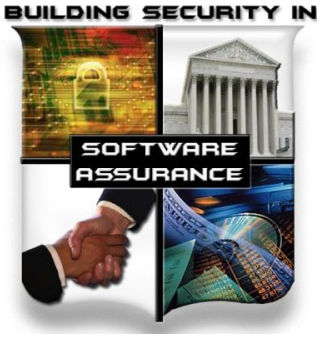


7th Annual
IT Security Automation Conference

Software Assurance

October 31, 2011

Homeland Security

MITRE

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

SOFTWARE ASSURANCE TRACK

10:45 – 11:30 am
- **Mitigating the Risk of Zero-Day Attacks with Software Security Automation**
  - **Joe Jarzombek (DHS), Tom Millar (DHS), and John Banghart (NIST)**

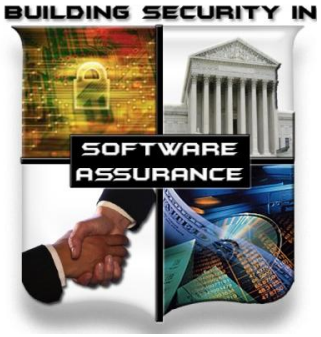
11:45 – 12:30 pm
- **Measure Software Security**
  - **Sean Barnum (MITRE)**

1:30 – 2:15 pm
- **Cyber Observables eXpression (CybOX) - Use Cases**
  - **Richard Struse (DHS) and Sean Barnum (MITRE)**

2:30 – 3:15 pm
- **Workshop: Risk Analysis and Measurement with CWRAF**
  - **Richard Struse (DHS) and Steve Christey (MITRE)**

3:45 – 4:30 pm
- **Malware Attribute Enumeration and Characterization (MAEC)**
  - **Penny Chase (MITRE) and Ivan Kirillov (MITRE)**

4:45 – 5:30 pm
- **Toward CWE Compatibility Effectiveness and CWE Coverage Claims Representation (CCR)**
  - **Paul E. Black (NIST) and Richard Struse (DHS)**



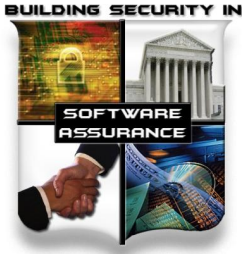
7th Annual
IT Security Automation Conference

Software Assurance:
Mitigating Risk of Zero-Day Attacks with Software Security Automation

October 31, 2011

Homeland Security

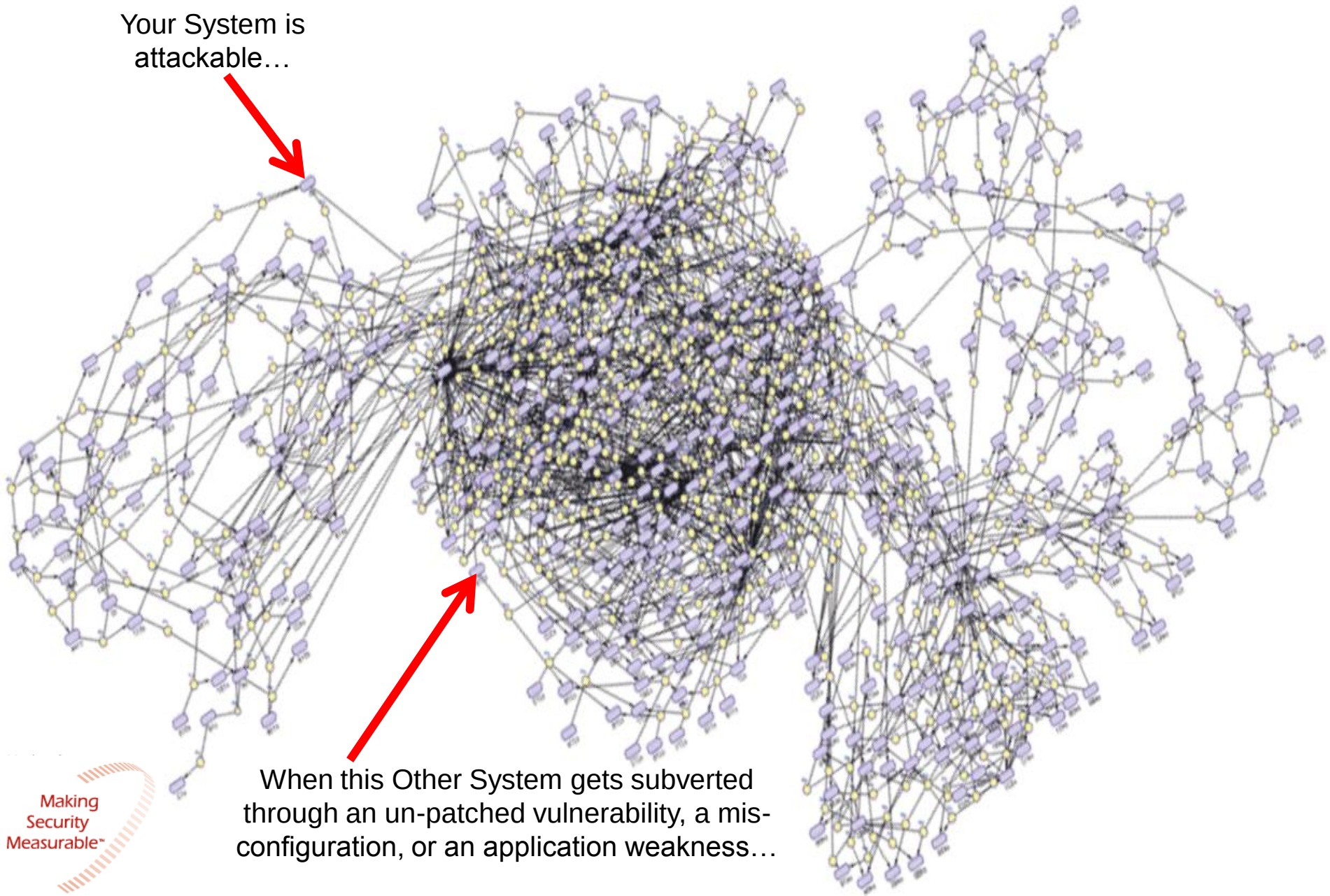
National Institute of Standards and Technology
U.S. Department of Commerce



Software Assurance

- Tom Millar: addressing the operational needs; what's the problem that has seen an exponential growth in vulnerabilities as a result exploitable software weaknesses being placed into operations, and what security automation is needed.
- John Banghart: addressing the NIST SP-enabled standards, such SCAP, Continuous Monitoring, and FISMA focused on securing what has been deployed.
- Joe Jarzombek: addressing address the use of security automation enumerations and languages; how they can be used today and how they are maturing to better enable software security automation to prevent exploitable software from being deployed.

Today Everything's Connected



Making Security Measurable™

Exploitable Software Weaknesses are sources for future Zero-Day Attacks

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**

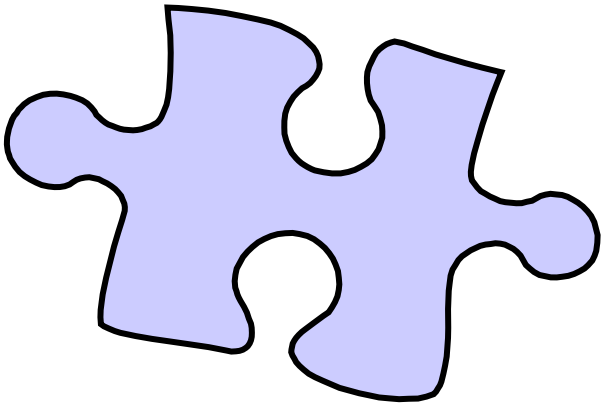
Automation

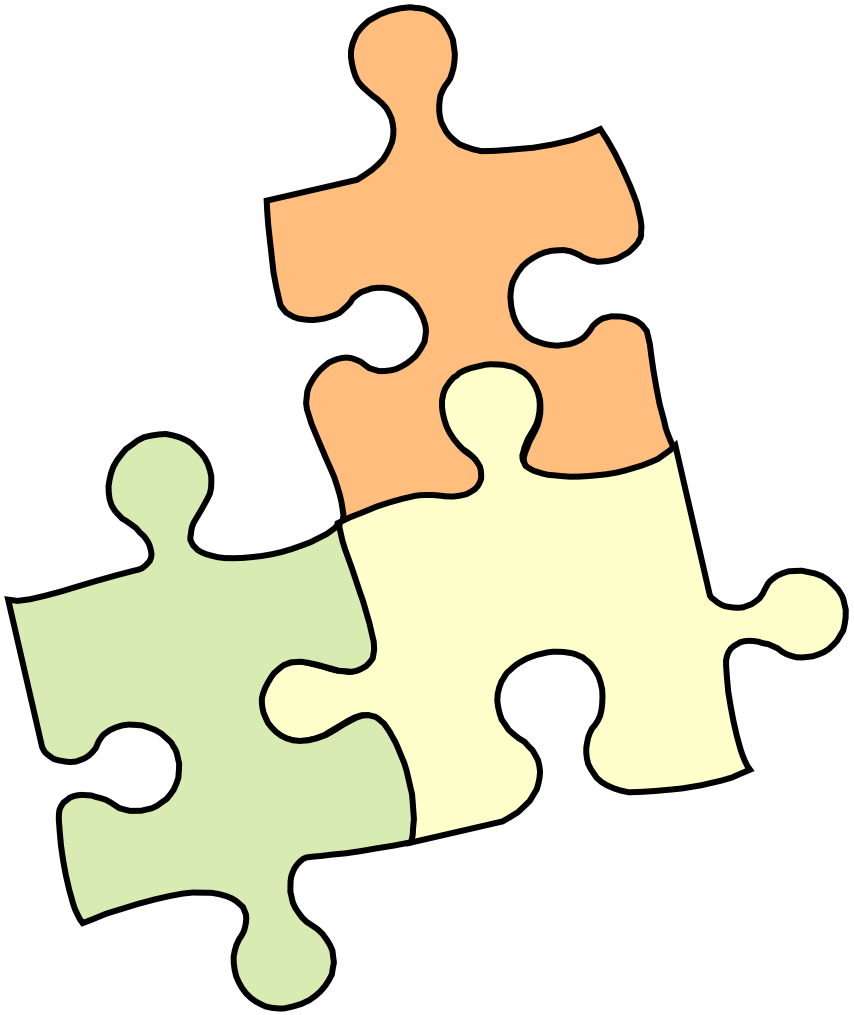
Languages, tools, enumerations and repositories

throughout the Lifecycle

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

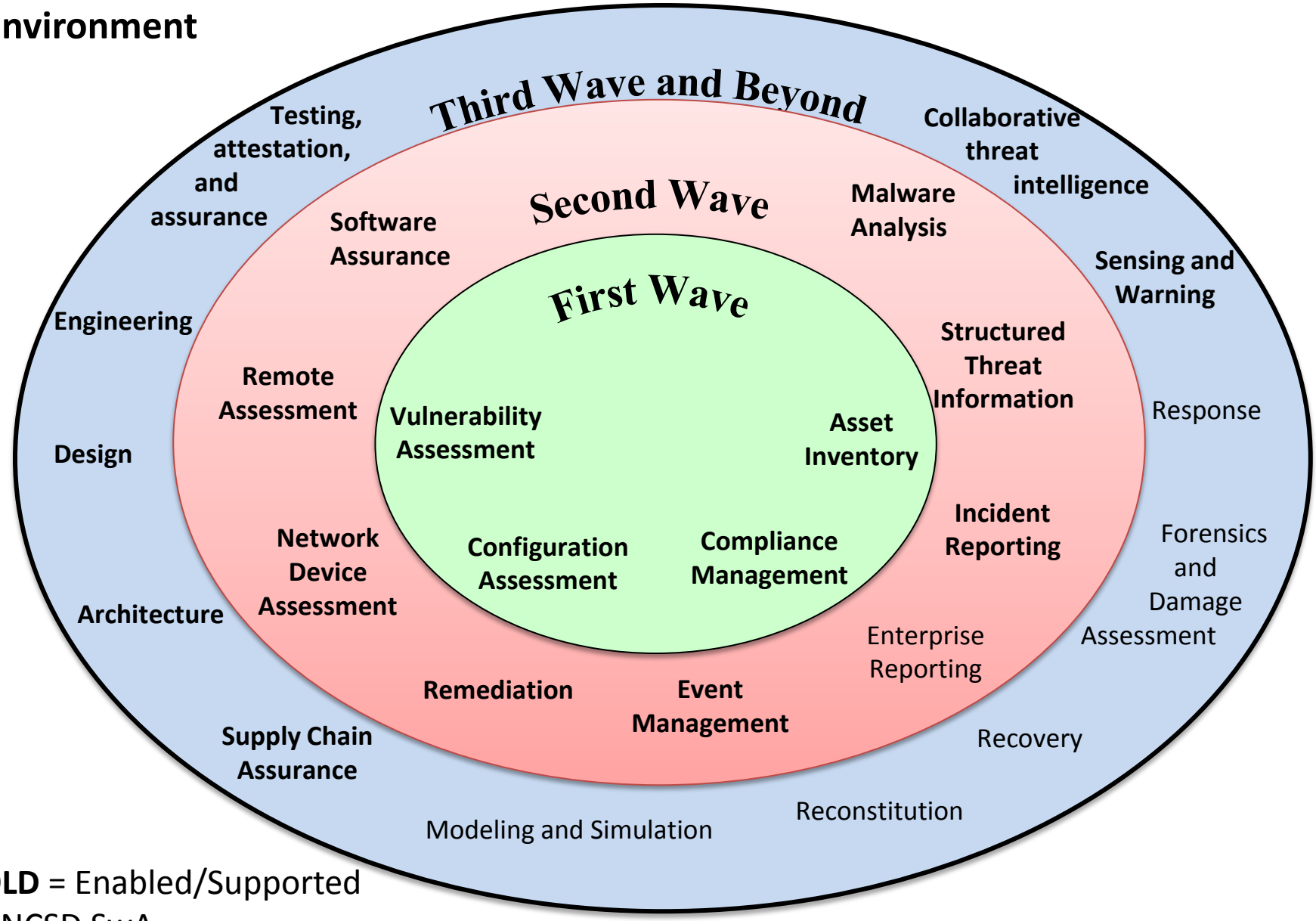
Automation is *one piece*





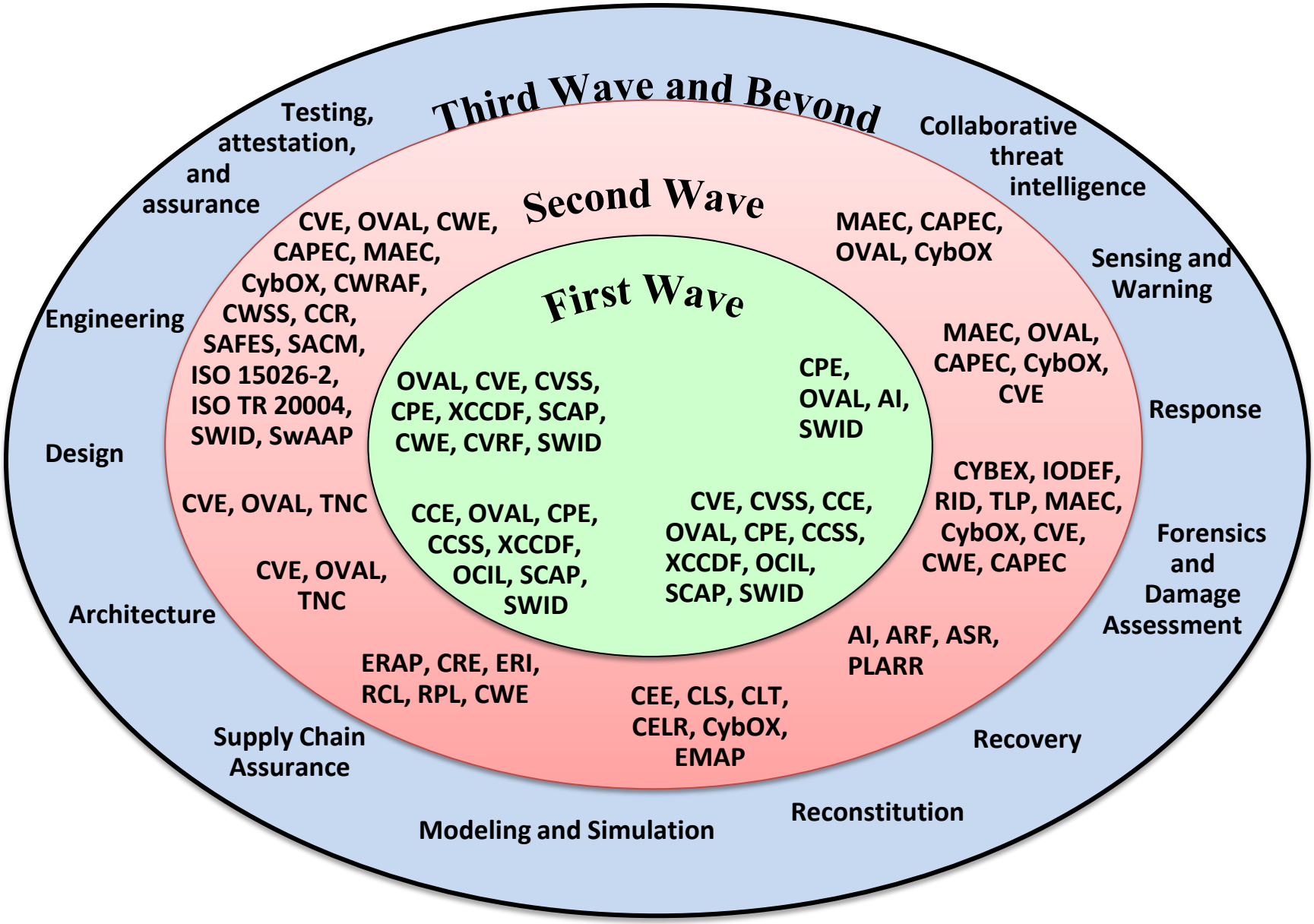
of the SwA puzzle.

"Enabling Distributed Security in Cyberspace: Building a Healthy and Resilient Cyber Ecosystem with Automated Collective Action" DHS Paper describes evolving environment

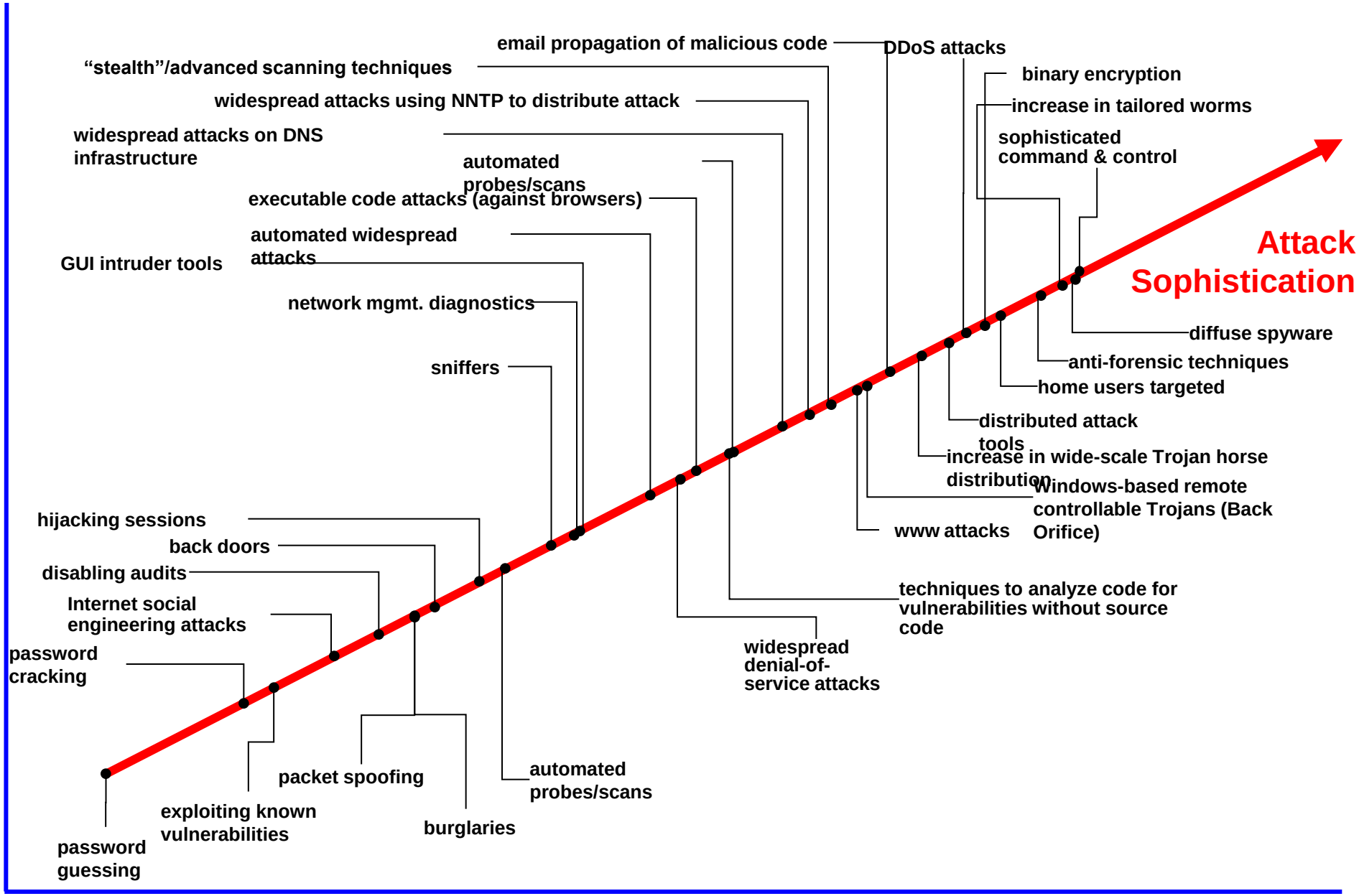


BOLD = Enabled/Supported by NCSD SwA

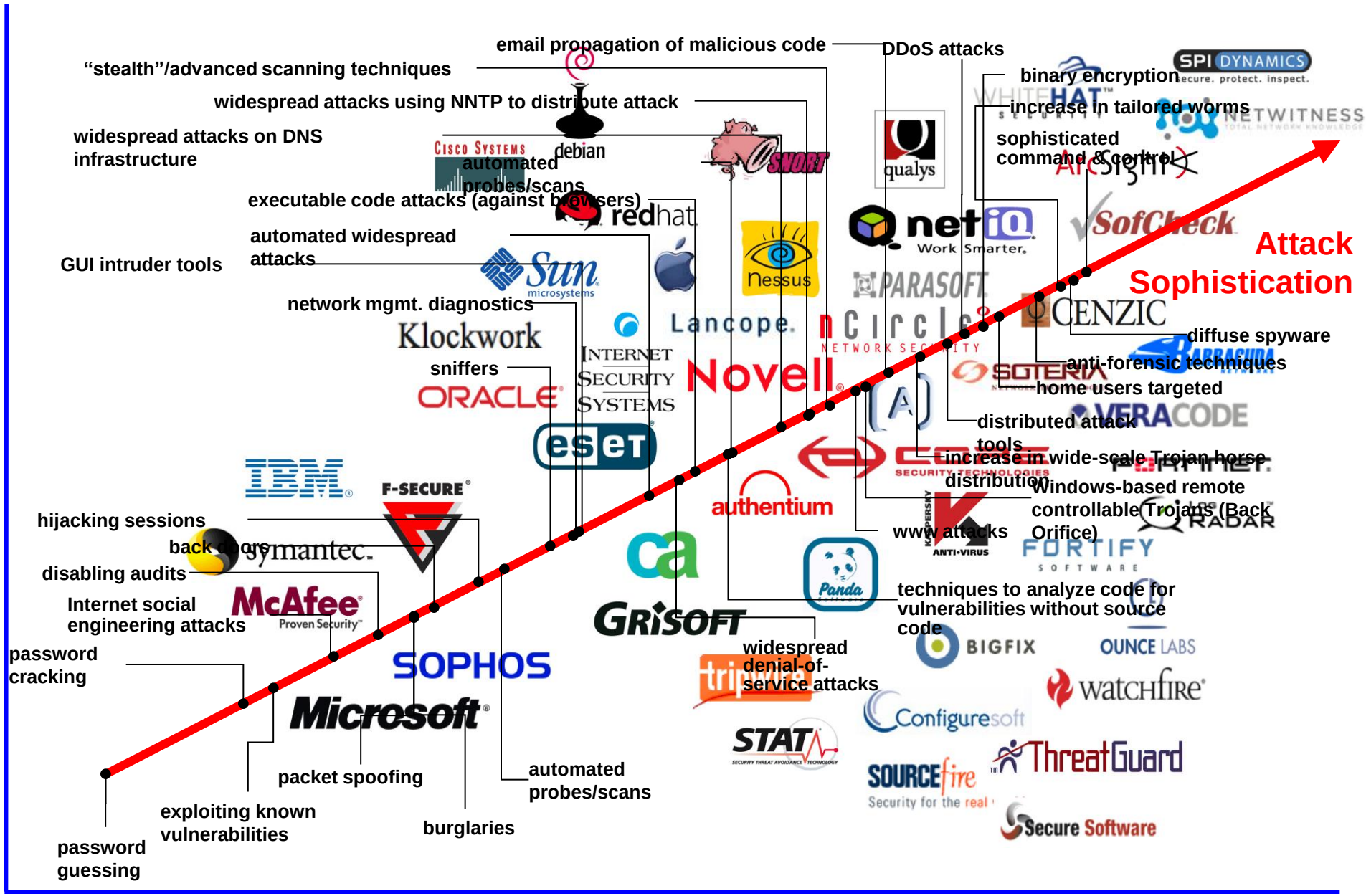
Ecosystem Areas Directly Enabled/Supported by Enumerations/Languages/Standards



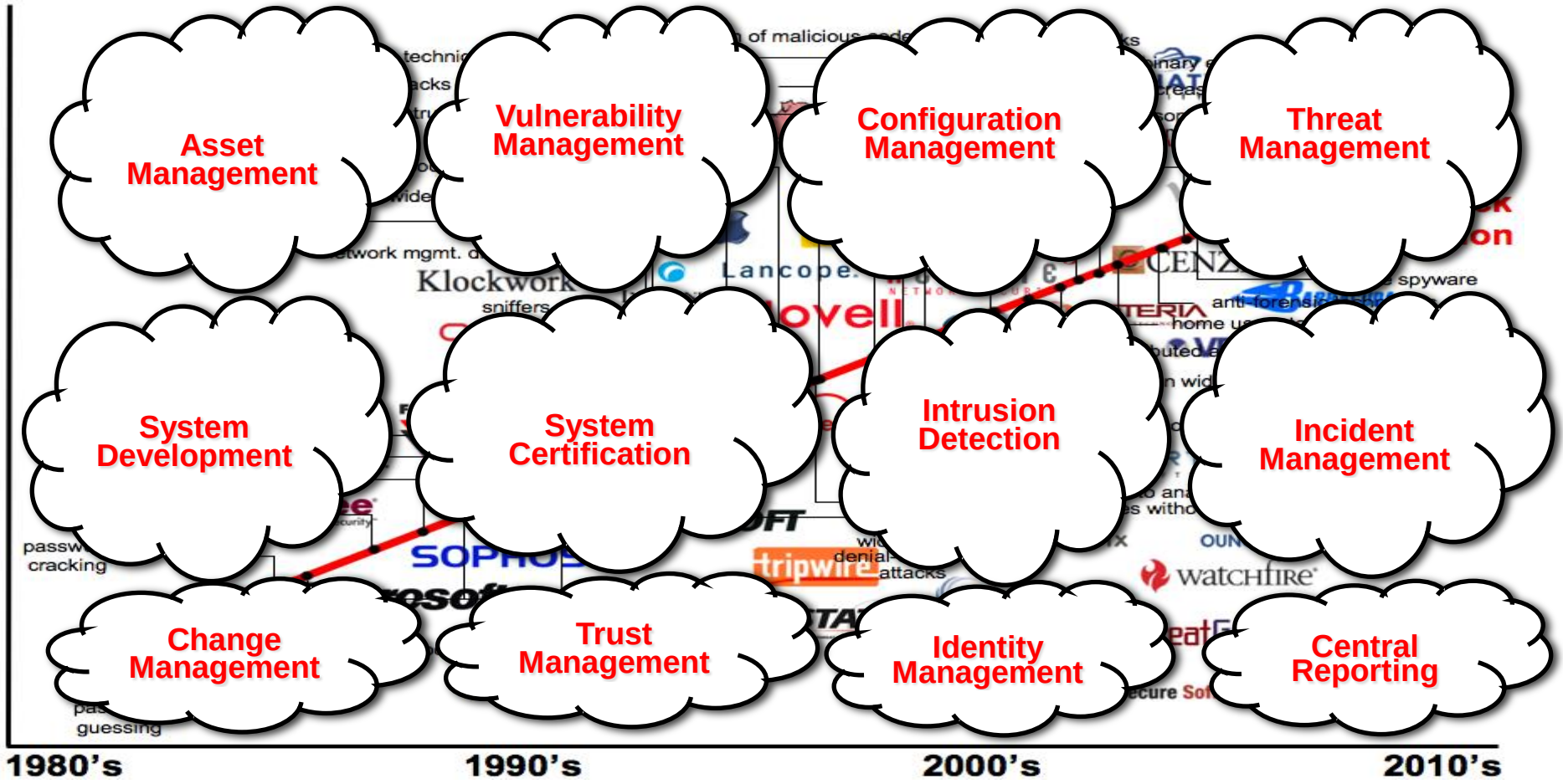
Cyber Threats Emerged Over Time



Solutions Also Emerged Over Time



Architecting Security with Information Standards for COIs



Making Security Measurable™

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

Making Security Measurable™

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)

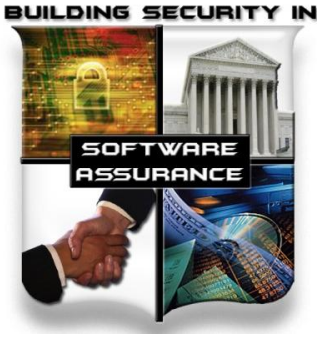
Standardization Efforts leveraged by the Security Content Automation Protocol (SCAP)

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



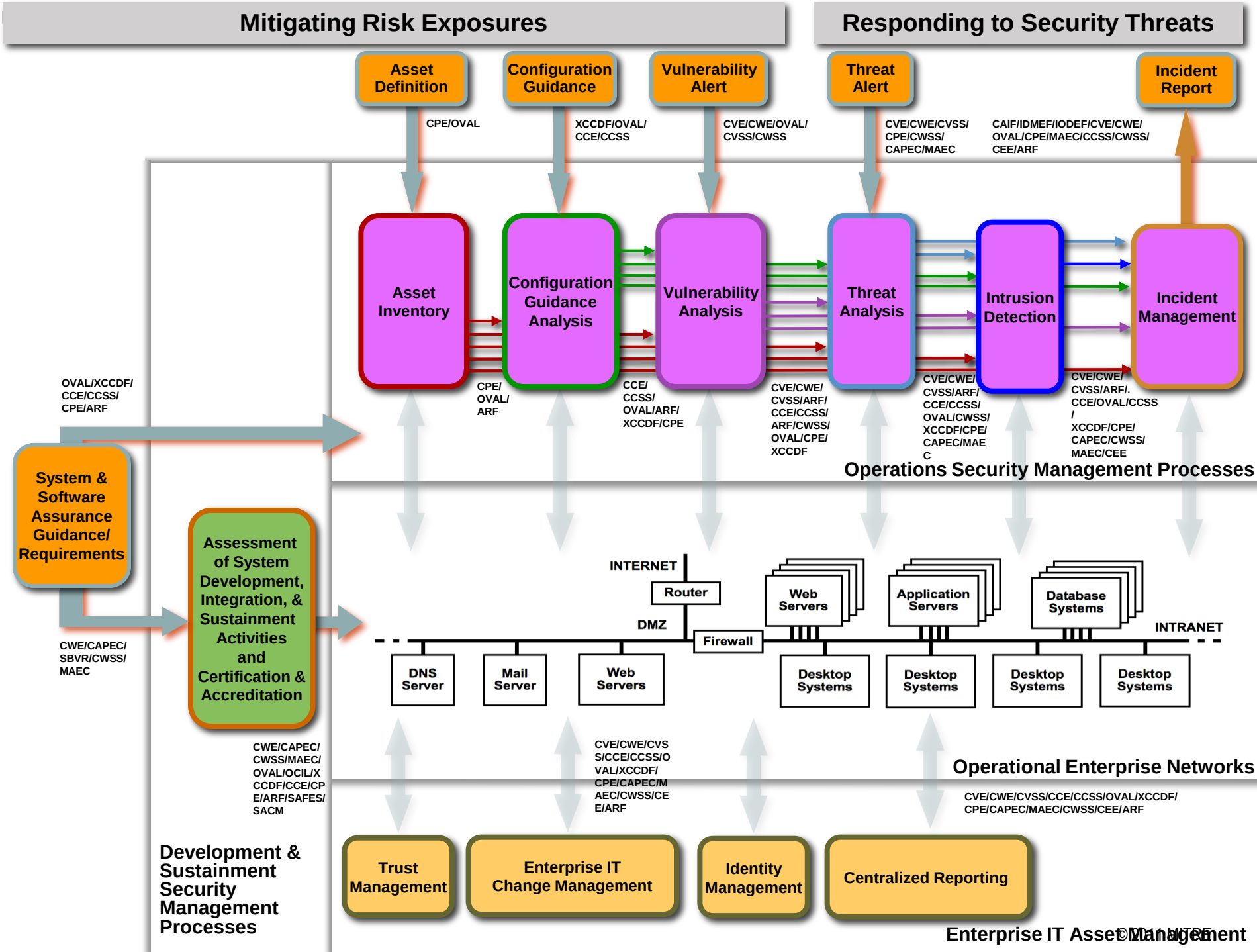
7th Annual
IT Security Automation Conference

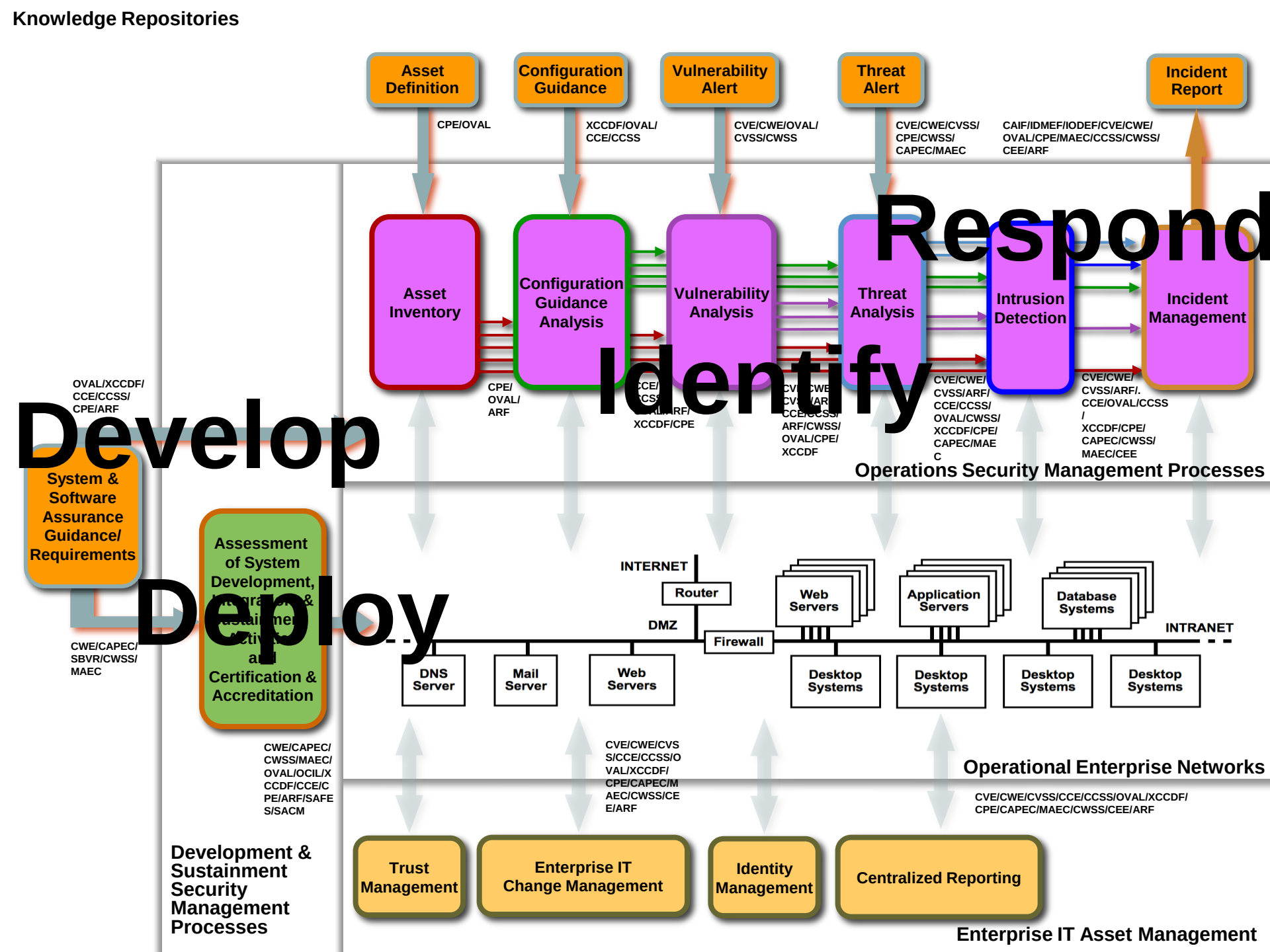
Software Assurance:
Mitigating Risk of Zero-Day Attacks with Software Security Automation

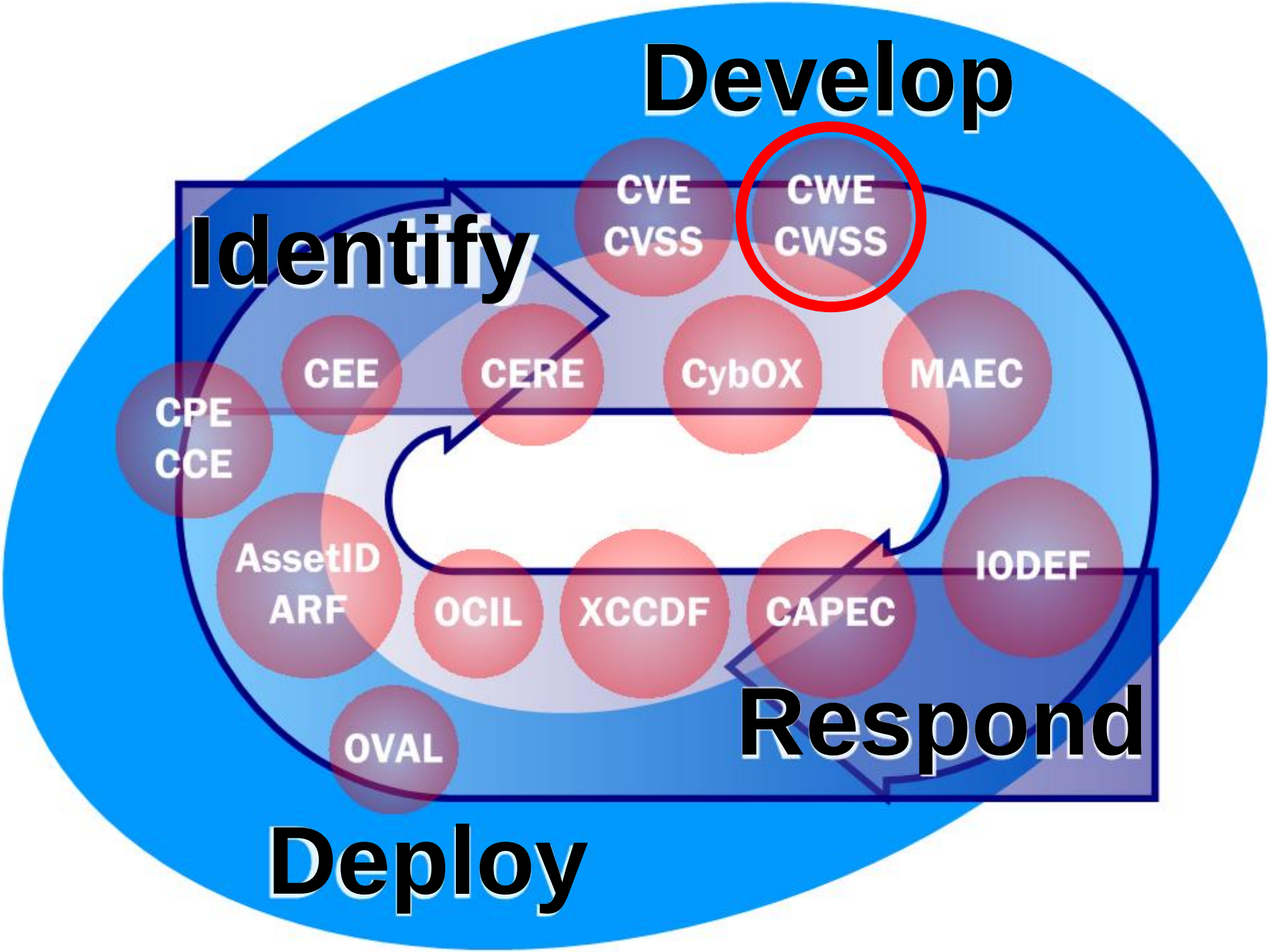
October 31, 2011

Homeland Security

National Institute of Standards and Technology
U.S. Department of Commerce


Mitigating Risk Exposures
Responding to Security Threats
Asset Definition
Configuration Guidance
Vulnerability Alert
Threat Alert
Incident Report
CPE/OVAL
XCCDF/OVAL/ CCE/CCSS
CVE/CWE/OVAL/ CVSS/CWSS
CVE/CWE/CVSS/ CPE/CWSS/ CAPEC/MAEC
CAIF/IDMEF/IODEF/CVE/CWE/ OVAL/CPE/MAEC/CCSS/CWSS/ CEE/ARF
Asset Inventory
Configuration Guidance Analysis
Vulnerability Analysis
Threat Analysis
Intrusion Detection
Incident Management
CPE/ OVAL/ ARF
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
OVAL/XCCDF/ CCE/CCSS/ CPE/ARF
Operations Security Management Processes
System & Software Assurance Guidance/ Requirements
Assessment of System Development, Integration, & Sustainment Activities and Certification & Accreditation
CWE/CAPEC/ SBVR/CWSS/ MAEC
INTERNET
Router
DMZ
Firewall
INTRANET
Web Servers
Application Servers
Database Systems
DNS Server
Mail Server
Web Servers
Desktop Systems
Desktop Systems
Desktop Systems
Desktop Systems
CWE/CAPEC/ CWSS/MAEC/ OVAL/OCIL/X CCDF/CCE/CP E/ARF/SAFES/ 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


Knowledge Repositories
Asset Definition
Configuration Guidance
Vulnerability Alert
Threat Alert
Incident Report
CPE/OVAL
XCCDF/OVAL/ CCE/CCSS
CVE/CWE/OVAL/ CVSS/CWSS
CVE/CWE/CVSS/ CPE/CWSS/ CAPEC/MAEC
CAIF/IDMEF/IODEF/CVE/CWE/ OVAL/CPE/MAEC/CCSS/CWSS/ CEE/ARF
Asset Inventory
Configuration Guidance Analysis
Vulnerability Analysis
Threat Analysis
Intrusion Detection
Incident Management
Respond
Identify
Develop
Deploy
OVAL/XCCDF/ CCE/CCSS/ CPE/ARF
CPE/ OVAL/ ARF
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
System & Software Assurance Guidance/ Requirements
Assessment of System Development, 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

Common Weakness Enumeration

A Community-Developed Dictionary of Software Weakness Types

CWE and SANS Institute TOP 25 MOST DANGEROUS SOFTWARE ERRORS

Search by ID: Go

CWE List
- Full Dictionary View
- Development View
- Research View
- Reports

About
- Sources
- Process
- Documents

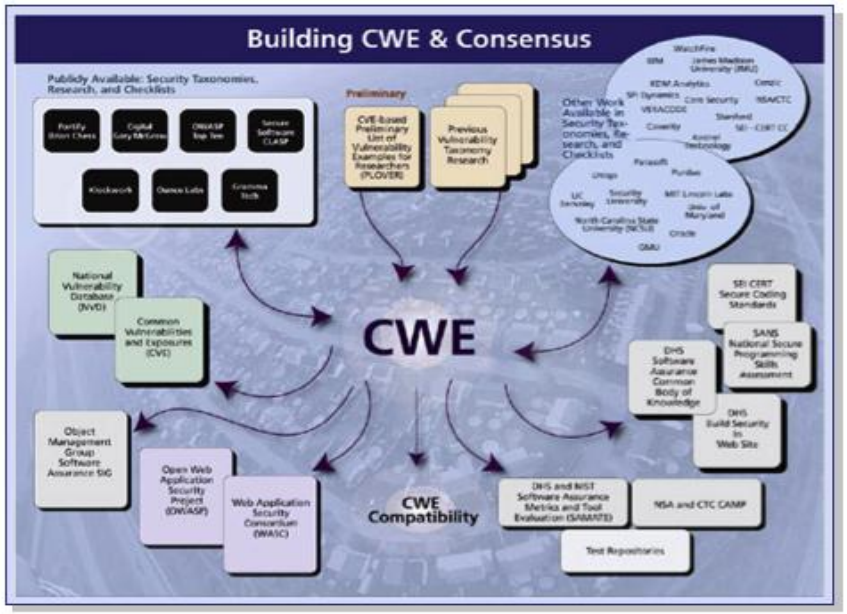
Community
- Related Activities
- Discussion List
- Research
- CWE/SANS Top 25
- CWSS

News
- Calendar
- Free Newsletter

Compatibility
- Program
- Requirements
- Declarations
- Make a Declaration

Contact Us
- Search the Site

International in scope and free for public use, CWE™ provides a unified, measurable set of software weaknesses that is enabling more effective discussion, description, selection, and use of software security tools and services that can find these weaknesses in source code and operational systems as well as better understanding and management of software weaknesses related to architecture and design.



Similar Standards

- Attack Patterns (CAPEC)
- Vulnerabilities (CVE)
- Configurations (CCE)
- Platforms (CPE)
- Malware (MAEC)
- Assessment Language (OVAL)
- Checklist Language (XCCDF)
- Log Format (CEE)
- Security Content Automation (SCAP)
- Making Security Measurable

News

- Updated Common Weakness Scoring System (CWSS) White Paper Now Available
- LDRA Makes Two Declarations of CWE Compatibility
- Software Assurance keynote and Making Security Measurable table booth at *International Conference on Software Quality*
- CWE/Making Security Measurable booth at *Black Hat DC 2011*

...more

Upcoming Events

- CWE/Making Security Measurable booth at *RSA 2011*, February 14-18
- CWE/CAPEC/MAEC briefings at *DHS/DoD/NIST SwA Forum*, February 28 - March 4
- CWE/Making Security Measurable booth at *2011 Information Assurance Symposium*, March 8-10

...more

Status Report

Version 1.11 posted December 13, 2010. 7 new entries were created, mostly related to synchronization and "functionality inclusion." One entry was deprecated. There are changes to 135 entries, especially potential mitigations, names, descriptions, demonstrative examples, and relationships. There were no schema changes.

More Information

cwe@mitre.org

“Making Security Measureable”:
measurablesecurity.mitre.org

Sponsored by DHS with
MITRE as technical lead

Resources provided for
voluntary adoption

Open, community efforts that
are *free* to use

XML-based

Some important things to note

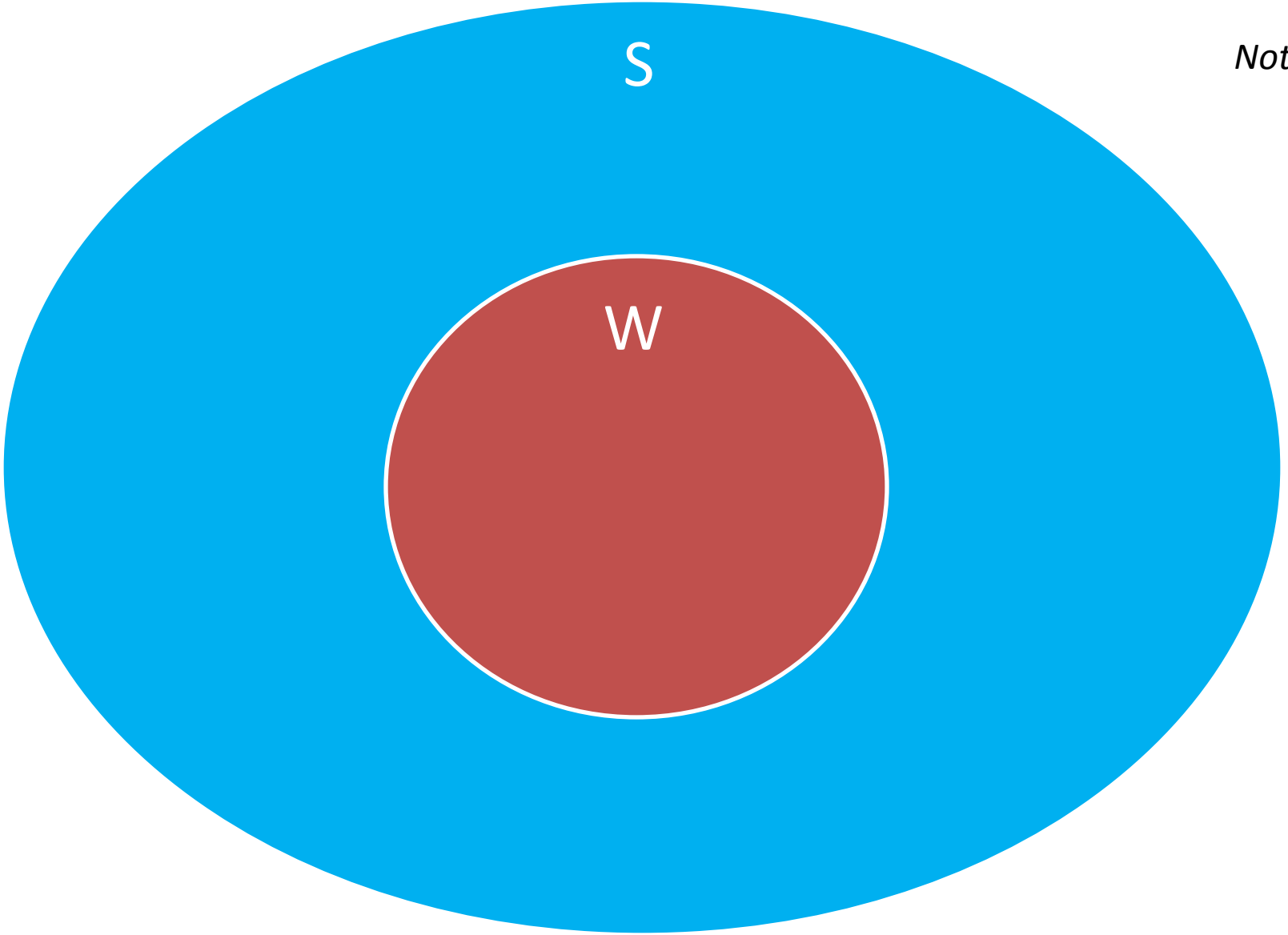
What is the context?

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

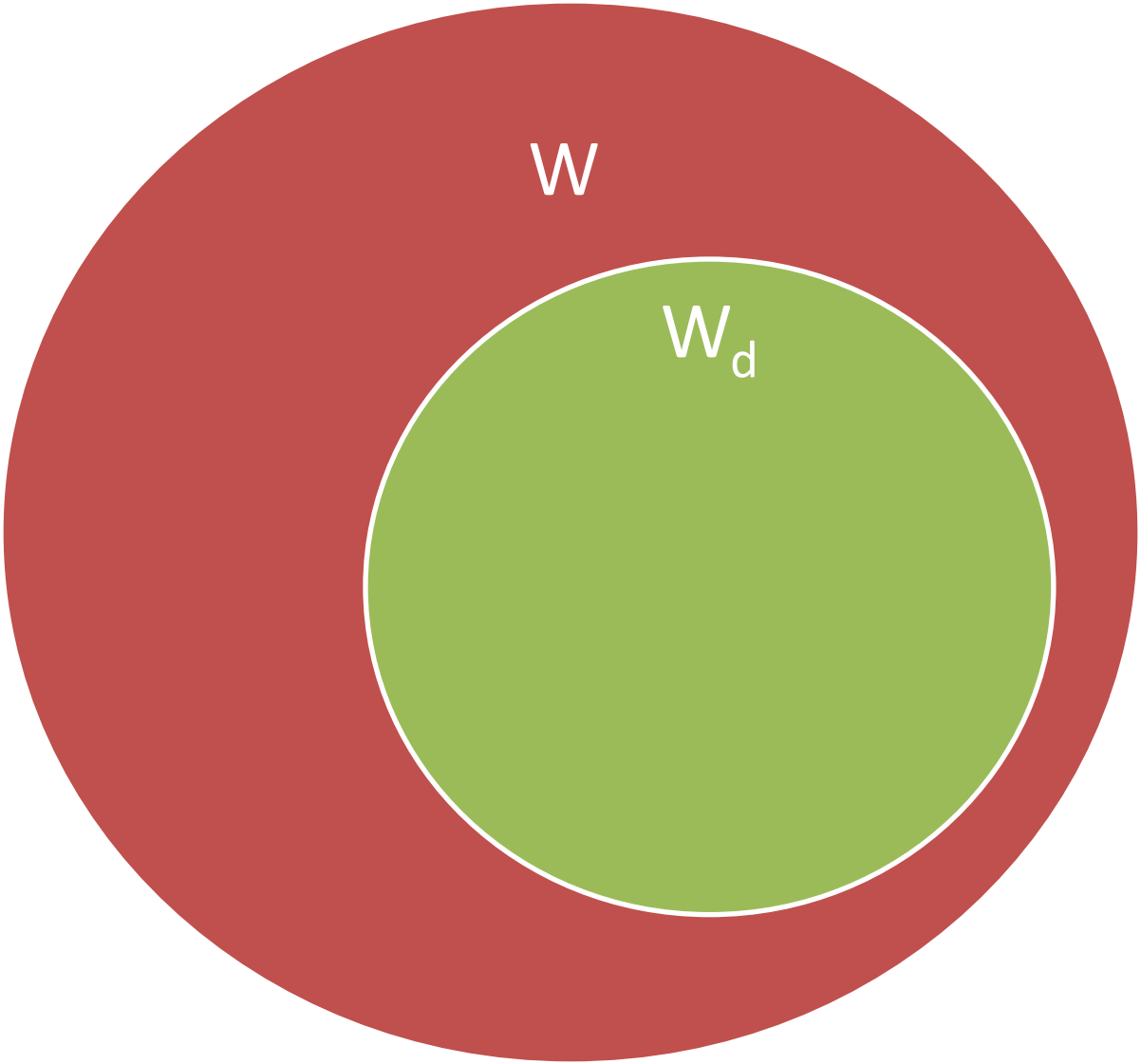
There are many definitions of “weakness.” What do we mean by weakness *in this context*?

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

There are many definitions of “vulnerability.” What do we mean by vulnerability *in this context*?

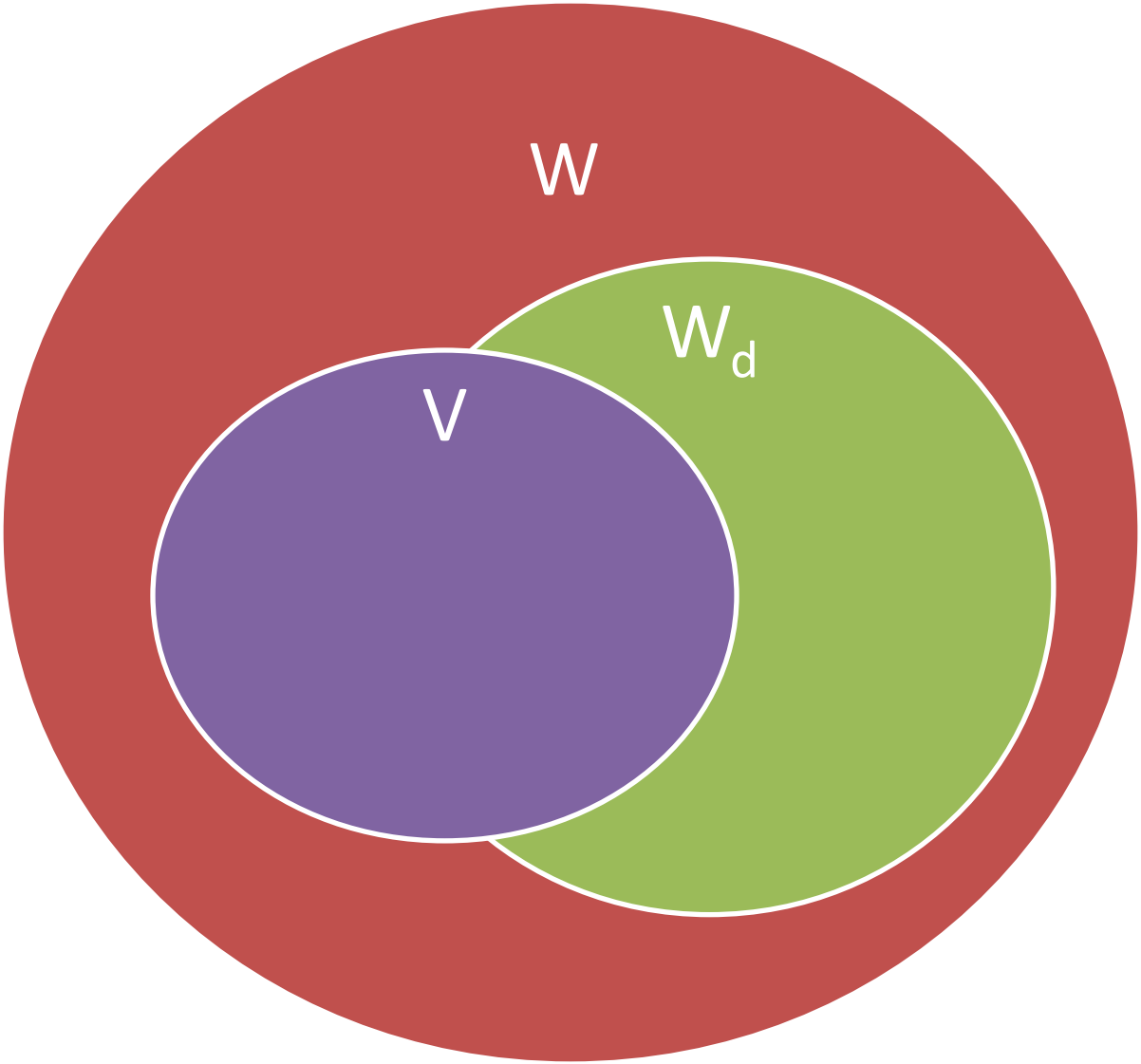
> A *(software) vulnerability* is a collection of one or more weaknesses that contain the right conditions to permit unauthorized parties to force the software to perform unintended behavior (a.k.a. “is exploitable”)

Notional



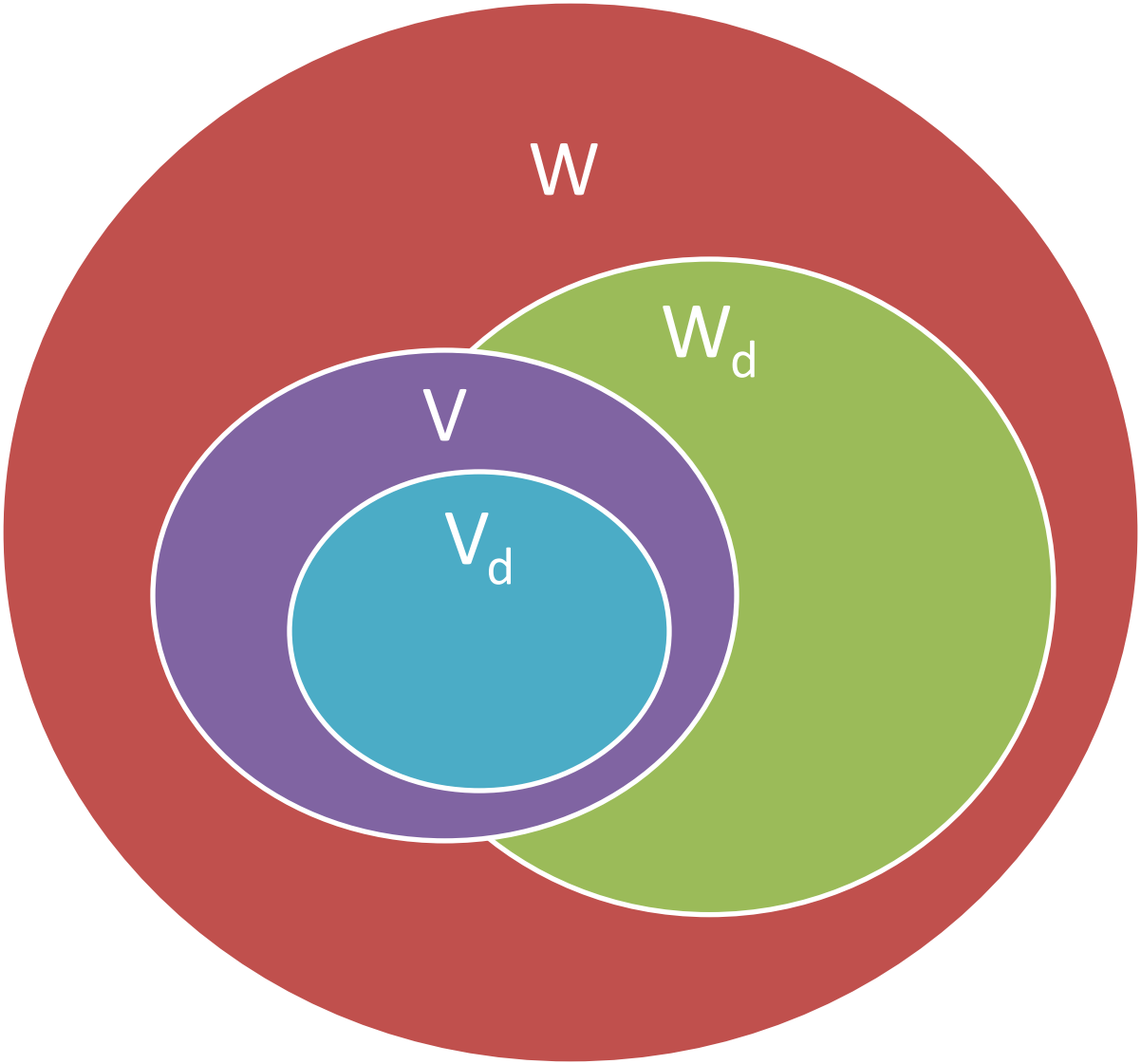
W_d: The set of all *discovered* software weaknesses in W

Notional



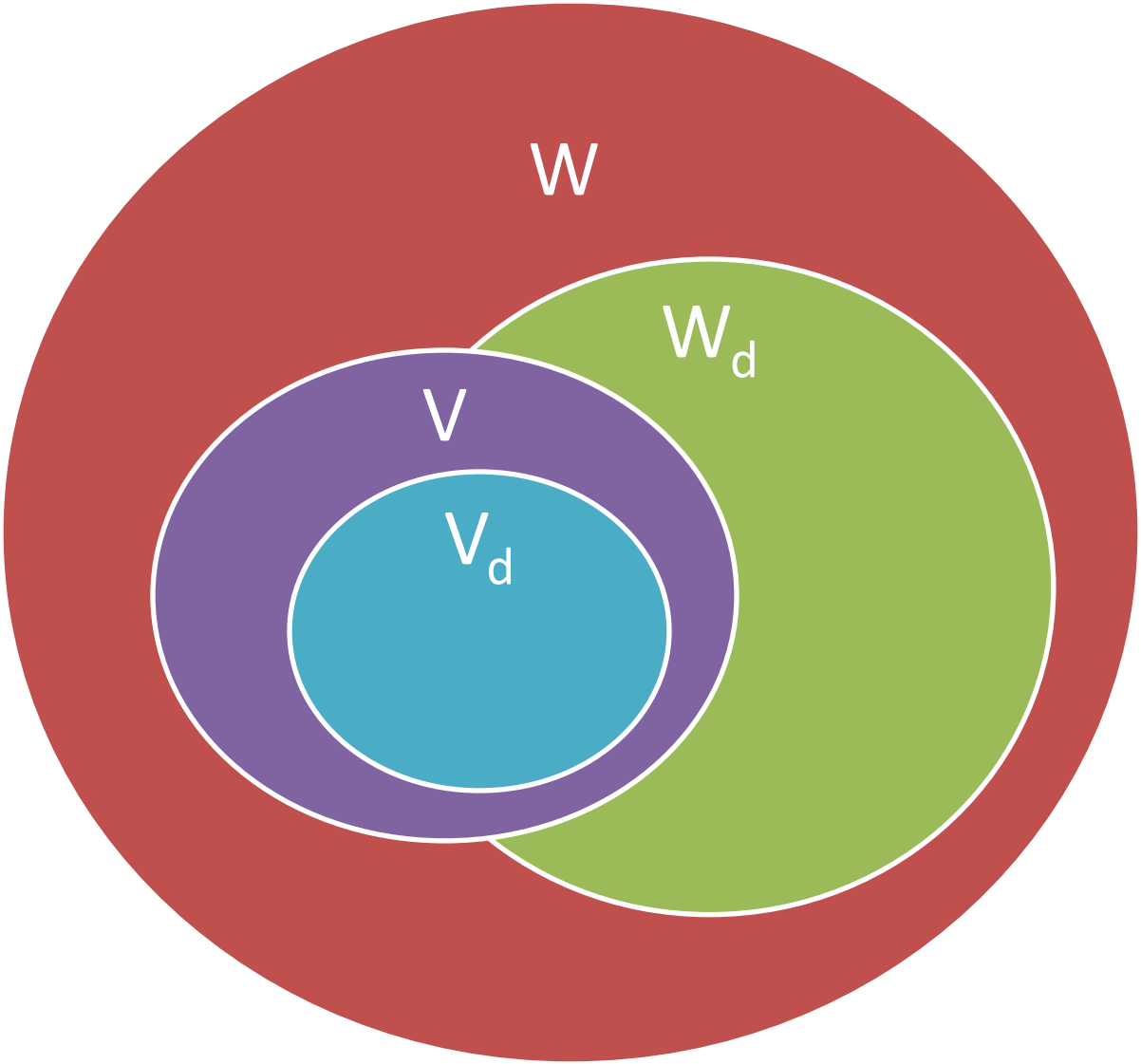
V: The set of all vulnerabilities in W

Notional



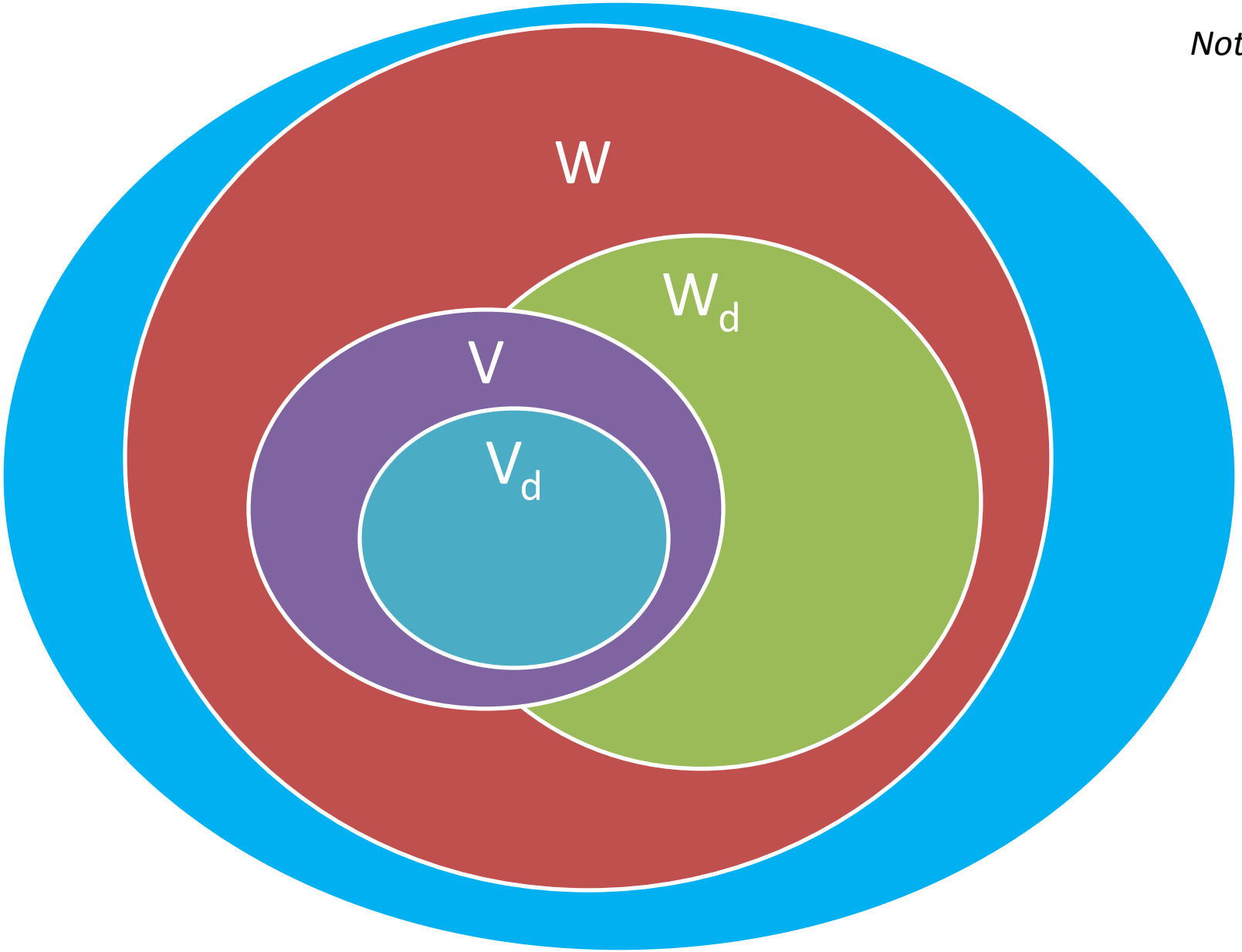
V_d: The set of all *discovered* vulnerabilities in V

Notional



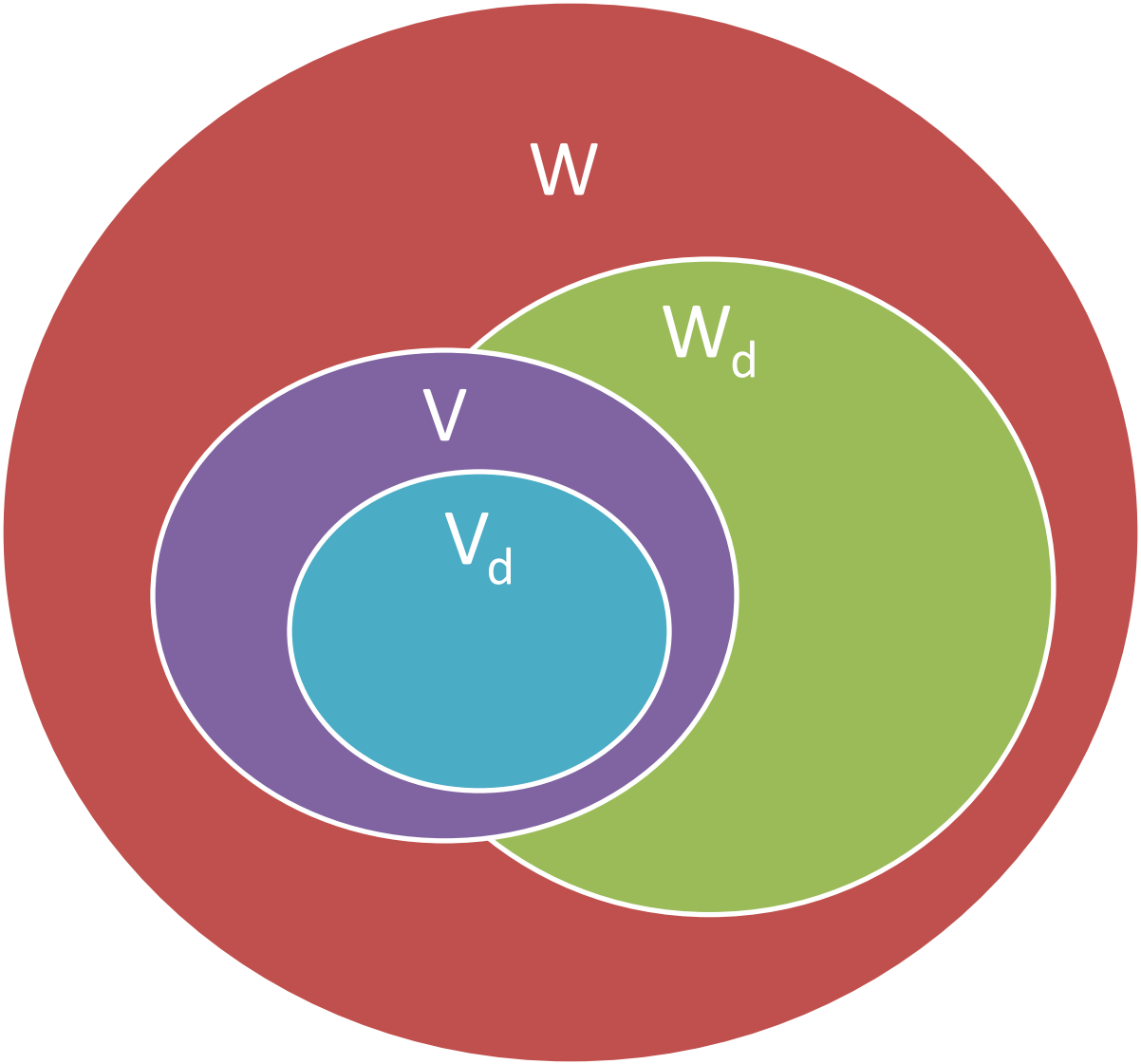
What does the future hold?

Notional



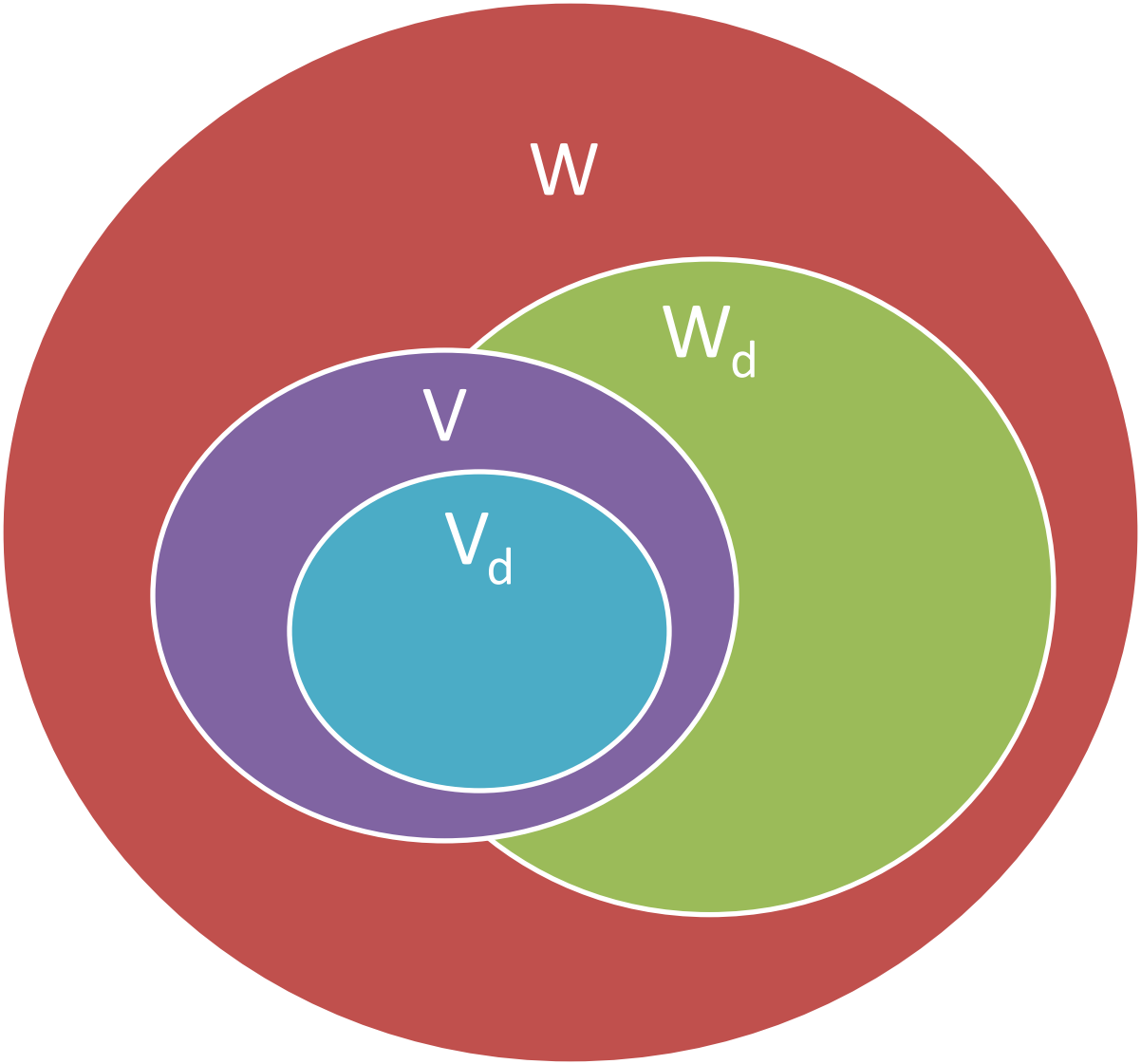
We know it's *not* this, at least not in the near-term

Notional



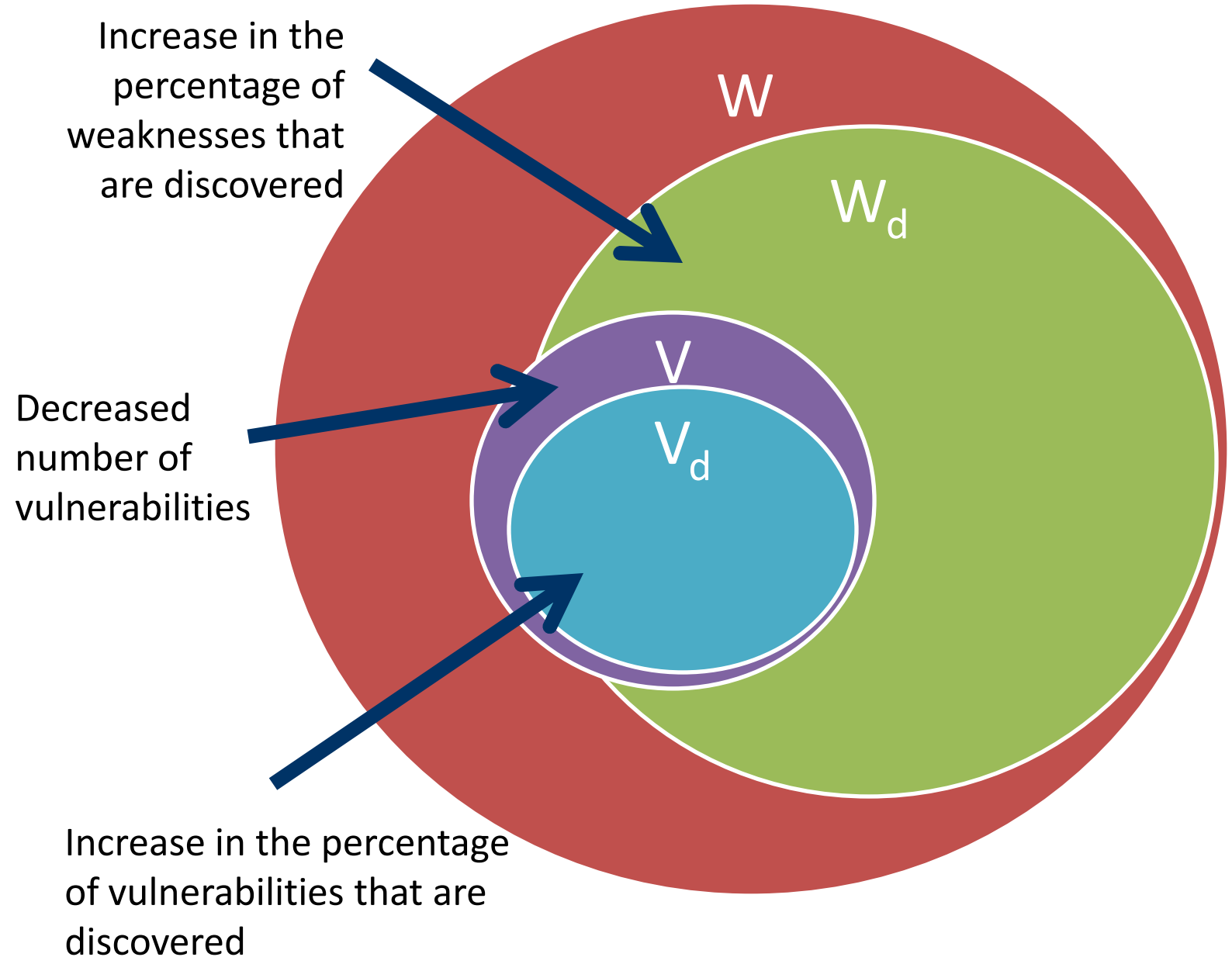
Maybe the problem grows unbounded?

Notional



One reasonable near-term goal

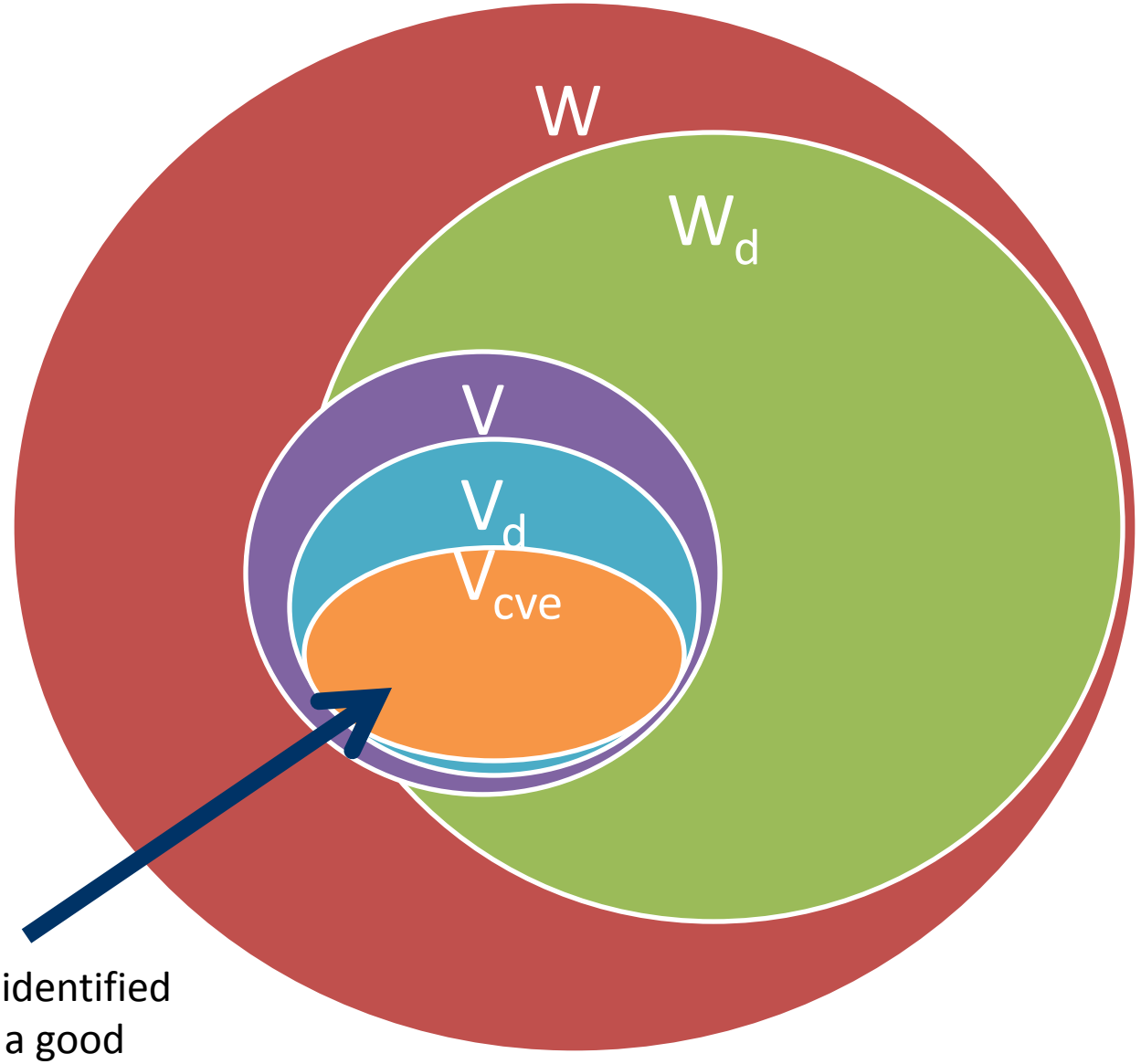
Notional



Is this really better? **Yes**

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

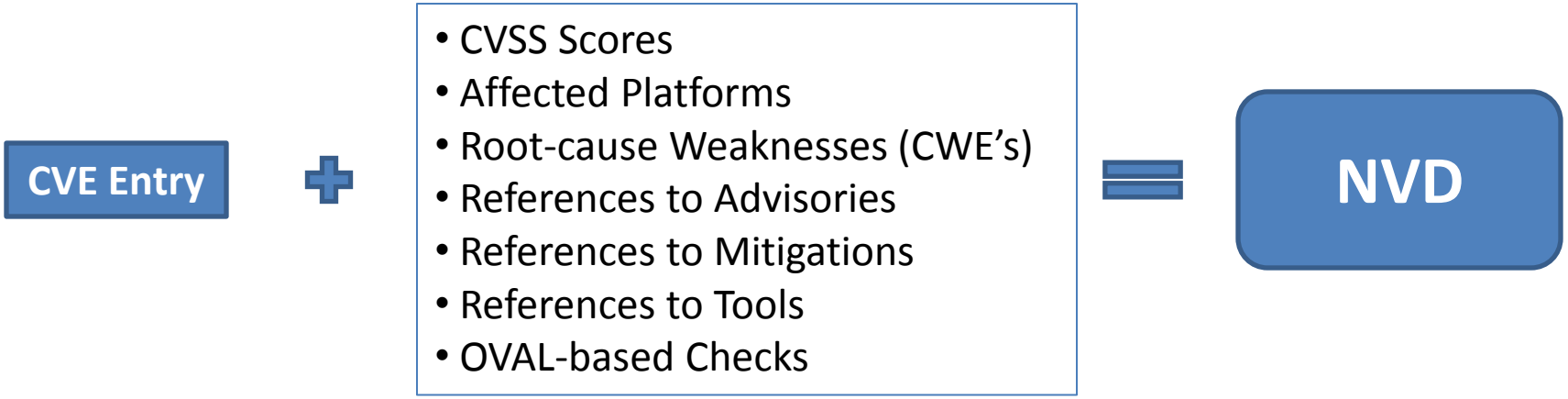
CVE entries feed into *NVD*

assert(CVE != Bug_Database);

47,258 entries (as of last week)

Common Vulnerabilities and Exposures (CVE)

National Vulnerability Database (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

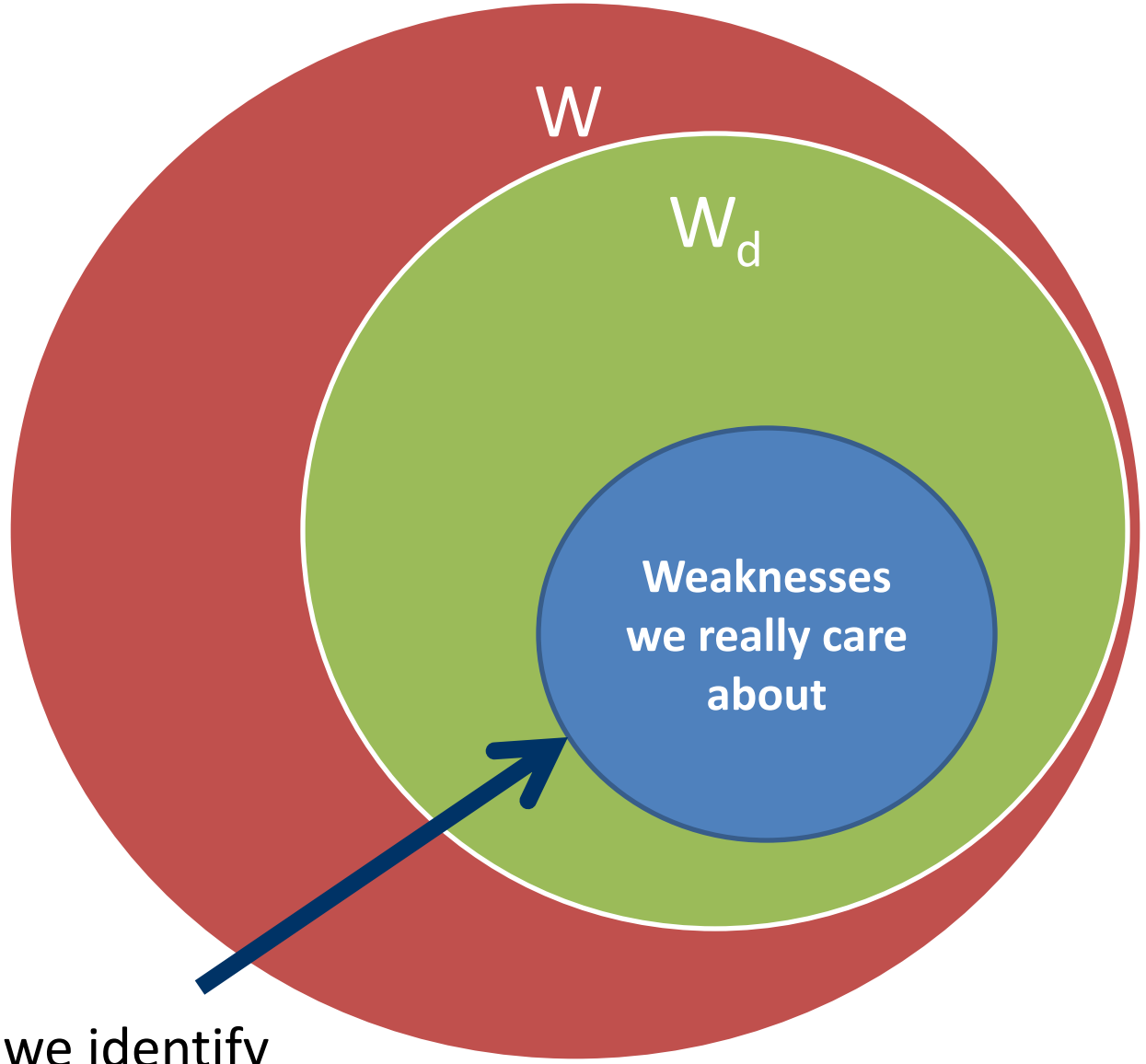
Plus much, much more

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

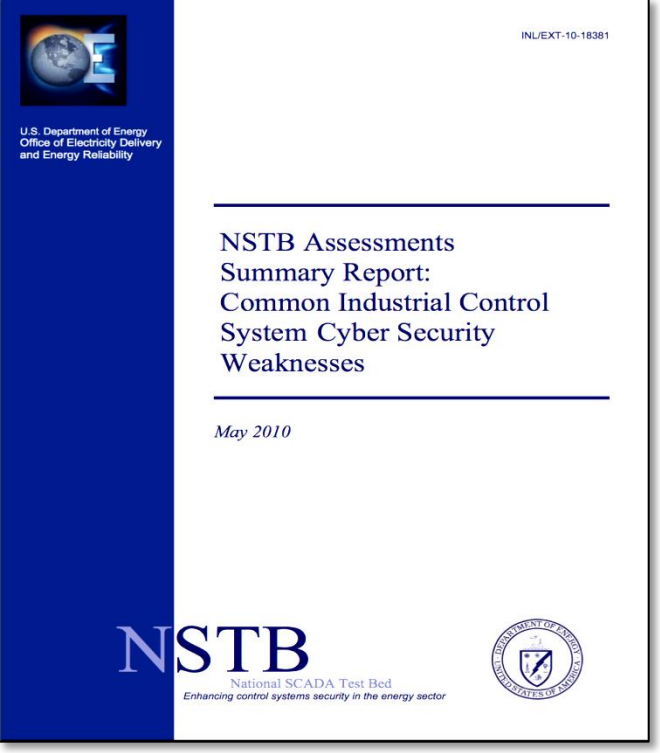
OWASP Top 10

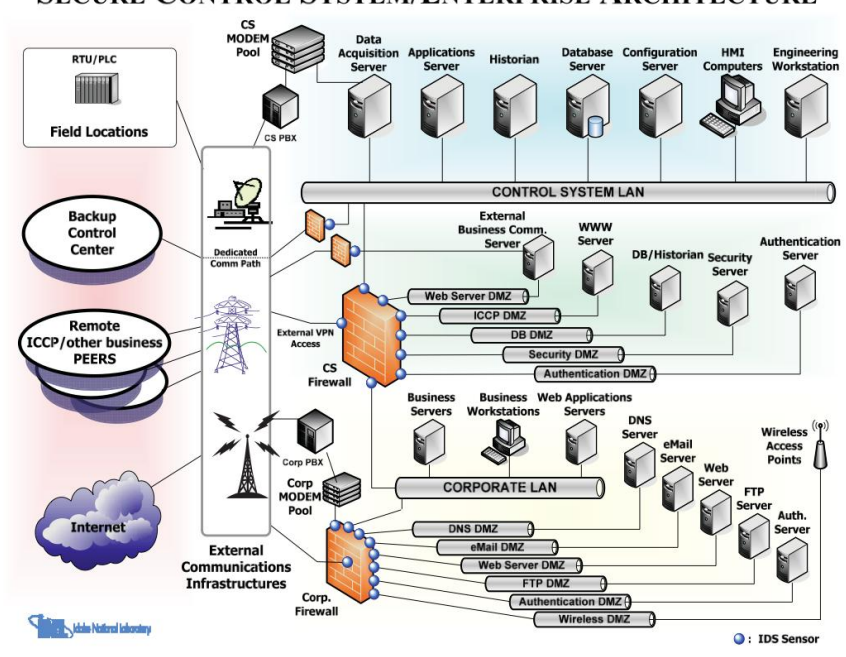
CWE/SANS Top 25

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

Idaho National Labs SCADA Report





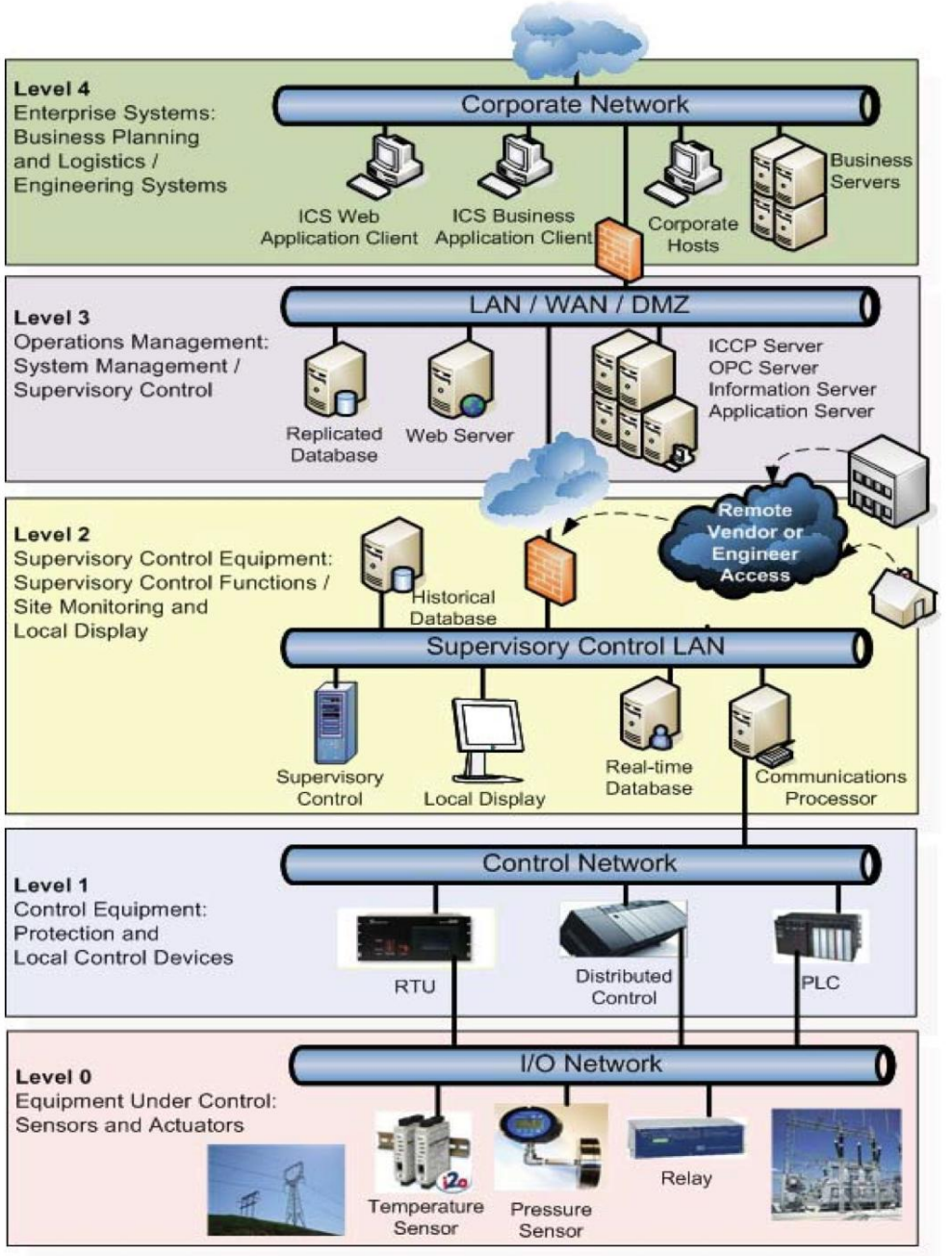


Table 27. Most common programming errors found in ICS code.

Weakness Classification	Vulnerability Type
CWE-19: Data Handling	CWE-228: Improper Handling of Syntactically Invalid Structure
	CWE-229: Improper Handling of Values
	CWE-230: Improper Handling of Missing Values
	CWE-20: Improper Input Validation
	CWE-116: Improper Encoding or Escaping of Output
	CWE-195: Signed to Unsigned Conversion Error
	CWE-198: Use of Incorrect Byte Ordering
CWE-119: Failure to Constrain Operations within the Bounds of a Memory Buffer	CWE-120: Buffer Copy without Checking Size of Input ("Classic Buffer Overflow")
	CWE-121: Stack-based Buffer Overflow
	CWE-122: Heap-based Buffer Overflow
	CWE-125: Out-of-bounds Read
	CWE-129: Improper Validation of Array Index
	CWE-131: Incorrect Calculation of Buffer Size
	CWE-170: Improper Null Termination
	CWE-190: Integer Overflow or Wraparound
	CWE-680: Integer Overflow to Buffer Overflow
CWE-398: Indicator of Poor Code Quality	CWE-454: External Initialization of Trusted Variables or Data Stores
	CWE-456: Missing Initialization
	CWE-457: Use of Uninitialized Variable
	CWE-476: NULL Pointer Dereference
	CWE-400: Uncontrolled Resource Consumption ("Resource Exhaustion")
	CWE-252: Unchecked Return Value
	CWE-690: Unchecked Return Value to NULL Pointer Dereference
	CWE-772: Missing Release of Resource after Effective Lifetime
CWE-442: Web Problems	CWE-22: Improper Limitation of a Pathname to a Restricted Directory ("Path Traversal")
	CWE-79: Failure to Preserve Web Page Structure ("Cross-site Scripting")
	CWE-89: Failure to Preserve SQL Query Structure ("SQL Injection")
CWE-703: Failure to Handle Exceptional Conditions	CWE-431: Missing Handler
	CWE-248: Uncaught Exception
	CWE-755: Improper Handling of Exceptional Conditions
	CWE-390: Detection of Error Condition Without Action

Common Weakness Risk Analysis Framework (CWRAF)

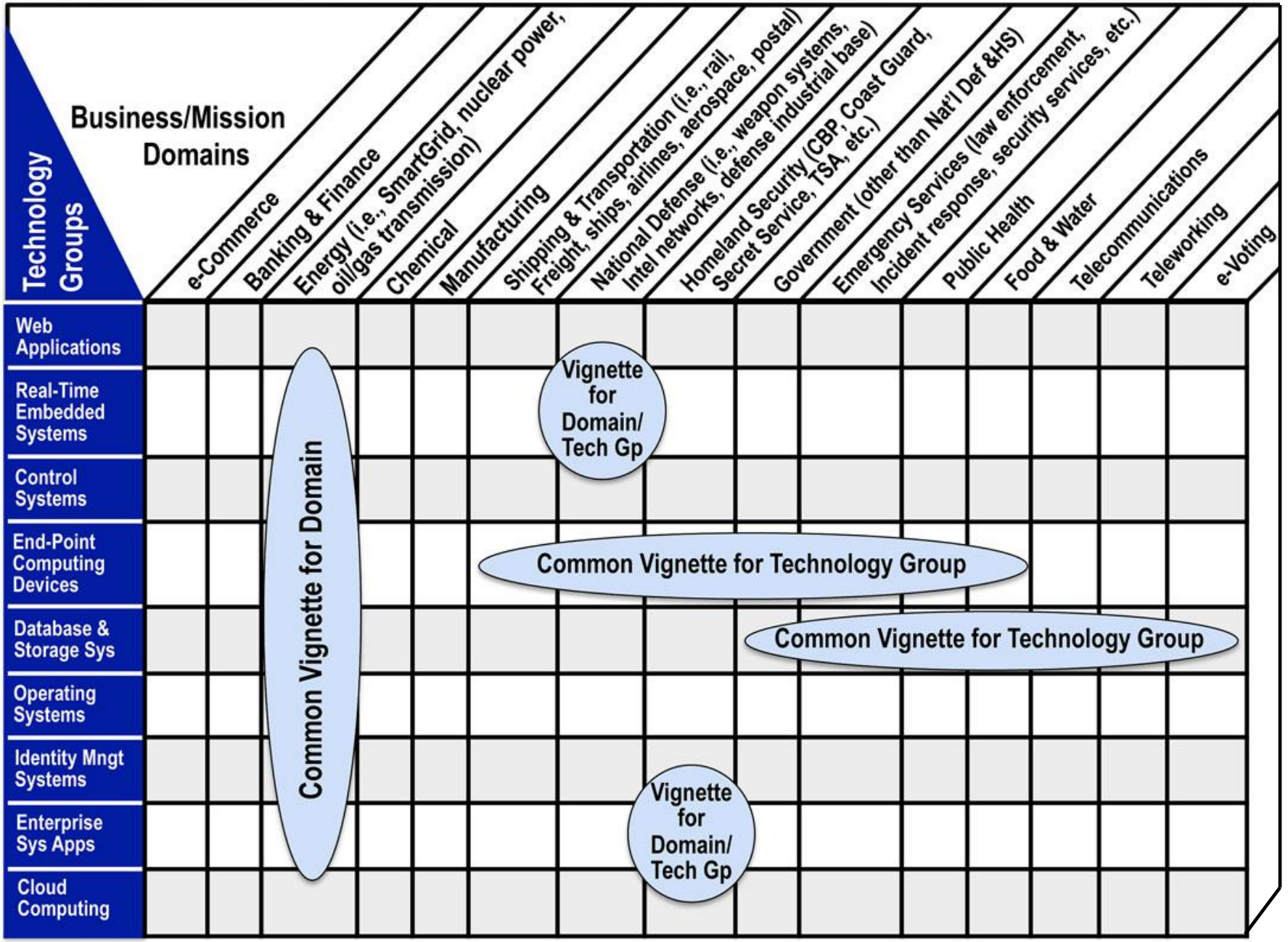
*How do I **identify** which of the 800+ CWE's are most important for my specific business domain, technologies and environment?*

Common Weakness Scoring System (CWSS)

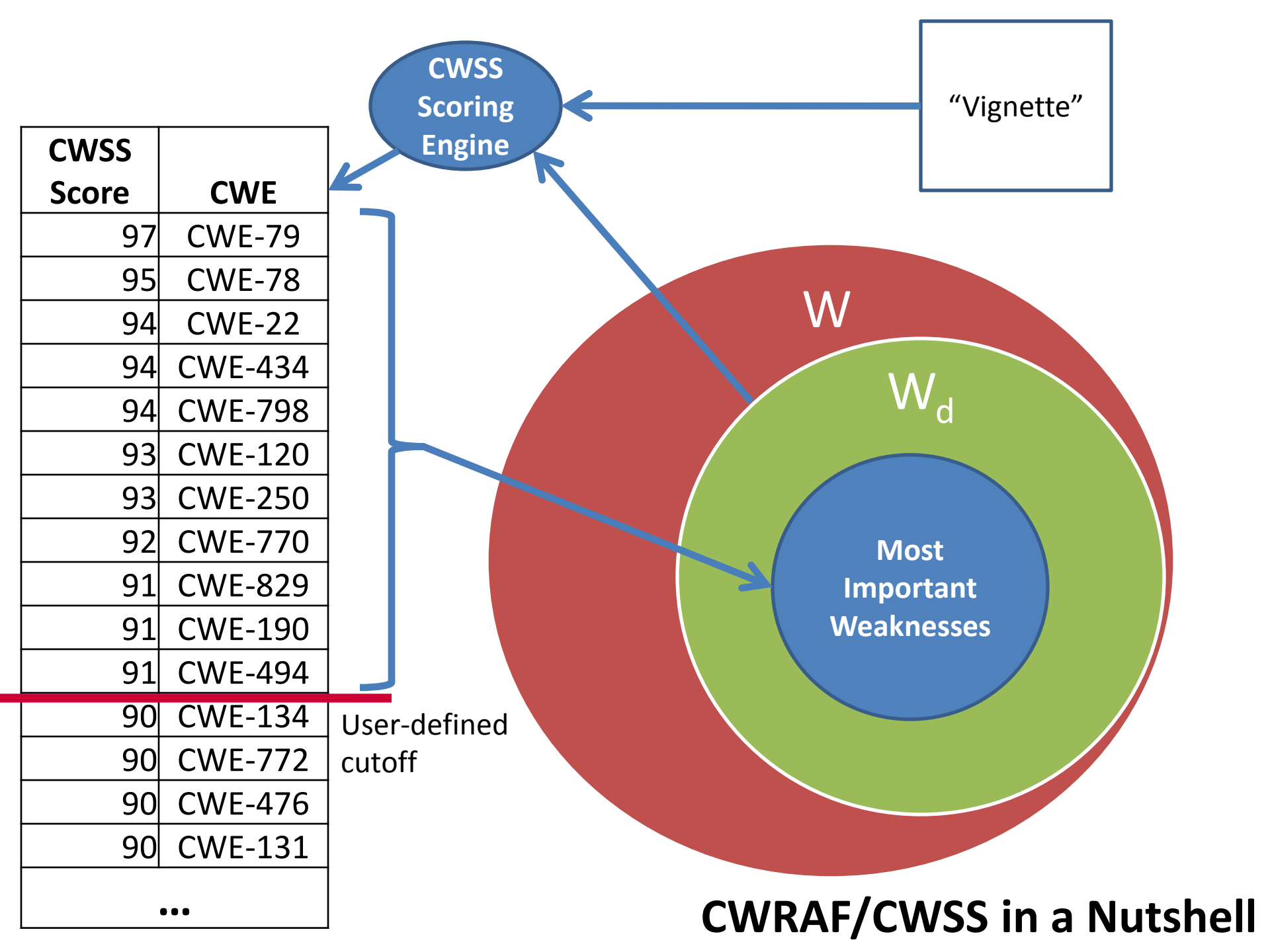
*How do I **rank** the CWE's I care about according to my specific business domain, technologies and environment?*

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

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



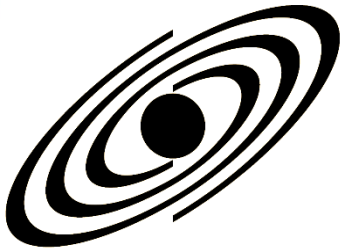
Common Weakness Risk Assessment Framework uses Vignettes with Archetypes to identify top CWEs in respective Domain/Technology Groups


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
...
User-defined cutoff
CWSS Scoring Engine
"Vignette"
W
W_d
Most Important Weaknesses
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

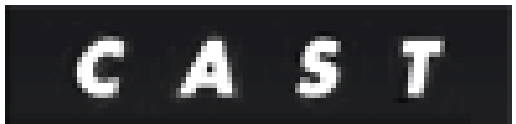
EC-Council

OWASP
The Open Web Application Security Project

SAIC®

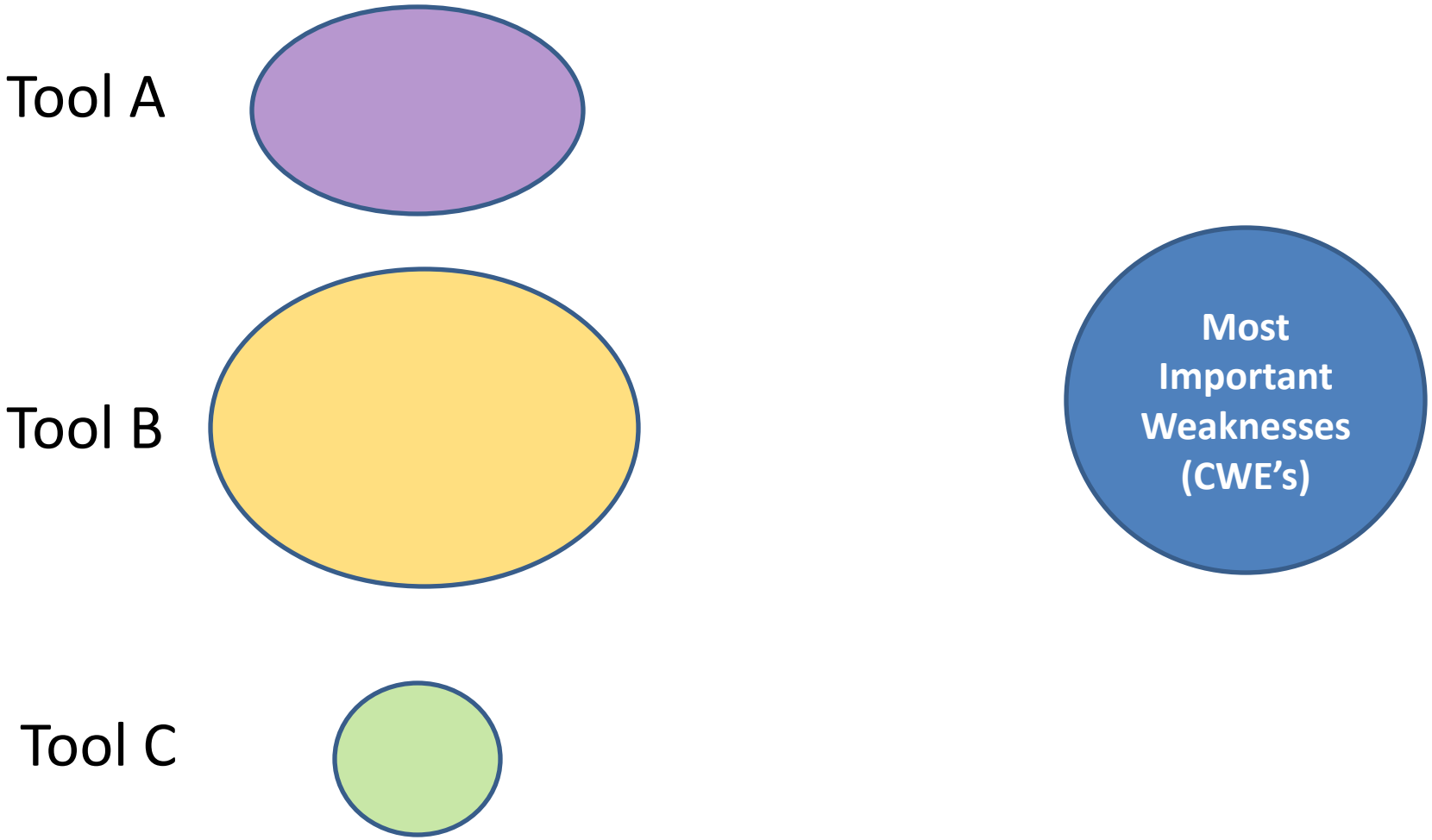
Common Weakness Risk Analysis Framework (CWRAF) and Common Weakness Scoring System (CWSS)

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.



CWE Coverage Claims Representation (CCR)

Set of CWE's tool *claims* to cover



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 on 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.

Common Attack Pattern Enumeration and Classification (CAPEC)

Dictionary of attack types (mostly software)

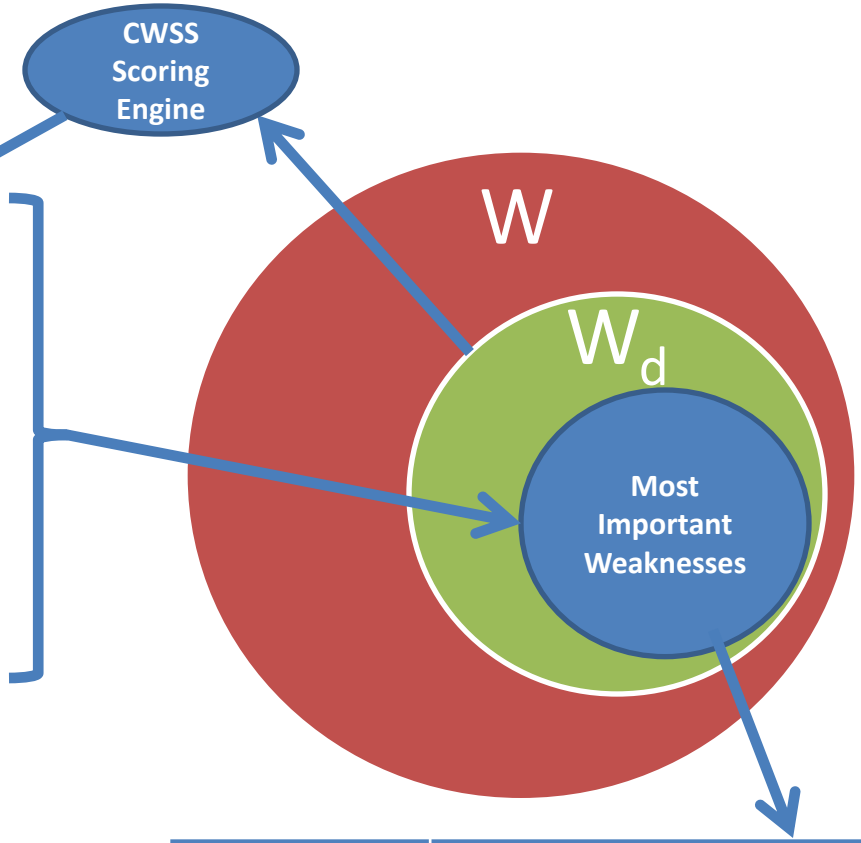
- CAPEC ID
- Name
- Description
- Attack Prerequisites
- Indicators of Attack
- Examples
- **Related Weaknesses (CWE's)**
- Mitigations

Plus much, much more

386 patterns, organized by categories, with views

What types of attacks should I test my system against?

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
...	



CWE	Related CAPEC ID's
CWE-79	CAPEC-232, CAPEC-106, CAPEC-19, …
CWE-78	CAPEC-108, CAPEC-15, CAPEC-43, CAPEC-6, …
…	…

Common Attack Pattern Enumeration and Classification

automation can help…

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

- 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**)

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

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

It can be used to begin the translation of SwA Activities across organizational leadership

COO

CEO

Business Functions

Define Business Goals

Development Organization

Prioritize funds and manage risks

CFO

CIO

Sustained environment to achieve business goals through technology

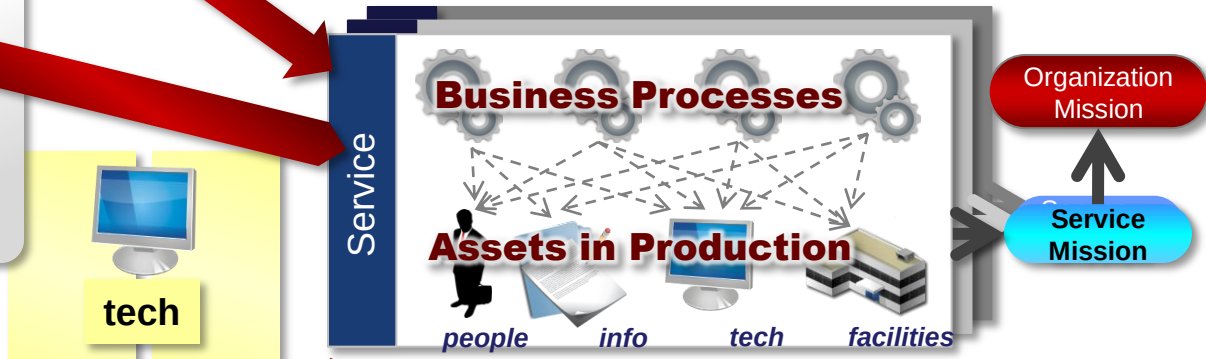
Enterprise Assurance Support

CTO

Enable Resilient Technology

Development Project

Development Engineering

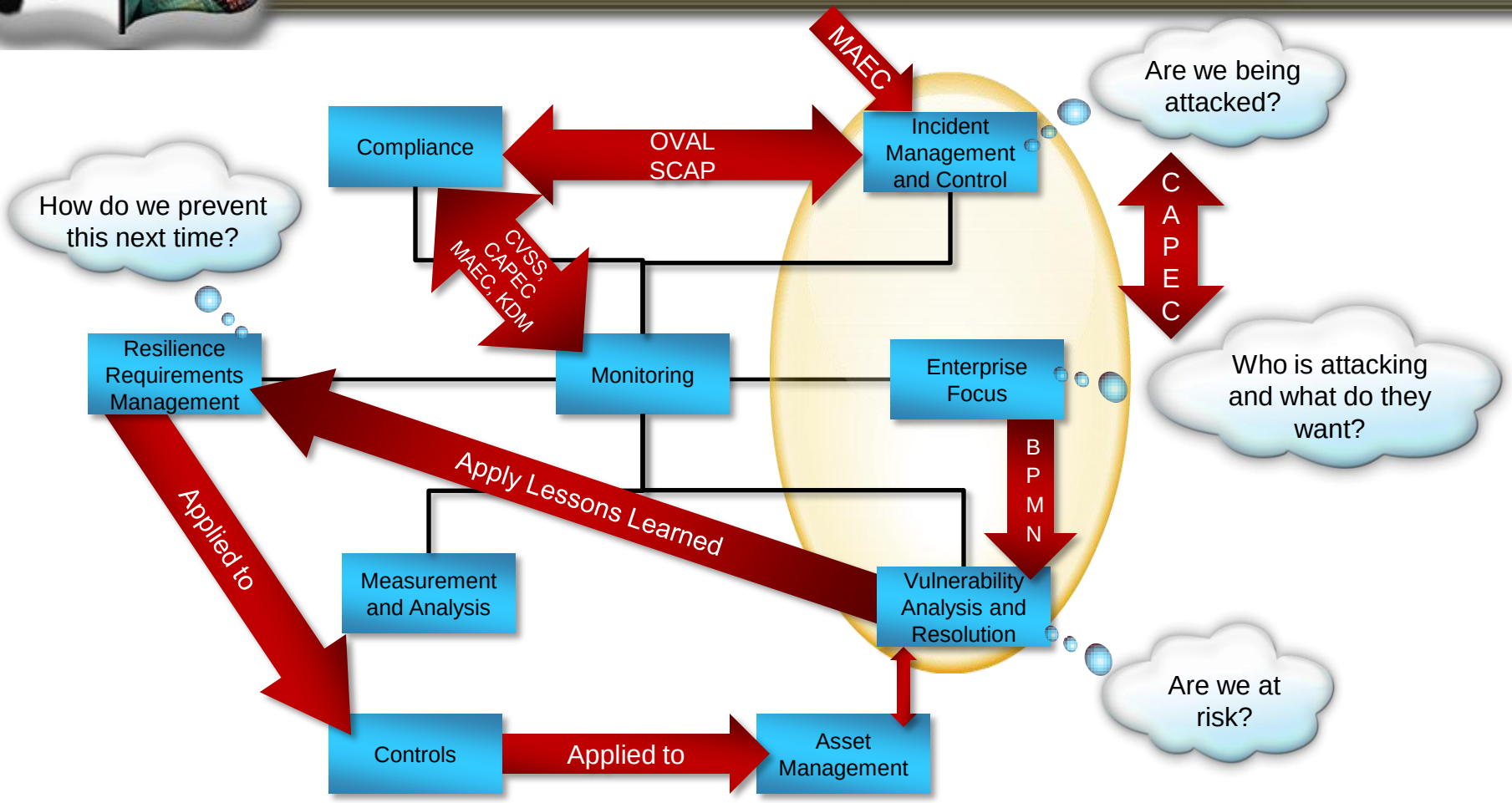


Adapted from: Source: November 2009 SwA Forum- Evolution in SwA Processes Panel – David White, SEI

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

SwA and Operational Resilience



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

Courtesy of Michele Moss

Software Assurance Ecosystem: The Formal Framework

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

Process Docs & Artifacts

Requirements/Design Docs & Artifacts

Reports
Risk Analysis, etc)

Process, People & Documentation Evaluation Environment

- Some point tools to assist evaluators but mainly manual work
- Claims in Formal SBVR vocabulary
- Evidence in Formal SBVR vocabulary
- Large scope requires large effort

Process, People, documentation Evidence

Formalized Specifications

Claims, Arguments and Evidence Repository

- Formalized in SBVR vocabulary
- Automated verification of claims against evidence
- Highly automated and sophisticated risk assessments using transitive inter-evidence point relationships

Software System / Architecture Evaluation

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

Software system Technical Evidence

Executable Specifications

Software System Artifacts

Hardware Environment

Protection Profiles

IA Controls

CWE

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

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

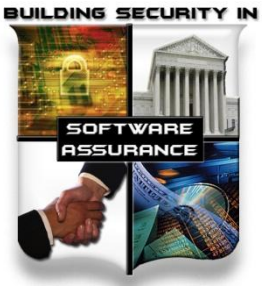
SwA Websites: www.us-cert.gov/swa

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

Homeland Security

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.

Next 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

Homeland Security

SOFTWARE ASSURANCE FORUM

Homeland Security

Commerce

National Defense

BUILDING SECURITY IN

Public/Private Collaboration Efforts for Software Supply Chain Risk Management

Next SwA Forum meets 28 Nov – 2 Dec 2011 at MITRE, McLean, VA

SOFTWARE ASSURANCE FORUM

Homeland Security

Commerce

National Defense

BUILDING SECURITY IN

Public/Private Collaboration Efforts for Software Supply Chain Risk Management

Next SwA Forum meets 12-16 Sep 2011 at SEI, Arlington, VA