security investment decisions; information security professionals lack widely accepted and unambiguous metrics for decision support. CIS established a consensus team of one hundred (100) industry experts to address this need. The result is a set of standard 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.

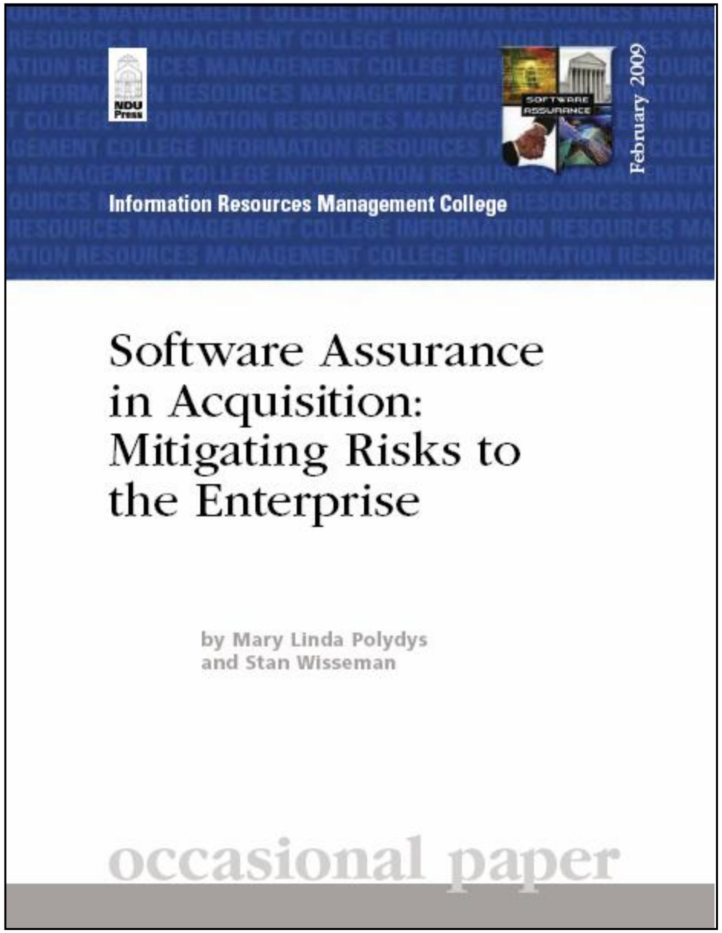
SwA Acquisition & Outsourcing Handbook

“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

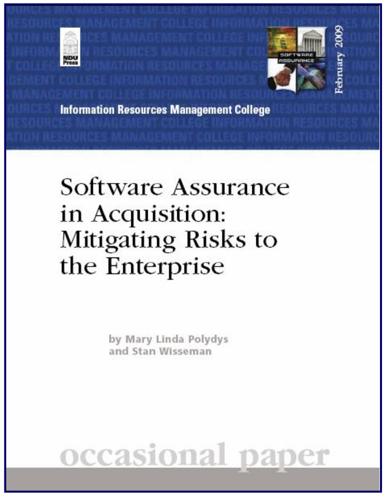
Information Resources Management College

Software Assurance in Acquisition: Mitigating Risks to the Enterprise

by Mary Linda Polydys
and Stan Wisseman

occasional paper

SwA Acquisition & Outsourcing Handbook


February 2009

Information Resources Management College

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?

More Due-Diligence Questions Relevant to Acquisition & Outsourcing

- **Relevant to deliberate actions that are controllable and preventable by developers that have security implications**
  - **“Were any compiler warnings disabled for the software being delivered?”**

- **Relevant to hosted applications and services**
  - **Cloud computing, “XXXX_as a Service,” SOA,**

Seeking more examples from “security aware” community

Supply Chain Risk Management (SCRM) processes, tools and techniques:

- Numerous SCRM processes, tools and techniques facilitate the implementation of SCRM USG-wide. Departments and Agencies shall adopt and tailor these recommended SCRM processes, tools, and techniques, and apply them to the procurement and operation of mission critical elements within NSS, to include those which:
  - Control the quality, configuration, and security of software, hardware, and systems throughout their lifecycles, including commercial elements or sub-elements.
  - Detect the occurrence, reduce the likelihood of occurrence, and mitigate the consequences of products containing counterfeit elements or malicious functions.
  - Develop requirements or capabilities to detect the occurrence of vulnerabilities within custom and commodity hardware and software through enhanced test and evaluation.

SCRM processes, tools and techniques:

- Enhance security through the implementation of system security engineering (e.g. criticality analysis and defensive engineering practices) throughout the system life cycle.
- Optimize acquisition and contracting to define requirements and source selection criteria that reduce supply chain risk, give preference to vendors that minimize supply chain risk in verifiable ways, and evaluate security equally with other desirable factors, such as low cost, rapid deployment, or new features.
- Implement acquisition processes to document and monitor risk mitigation methods and requirements and provide for the update of documentation throughout the system lifecycle.

Best Practices, Tools and Techniques References

General SCRM References

The following documents provide systems security engineering guidance and detailed risk management best practice for use in commercial or government systems.

Draft NISTIR 7622, *Piloting Supply Chain Risk Management for Federal Information Systems*, June 2010.

National Defense Industrial Association (NDIA) System Assurance Committee. 2008. *Engineering for System Assurance*.

SCRM References from the Department of Defense

The following documents describe SCRM best practice for NSS, provide guidance on the successful implementation of SCRM pilots that incorporate all-source threat information, summarize the DoD pilot experience, and identify trusted suppliers of integrate circuits as accredited by the Defense Microelectronic Agency.

Key Practices and Implementation Guide for the DoD Comprehensive National Cybersecurity Initiative 11: Supply Chain Risk Management Pilot Program. February 25, 2010.

Concept of Operations for the DoD Comprehensive National Cybersecurity Initiative 11: Supply Chain Risk Management Pilot Program. August 25, 2009.

Draft Comprehensive National Cybersecurity Initiative (CNCI) DoD Supply Chain Risk Management (SCRM) Pilot Program Report, November 30, 2010

List of Trusted Integrated Circuits (IC) Suppliers available at http://www.dmea.osd.mil

SCRM References from the Department of Homeland Security

The following documents and the assessment tool provide guidance for civilian Departments and agencies guidance for the successful implementation of a SCRM pilot. Used with the NISTIR 7622, which identifies key practices, the following documents enable the development and operation of systems to manage supply chain risks.

- Concept of Operations for the Civilian Agency Pilot Program (CAPP)
- Template for a SCRM Pilot Plan of Action and Milestones
- SCRM Capability Assessment Tool

Best Practices, Tools and Techniques References

Software Assurance Community documents from Software.Assurance@dhs.gov

Software Assurance in Acquisition and Contract Language (https://buildsecurityin.us-cert.gov/swa/pocket_guide_series.html#acquisition)

Software Supply Chain Risk Management and Due Diligence (https://buildsecurityin.us-cert.gov/swa/pocket_guide_series.html#acquisition)

Polydys, Mary L. and Wisseman, Stan., Software Assurance in Acquisition: Mitigating Risks to the Enterprise, A Reference Guide for Security-Enhanced Software Acquisition and Outsourcing, Information Resources Management College Occasional Paper, National Defense University Press, Washington, D.C. February 2009. (http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA495389)

Polydys, Mary L. and Wisseman, Stan. (2007, May). "Software Assurance: Five Essential Considerations for Acquisition Officials." CrossTalk—The Journal of Defense Software Engineering, Vol. 20, No. 5. (https://buildsecurityin.us-cert.gov/swa/downloads/PolydysWisseman.pdf)

Robert J. Ellison, John B. Goodenough, Charles B. Weinstock, Carol Woody Evaluating and Mitigating Software Supply Chain Security Risks, May 2010 (https://buildsecurityin.us-cert.gov/swa/downloads/MitigatingSWsupplyChainRisks10tn016.pdf)

Goertzel, Karen, Theodore Winograd, et al. for Department of Homeland Security and Department of Defense Data and Analysis Center for Software. Enhancing the Development Life Cycle to Produce Secure Software: A Reference Guidebook on Software Assurance, October 2008. (https://www.thedacs.com/techs/enhanced_life_cycles/)

Bob Ellison, CERT, Software Engineering Institute and Carol Woody, CERT, Software Engineering Institute, " Considering Software Supply Chain Risks," CrossTalk—The Journal of Defense Software Engineering,, September/October 2010 (https://buildsecurityin.us-cert.gov/bsi/1207-BSI/version/1/part/4/data/1009EllisonWoody.pdf?branch=main&language=default)

Bob Ellison, CERT, Software Engineering Institute and Carol Woody, CERT, Software Engineering Institute, Supply-Chain Risk Management: Incorporating Security into Software Development, March 2010 (https://buildsecurityin.us-cert.gov/bsi/articles/best-practices/acquisition/1140-BSI.html)

Best Practices, Tools and Techniques References

Industry Standards for SCRM

EIA-4899 - Standard for Preparing an Electronic Component Management Plan

IDEA-STD-1010 – Diminishing Manufacturing Sources and Material Shortages (DMSMS) Guidebook

SAE-AS9120 – Quality Management Systems for Aerospace Product Distributors

SAE-AS5553 – Counterfeit Electronic Parts; Avoidance, Detection, Mitigation and Disposition

Federal IT Security References

The following documents provide a foundation of federal information technology security practices or provide detailed guidance specific to managing risks inherent in the information technology product or services supply chain.

CNSS Instruction No. 1253, *Security Categorization and Control Selection for National Security Systems*, October 2009

NIST Special Publication 800-53 Revision 3, *Recommended Security Controls for Federal Information Systems and Organizations*, August 2009 (includes updates as of 05-01-2010).

NIST Special Publication 800-37 Revision 1, Guide for Applying the Risk Management Framework to Federal Information Systems: *A Security Life Cycle Approach*, February 2010.

What Do The Informational Building Blocks for “Architecting Security” Look Like?

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

The Building Blocks Are:

- Enumerations
  - **Catalog the fundamental entities in IA, Cyber Security, and Software Assurance**
    - **Vulnerabilities (CVE), configuration issues (CCE), software packages (CPE), attack patterns (CAPEC), weaknesses in code/design/architecture (CWE), observables (CYBOX)**
- Languages/Formats
  - **Support the creation of machine-readable state assertions, assessment results, and messages**
    - **Configuration/vulnerability/patch/asset patterns (XCCDF & OVAL), results from standards-based assessments (ARF), event patterns (CEE), malware patterns (MAEC), risk of a vulnerability (CVSS), config risk (CCSS), weakness risk (CWSS), assessment findings (SAFES/SACM), information messages (CYBEX/IODEF)**
- Knowledge Repositories
  - **Packages of assertions supporting a specific application**
    - **Vulnerability advisories & alerts, (US-CERT Advisories/IAVAs), configuration assessment (NIST Checklists, CIS Benchmarks, NSA Configuration Guides, DISA STIGS), asset inventory (NIST/DHS NVD), code assessment & certification (NIST SAMATE, DoD DIACAP & eMASS)**

Tools

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

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)

SwA-funded Cyber Ecosystem Standardization ★

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)	

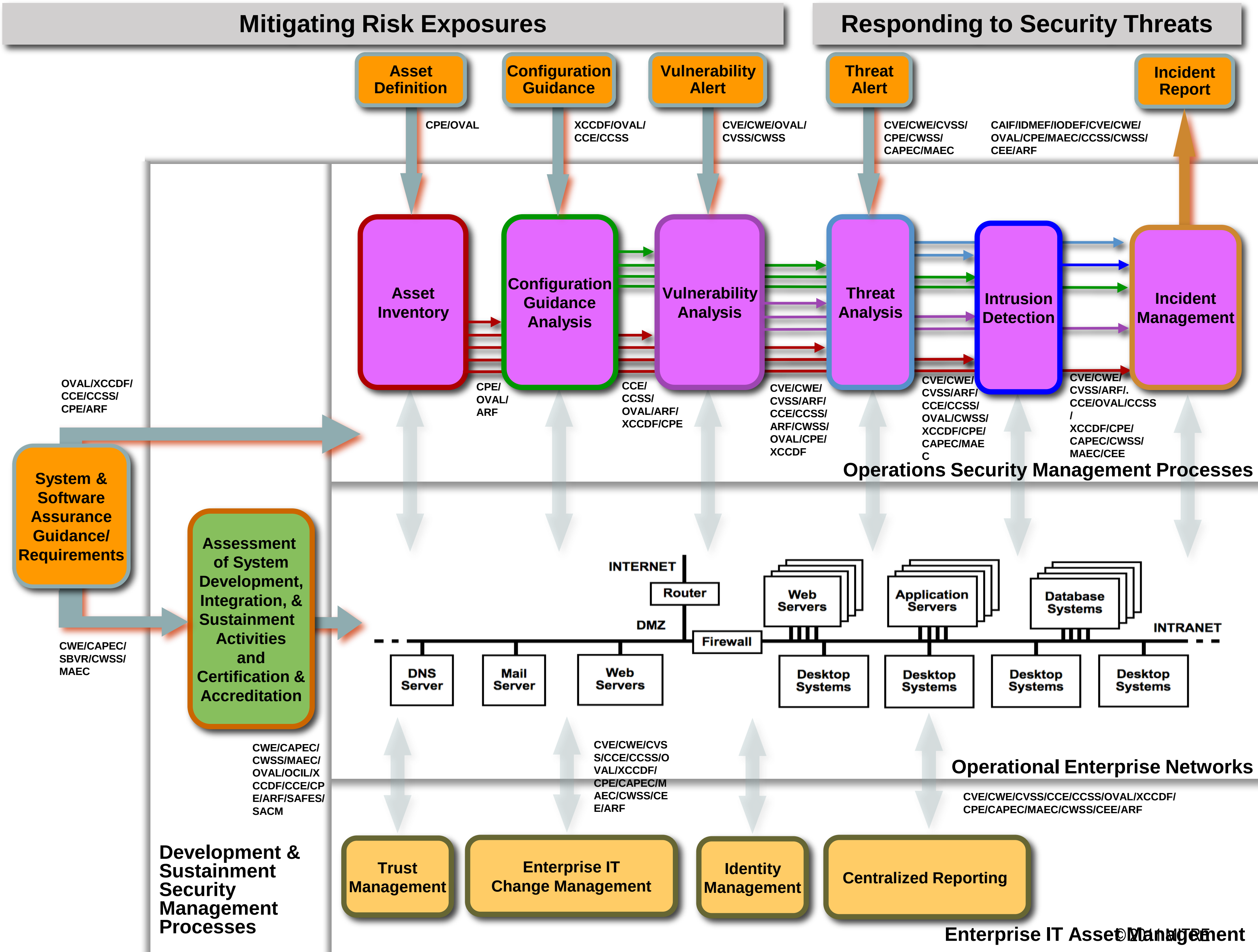
Standardization Efforts leveraged by the Security Content Automation Protocol (SCAP)

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

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)	

New FISMA reporting requirements



Mitigating Risk Exposures

Responding to Security Threats

Asset Definition

CPE/OVAL

Configuration Guidance

XCCDF/OVAL/
CCE/CCSS

Vulnerability Alert

CVE/CWE/OVAL/
CVSS/CWSS

Threat Alert

CVE/CWE/CVSS/
CPE/CWSS/
CAPEC/MAEC

CAIF/IDMEF/IODEF/CVE/CWE/
OVAL/CPE/MAEC/CCSS/CWSS/
CEE/ARF

Incident Report

Asset Inventory

CPE/
OVAL/
ARF

Configuration Guidance Analysis

CCE/
CCSS/
OVAL/ARF/
XCCDF/CPE

Vulnerability Analysis

CVE/CWE/
CVSS/ARF/
CCE/CCSS/
ARF/CWSS/
OVAL/CPE/
XCCDF

Threat Analysis

CVE/CWE/
CVSS/ARF/
CCE/CCSS/
OVAL/CWSS/
XCCDF/CPE/
CAPEC/MAE
C

Intrusion Detection

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

Incident Management

Operations Security Management Processes

OVAL/XCCDF/
CCE/CCSS/
CPE/ARF

System & Software Assurance Guidance/ Requirements

CWE/CAPEC/
SBVR/CWSS/
MAEC

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

CWE/CAPEC/
CWSS/MAEC/
OVAL/OCIL/X
CCDF/CCE/CP
E/ARF/SAFES/
SACM

INTERNET

Router

DMZ

Firewall

INTRANET

DNS Server

Mail Server

Web Servers

Web Servers

Application Servers

Database Systems

Desktop Systems

Desktop Systems

Desktop Systems

Desktop Systems

Operational Enterprise Networks

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

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

Trust Management

Enterprise IT Change Management

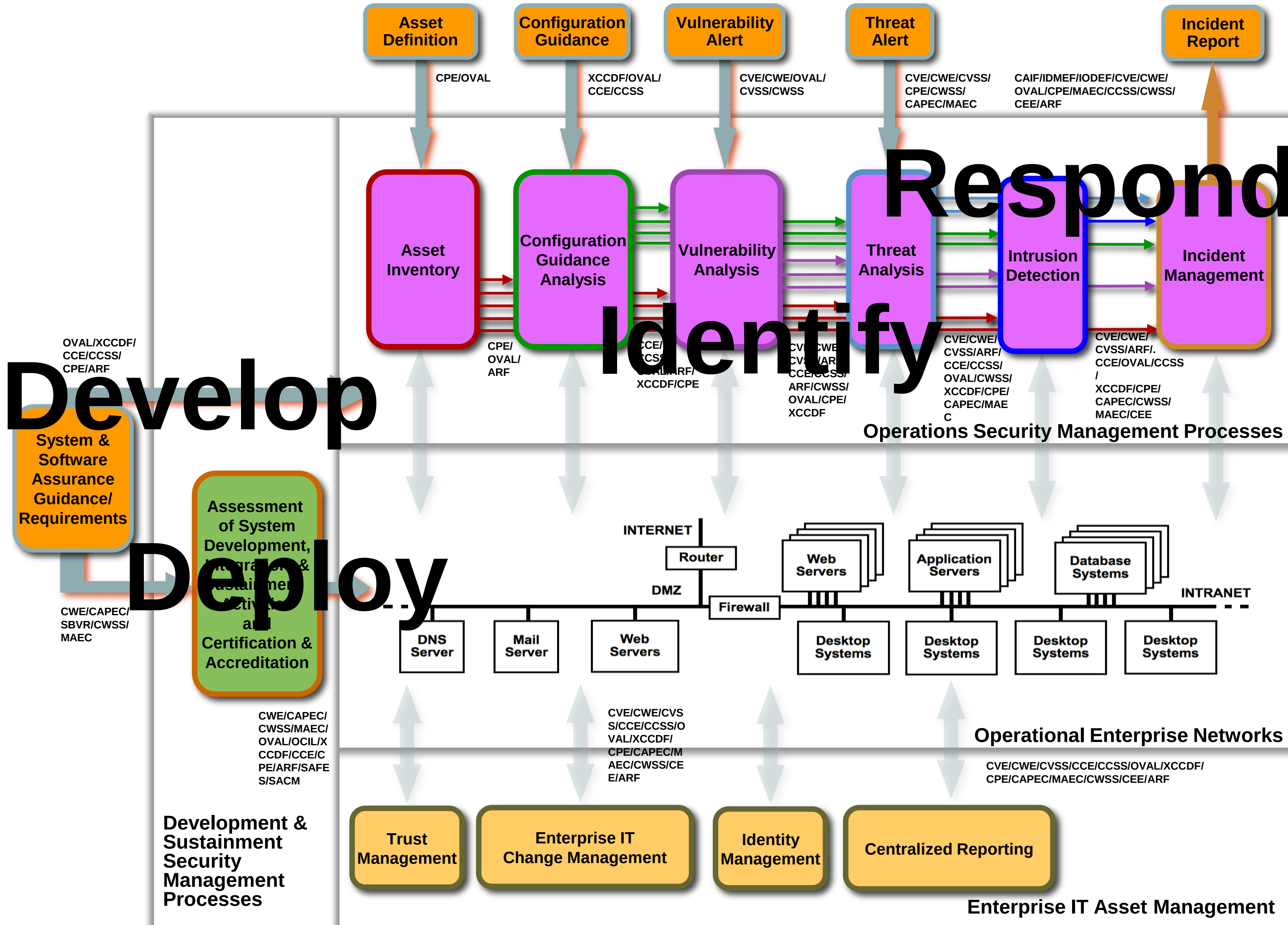
Identity Management

Centralized Reporting

Development & Sustainment Security Management Processes

Enterprise IT Asset Management

Knowledge Repositories



Asset Definition

CPE/OVAL

Configuration Guidance

XCCDF/OVAL/
CCE/CCSS

Vulnerability Alert

CVE/CWE/OVAL/
CVSS/CWSS

Threat Alert

CVE/CWE/CVSS/
CPE/CWSS/
CAPEC/MAEC

CAIF/IDMEF/IODEF/CVE/CWE/
OVAL/CPE/MAEC/CCSS/CWSS/
CEE/ARF

Incident Report

Respond

Asset Inventory

Configuration Guidance Analysis

Vulnerability Analysis

Threat Analysis

Intrusion Detection

Incident Management

CPE/
OVAL/
ARF

Identify

CCE/
CCSS/
OVAL/ARF/
XCCDF/CPE

CVE/CWE/
CVSS/ARF/
CCE/CCSS/
ARF/CWSS/
OVAL/CPE/
XCCDF

CVE/CWE/
CVSS/ARF/
CCE/CCSS/
OVAL/CWSS/
XCCDF/CPE/
CAPEC/MAE
C

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

Operations Security Management Processes

OVAL/XCCDF/
CCE/CCSS/
CPE/ARF

Develop

System & Software Assurance Guidance/ Requirements

Deploy

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

CWE/CAPEC/
SBVR/CWSS/
MAEC

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/
CWSS/MAEC/
OVAL/OCIL/X
CCDF/CCE/C
PE/ARF/SAFE
S/SACM

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

Operational Enterprise Networks

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

Development & Sustainment Security Management Processes

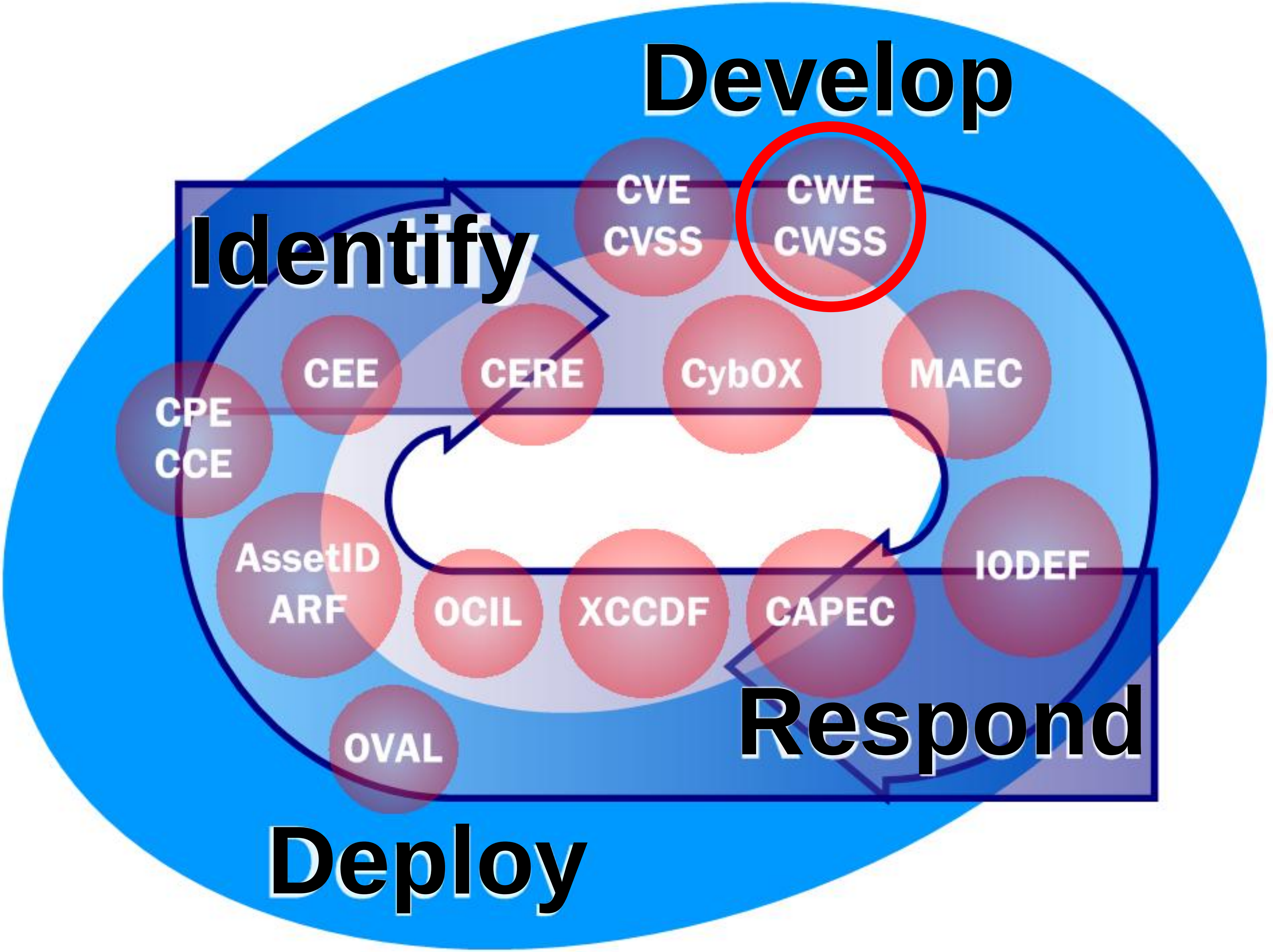
Trust Management

Enterprise IT Change Management

Identity Management

Centralized Reporting

Enterprise IT Asset Management


Develop
Identify
CVE
CVSS
CWE
CWSS
CEE
CERE
CybOX
MAEC
CPE
CCE
AssetID
ARF
OCIL
XCCDF
CAPEC
IODEF
OVAL
Respond
Deploy

Leverage Common Weakness Enumeration (CWE) to mitigate risks to mission/business domains

CWE is a formal list of software weakness types created to:

- Serve as a common language for describing software security weaknesses in architecture, design, or code.
- Serve as a standard measuring stick for software security tools targeting these weaknesses.
- Provide a common baseline standard for weakness identification, mitigation, and prevention efforts.

Some Common Types of Software Weaknesses:

Buffer Overflows, Format Strings, Etc.
Structure and Validity Problems
Common Special Element Manipulations
Channel and Path Errors
Handler Errors
User Interface Errors
Pathname Traversal and Equivalence
Errors
Authentication Errors
Resource Management Errors
Insufficient Verification of Data
Code Evaluation and Injection
Randomness and Predictability

Free Resources and Events

SwA Working Groups – Next meeting: Week of Nov 28, 2011
@ MITRE in McLean, VA

SwA Forum – Next Forum: Week of March 26, 2012
@ MITRE in McLean, VA

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

Making Security Measureable: measurablesecurity.mitre.org

Email: software.assurance@dhs.gov

See Language for sharing correlation of incident information -- Cyber Observables eXpression (CybOX) at http://cybox.mitre.org

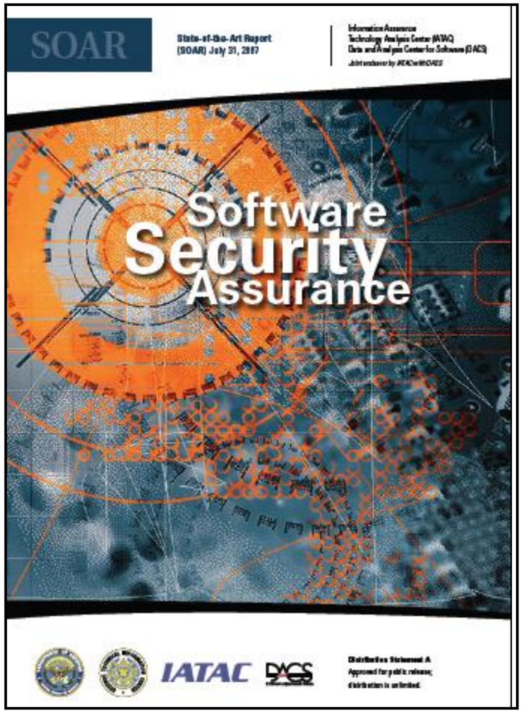
IT/Software Supply Chain Management is a National Security & Economic Issue

- Adversaries can gain “intimate access” to target systems, especially in a global supply chain that offers limited transparency
- Advances in science and technology will always outpace the ability of government and industry to react with new policies and standards
  - National security policies must conform with international laws and agreements while preserving a nation’s rights and freedoms, and protecting a nation’s self interests and economic goals
  - Forward-looking policies can adapt to the new world of global supply chains
  - Information standards, process standards, and product standards must mature to better address supply chain risk management, security, & systems/software assurance
  - Assurance Rating Schemes for software products and organizations are needed
- IT/software suppliers and buyers can take more deliberate actions to security-enhance their processes and practices to mitigate risks
  - Government & Industry have significant leadership roles in solving this
  - Individuals can influence the way their organizations adopt security practices

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

State of the Art Report on Software Security Assurance

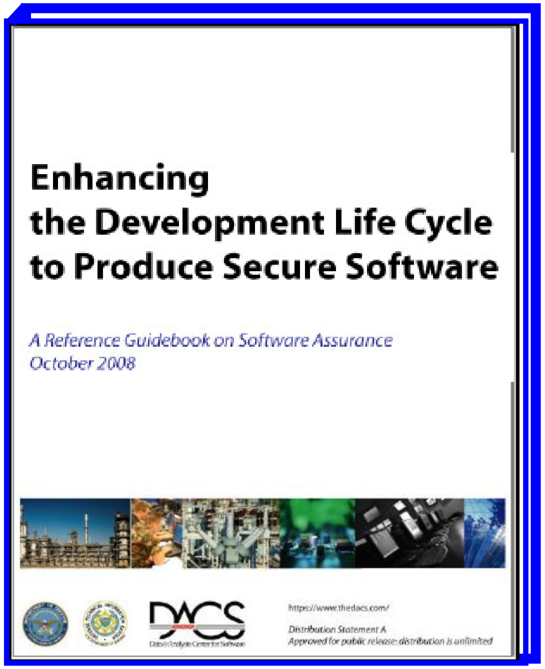
- An IATAC/DACS report identifying and describing the current state of the art in software security assurance, including trends in:
  - Techniques for the production of secure software
  - Technologies that exist or are emerging to address the software security challenge
  - Current activities and organizations in government, industry, and academia, in the U.S. and abroad, that are devoted to systematic improvement of software security
  - Research trends worldwide that might improve the state of the art for software security
- Available free via http://iac.dtic.mil/iatac/download/security.pdf


SOAR
State-of-the-Art Report
(SOAR) July 31, 2007
Information Assurance
Technology Analysis Center (IATAC)
Data and Analysis Center for Software (DACS)
Joint endeavor by IATAC with DACS
Software
Security
Assurance
IATAC
DACS
Distribution Statement A
Approved for public release;
distribution is unlimited.

Enhancing the Development Life Cycle to Produce Secure Software

- Describes how to integrate security principles and practices in software development life cycle
- Addresses security requirements, secure design principles, secure coding, risk-based software security testing, and secure sustainment
- Provides guidance for selecting secure development methodologies, practices, and technologies
- Available free via

  https://www.thedacs.com/techs/enhanced_life_cycles/



Enhancing the Development Life Cycle to Produce Secure Software

A Reference Guidebook on Software Assurance
October 2008

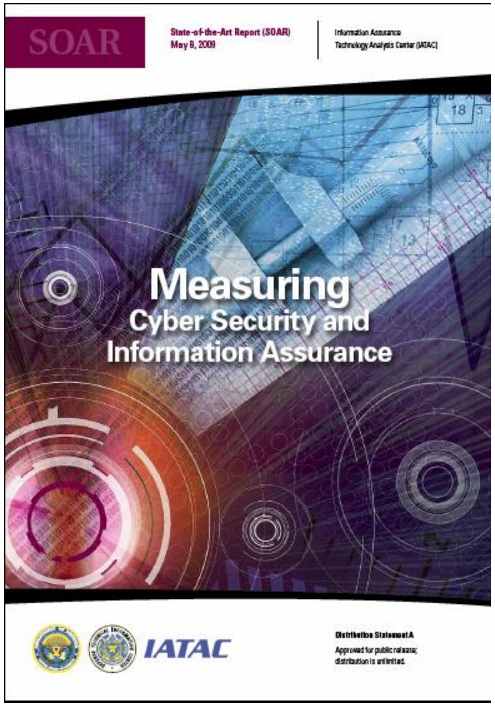
DACS
Data & Analysis Center for Software

https://www.thedacs.com/

Distribution Statement A
Approved for public release: distribution is unlimited

Measuring Cyber Security and Information Assurance

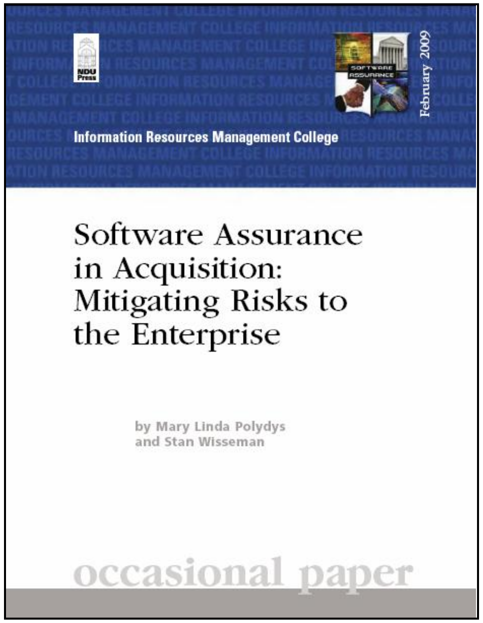
- Provides a broad picture of the current state of cyber security and information assurance (CS/IA), as well as, a comprehensive look at the progress made in the CS/IA measurement discipline over the last nine years since IATAC published its IA Metrics Critical Review and Technology Assessment (CR/TA) Report in 2000
- Available free via
- http://iac.dtic.mil/iatac/download/cybersecurity.pdf


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 in Acquisition: Mitigating Risks to the Enterprise

- Provides information on how to incorporate Software Assurance considerations in key decisions
  - How to exercise due diligence throughout the acquisition process relative to potential risk exposures that could be introduced by the supply chain
  - Includes practices that enhance SwA in the purchasing process
    - Due diligence questionnaires designed to support risk mitigation efforts by eliciting information about the software supply chain (these are also provided in Word format so they can be customized)
    - Sample contract provisions
    - Sample language to include in statements of work
- Pre-publication version available free via

  https://buildsecurityin.us-cert.gov/swa/downloads/SwA_in_Acquisition_102208.pdf
- Final version published by National Defense University Press, Feb 2009



NDU Press

SOFTWARE ASSURANCE

February 2009

Information Resources Management College

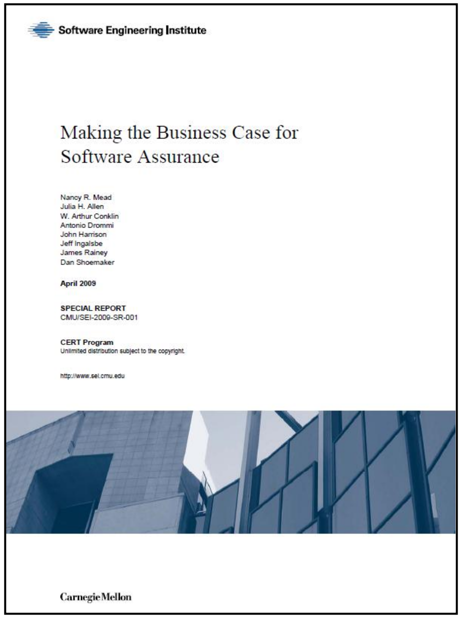
Software Assurance in Acquisition: Mitigating Risks to the Enterprise

by Mary Linda Polydys and Stan Wisseman

occasional paper

Making the Business Case for Software Assurance

- Provides background, context and examples for making the business case for software assurance:
  - Motivators
  - Cost/Benefit Models Overview
  - Measurement
  - Risk
  - Prioritization
  - Process Improvement & Secure Software
  - Globalization
  - Organizational Development
  - Case Studies and Examples
- Available free via
- http://www.sei.cmu.edu/library/abstracts/reports/09sr001.cfm



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.

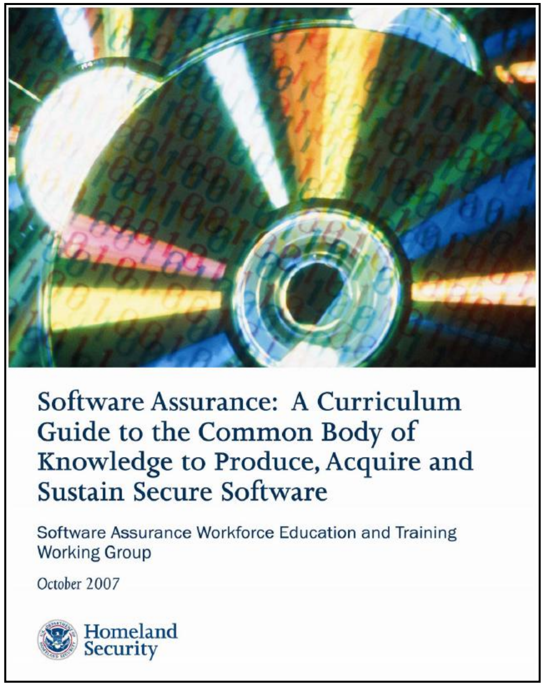
http://www.sei.cmu.edu

CarnegieMellon

Software Assurance: A Curriculum Guide to the Common Body of Knowledge to Produce, Acquire, and Sustain Secure Software

- Provides a framework intended to identify workforce needs for competencies, leverage sound practices, and guide curriculum development for education and training relevant to software assurance
- Available free via

  https://buildsecurityin.us-cert.gov/bsi/940-BSI/version/default/part/AttachmentData/data/CurriculumGuideToTheCBK.pdf



Software Assurance: A Curriculum Guide to the Common Body of Knowledge to Produce, Acquire and Sustain Secure Software

Software Assurance Workforce Education and Training Working Group

October 2007

Homeland Security

Useful Links

- DHS Build Security In Web Site
  - A wealth of software and information assurance information, including white papers on static code analysis tools
  - More information on Build Security In can be found at: https://buildsecurityin.us-cert.gov/daisy/bsi/home.html
- Common Weaknesses Enumeration (CWE)
  - This site provides a formal list of software weakness types created to Serve as a common language for describing software security weaknesses in architecture, design, or code
  - More information on CWEs can be found at:
    - http://cwe.mitre.org/
- CWE/SANS Top 25 Most Dangerous Software Errors
  - The 2010 CWE/SANS Top 25 Most Dangerous Software Errors is a list of the most widespread and critical programming errors that can lead to serious software vulnerabilities.
  - More information on the CWE/SANS Top 25 can be found at:
    - http://cwe.mitre.org/top25/archive/2010/2010_cwe_sans_top25.pdf
- NIST SAMATE Static Analysis Tool Survey
  - The National Institutes for Science and Technology (NIST), Software Assurance Metrics and Tool Evaluation (SAMATE) project, provides tables describing current static code analysis tools for source, byte, and binary code analysis
  - More information on SAMATE can be found at: http://samate.nist.gov/

Working for Homeland Security

The DHS Office of Cybersecurity and Communications (CS&C) serves as the national focal point for securing cyber space and the nation's cyber assets.

CS&C is actively seeking top notch talent in several areas including:

- Software assurance
- Information technology
- Telecommunications
- Program management
- Public affairs

To learn more about CS&C and potential career opportunities, please visit USAJOBS at www.usajobs.gov .