Software Assurance (SwA) in Education, Training & Certification

Pocket Guide v2.1

Robin A. Gandhi

Nebraska University Center on Information Assurance (NUCIA)
University of Nebraska at Omaha

What is a Pocketguide?

- Self-contained
- Concise
- Enumeration of resources
- Theme
- Living document
- Reprints and redistribution possible
- Fits in the coat pocket

SwA ETC Pocketguide Theme

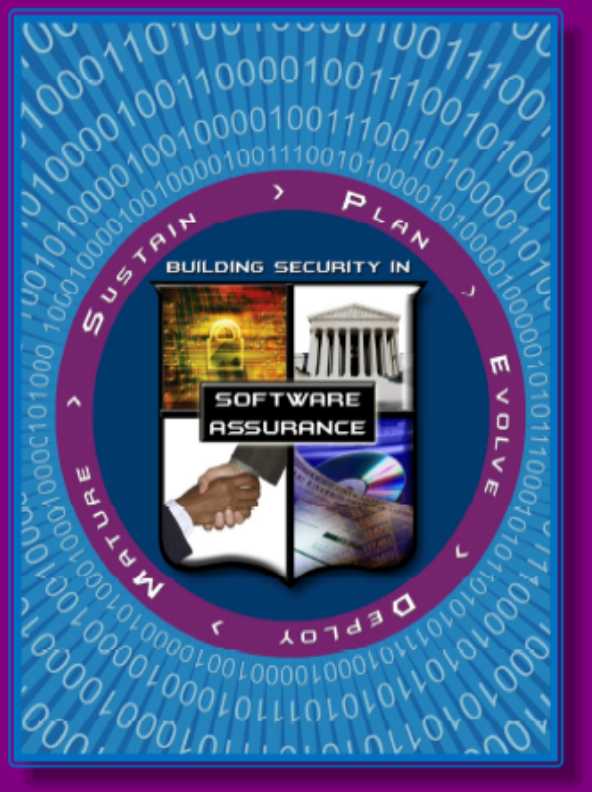
- Educating the Educator/Trainer on available SwA resources

- Purpose:
  - Awareness resource for “getting started” in educating, training and sustaining a workforce capable of producing secure software
  - An “**index**” in to a vast amount of resources, tools, curricula, and certification and training opportunities for software assurance

Purple, v 2.1, March 2011

Software Assurance in Education, Training & Certification

Software Assurance Pocket Guide Series:
Life Cycle Support, Volume I
Version 2.1, March 1, 2011



Software Assurance (SwA) Pocket Guide Resources

This is a resource for 'getting started' in educating, training and certifying a workforce to build secure software. It describes how to promote awareness of the engineering activities and knowledge areas needed to build software that operates as expected, free from vulnerabilities. It summarizes how to train to prevent vulnerabilities from being intentionally designed into the software or accidentally inserted at any time during its life cycle. To do so, this guide describes knowledge areas for software assurance, starting with the core areas of study and extending to sub-disciplines to enhance with software security subject materials. It then presents lists of resources for accomplishing such study, including programs, tools, and books, with pointers on their use. Lastly, this guide describes the people who make up a security-conscious system development team, their education, titles, credentials, and standards. As part of the Software Assurance (SwA) Pocket Guide series, this resource is for information only. For details, see referenced source documents. For proper attribution, please include mention of these sources when referencing any part of this document

This volume of the SwA Pocket Guide series focuses on enumerating education, training and certification resources. It identifies the most effective strategies to inject software assurance topics into existing college curriculums and workforce training and certification programs.

At the back of this pocket guide are references, limitation statements, and a listing of topics addressed in the SwA Pocket Guide series. All SwA Pocket Guides and SwA-related documents are freely available for download via the SwA Community Resources and Information Clearinghouse at https://buildsecurityin.us-cert.gov/swa.

Acknowledgements

The SwA community collaborates to develop SwA Pocket Guides. The SwA Forum and Working Groups function as a stakeholder meta-community that welcomes additional participation in advancing and refining software security. All SwA-related information resources are offered free for public use. The SwA community invites your input: please contact Software.Assurance@dhs.gov for comments and inquiries. For the most current pocket guides, refer to the SwA community website at https://buildsecurityin.us-cert.gov/swa/.

Members from government, industry, and academia comprise the SwA Forum and Working Groups. The Groups focus on incorporating SwA considerations into acquisition and development processes to manage potential risk exposure from software and from the supply chain.

Participants in the SwA Forum's Workforce Education and Training Working Group contributed to developing the material used in this pocket guide as a step in raising awareness on how to incorporate SwA topics in education, training and certification of a knowledgeable workforce. One that is ready to perform engineering or technical activities that promote software assurance throughout the Software Development Life Cycle (SDLC).

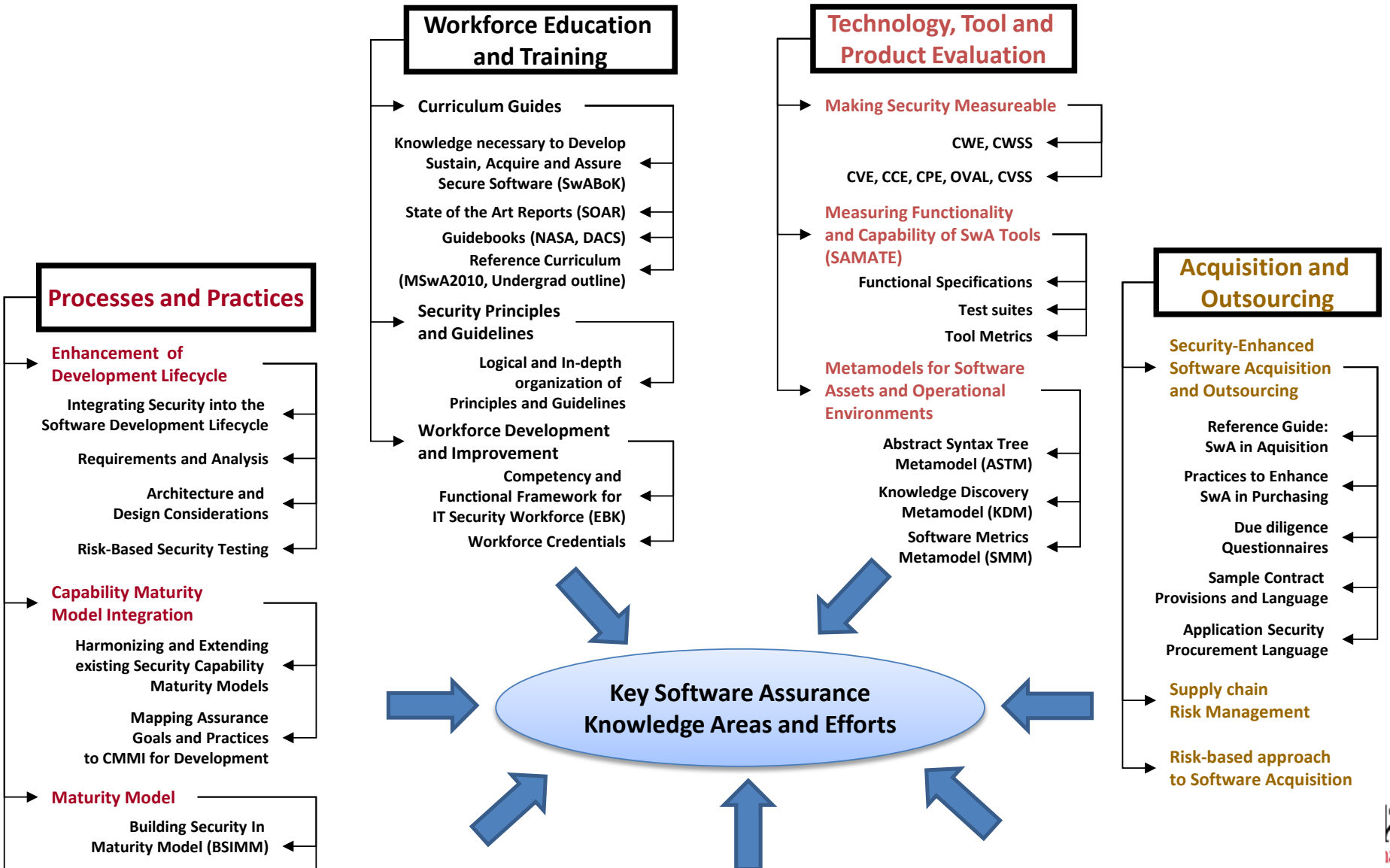
Software Assurance?

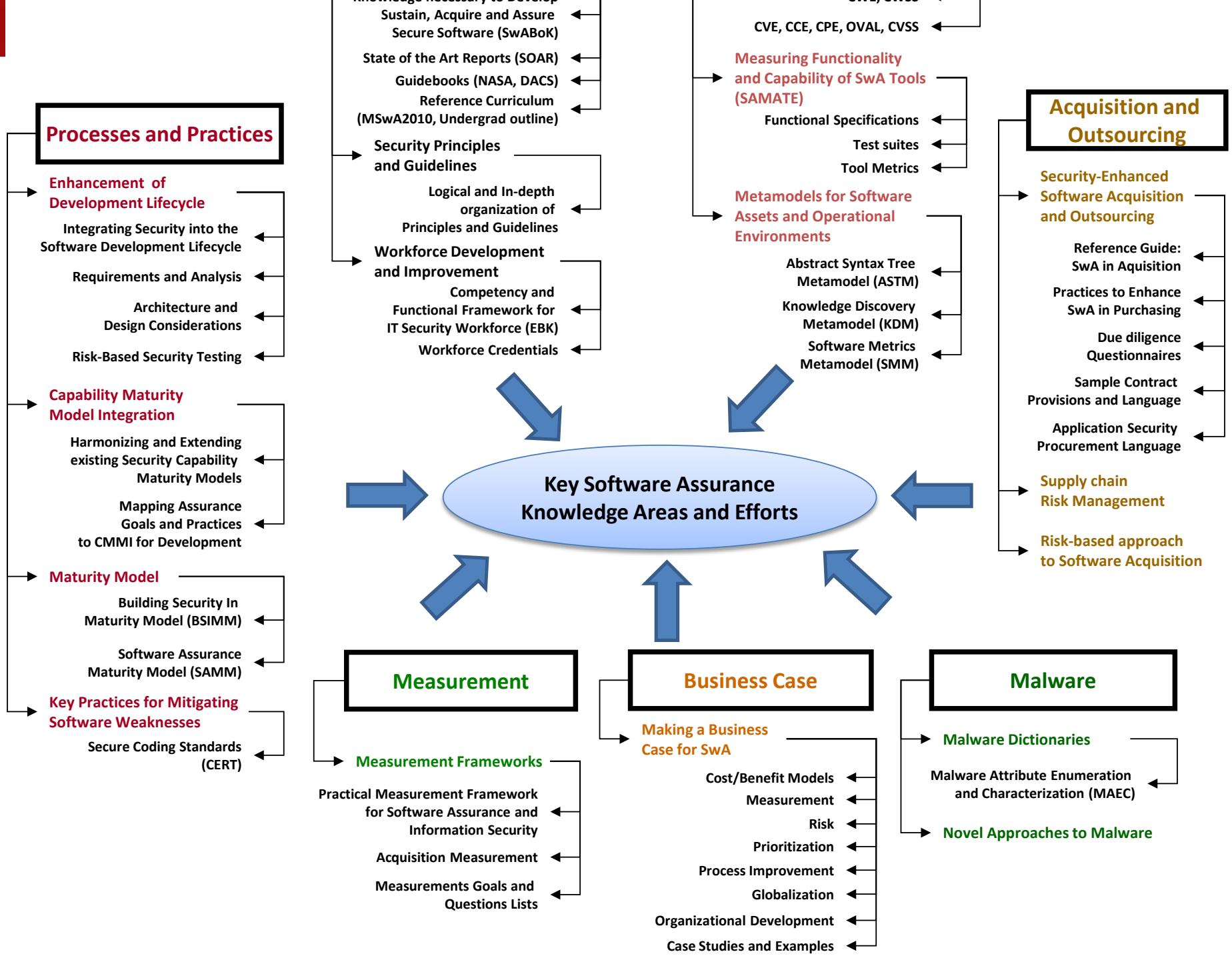
- The *basis for the belief* that software will work as expected
  - Claims, arguments, evidences that span the software lifecycle from cradle to grave
  - People, Process, Technology that enable us to promote assurances in the software that is mission and business critical

Table of Contents

SwA Knowledge Areas and Efforts

SwA Knowledge Areas and Efforts




Sustain, Acquire and Assure Secure Software (SwABoK)
State of the Art Reports (SOAR)
Guidebooks (NASA, DACS)
Reference Curriculum (MSwA2010, Undergrad outline)
Security Principles and Guidelines
Logical and In-depth organization of Principles and Guidelines
Workforce Development and Improvement
Competency and Functional Framework for IT Security Workforce (EBK)
Workforce Credentials
CVE, CCE, CPE, OVAL, CVSS
Measuring Functionality and Capability of SwA Tools (SAMATE)
Functional Specifications
Test suites
Tool Metrics
Metamodels for Software Assets and Operational Environments
Abstract Syntax Tree Metamodel (ASTM)
Knowledge Discovery Metamodel (KDM)
Software Metrics Metamodel (SMM)
Processes and Practices
Enhancement of Development Lifecycle
Integrating Security into the Software Development Lifecycle
Requirements and Analysis
Architecture and Design Considerations
Risk-Based Security Testing
Capability Maturity Model Integration
Harmonizing and Extending existing Security Capability Maturity Models
Mapping Assurance Goals and Practices to CMMI for Development
Maturity Model
Building Security In Maturity Model (BSIMM)
Software Assurance Maturity Model (SAMM)
Key Practices for Mitigating Software Weaknesses
Secure Coding Standards (CERT)
Acquisition and Outsourcing
Security-Enhanced Software Acquisition and Outsourcing
Reference Guide: SwA in Aquisition
Practices to Enhance SwA in Purchasing
Due diligence Questionnaires
Sample Contract Provisions and Language
Application Security Procurement Language
Supply chain Risk Management
Risk-based approach to Software Acquisition
Key Software Assurance Knowledge Areas and Efforts
Measurement
Measurement Frameworks
Practical Measurement Framework for Software Assurance and Information Security
Acquisition Measurement
Measurements Goals and Questions Lists
Business Case
Making a Business Case for SwA
Cost/Benefit Models
Measurement
Risk
Prioritization
Process Improvement
Globalization
Organizational Development
Case Studies and Examples
Malware
Malware Dictionaries
Malware Attribute Enumeration and Characterization (MAEC)
Novel Approaches to Malware

Curriculum and Training Guides

Table 1– SwA Curriculum and Training Development Guides		
Identifier	**Relevant Documents and Links**	**Purpose**
SwA Curriculum Project[1]	Volume I: Master of Software Assurance Reference Curriculum. Mead, Nancy R. et al. SEI/CMU. http://www.cert.org/mswa/ ; http://www.cert.org/podcast/show/20101026mead.html	Offers a core body of knowledge from which to create a master's level degree program in software assurance, as a standalone offering and as a track within existing software engineering and computer science master's degree programs. Last updated **2010**.
	Volume II: Undergraduate Course Outlines. Mead, Nancy R. et al. SEI/CMU. http://www.cert.org/mswa/	Focuses on an undergraduate curriculum specialization for software assurance. Intended to provide students with fundamental skills for either entering the field directly or continuing with graduate level education. Last updated **2010.**
Software Security Assurance SOAR	Software Security Assurance: A State-of-the-Art Report. Goertzel, Karen Mercedes, et al, IATAC of the DTIC. http://iac.dtic.mil/iatac/download/security.pdf	Identifies the current "state-of-the-art" in software security assurance. Last updated July **2007**.
	Enhancing the Development Life Cycle to Produce Secure Software: A Reference Guidebook on Software Assurance. Goertzel, Karen et al. For DHS and DTIC, https://www.thedacs.com/techs/enhanced_life_cycles/	Complements the Software Security Assurance: A State-of-the-Art Report with further details. Last updated October **2008**
SwA CBK and Organization of Principles and Guidelines	Software Assurance Body of Knowledge. Version 1.2, Samuel T. Redwine, Jr. (Editor), DHS, https://buildsecurityin.us-cert.gov/bsi/dhs/927-BSI.html	Provides a comprehensive set of principles and guidelines from the disciplines of software engineering, systems engineering, information system, computer science, safety, security, testing, information assurance, and project management. Last updated October **2007**.
	Towards an Organization for Software System Security Principles and Guidelines. Version 1.0, Samuel T. Redwine, Jr., https://buildsecurityin.us-cert.gov/bsi/dhs/927-BSI.html	Provides an extensive set of software system security principles and guidelines organized in a logical, in-depth fashion. Last updated February **2008**.

Workforce Development and Improvement

Table 2– Workforce Development and Improvement

Identifier	Relevant Documents and Links	Purpose
DoD 8570.01-M	Information Assurance Workforce Improvement Program. Assistant Secretary of Defense for Networks and Information Integration/Department of Defense Chief Information Officer. http://www.dtic.mil/whs/directives/corres/pdf/857001m.pdf	Provides guidance and procedures for the training, certification, and management of the DoD workforce conducting Information Assurance (IA) functions in assigned duty positions. Last update: Incorporating Change 2, April 20, **2010**.
EBK	IT Security Essential Body of Knowledge (EBK): A Competency and Functional Framework for IT Security Workforce Development. DHS US-CERT http://www.us-cert.gov/ITSecurityEBK/	Characterizes the IT security workforce and provides a national baseline of essential knowledge and skills that IT security practitioners should have in order to perform specific roles and responsibilities. Last updated September**2008.**
Information Security Workforce Development Matrix Project – Information Security Systems and Software Development Professional Role	See Section *Role Descriptions* section, page *17*, for materials from the matrix project developed by Federal CIO Council's IT Workforce Committee and Information Security and Identity Management Committee. Contacts: http://www.cio.gov/committees.cfm/csec/3/cid/4 http://www.cio.gov/committees.cfm/csec/3/cid/5	This project produces role-based information security workforce development matrices. The matrices are one-page dashboards intended to establish a baseline across the Federal Government for staff engaged in information security work. This initiative provides a government-wide perspective on common information security roles. The ISS&SDP is one of 11 roles that have been identified to date. Each matrix contains a uniform framework, by performance level, describing the recommended competencies/skills, education, experience, credentials, and training for a particular role. The matrices provide guidance for federal agencies and do not replace OPM basic qualifications.

Strategies for Injecting SwA Knowledge

Table 3– Strategies for Injecting SwA Knowledge Areas into existing Education and Training Programs

Strategy	Relevant Documents and Links
Degree programs and specializations in SwA	Reference curriculums available from the Software Engineering Institute, Carnegie Mellon University can be used as recommendations for designing Masters of Software Assurance degree program and undergraduate curriculum specialization in software assurance. These reference curriculums are available at http://repository.cmu.edu/sei/3/ and http://repository.cmu.edu/sei/4/ Graduate Certificates and Master Degree Concentrations at the Stevens Institute of Technology:http://dc.stevens.edu/academic-programs/software-assurance/
Community Support	**LinkedIn SwA Education Discussion Group** Nancy Mead, SwA Curriculum Team lead The objective of the SwA Curriculum Development Team in establishing this group is to provide a venue for dialog about software assurance education. http://www.linkedin.com/groups?mostPopular=&gid=3430456 **Software Assurance Mobile Instructional (device) SAMI** Dan Shoemaker, University of Detroit Mercy A catalogue of available software assurance materials that are packaged and delivered using an iPad based instruction medium for educators
Credentialing	Several certification options are now available to suit the needs of specific job functions required in an enterprise. More information can be found in the Workforce Credentials section of this guide.

Table 3– Strategies for Injecting SwA Knowledge Areas into existing Education and Training Programs

Strategy	Relevant Documents and Links
Stand-alone Courses	New course offerings based on SwA knowledge areas complement existing Software Engineering courses. Examples: http://www.cs.jmu.edu/sss https://www.securecoding.cert.org/confluence/display/sci/S08+15392+Secure+Programming Also: graduate-level Software Assurance courses that cover secure software engineering activities during the SDLC are offered at the University of North Carolina at Charlotte, and The University of Nebraska at Omaha.
Augmenting Existing Courses	The SwA CBK and State-of-the-Art reports are catalogs of secure software development practices, processes, and techniques that can be mapped to topics relevant to current curriculums. The identified gaps can then be filled using relevant materials.
Micro-Modules	Problem-based learning exercises, in class workshops, or short talks to inject topics such as Misuse Cases and Assurance Cases into existing software engineering or information security courses.
Capstone and Class Projects	Software Engineering capstone courses or class projects which can be geared towards a security critical domain such as designing a software system for the Department of Defense, Cyber-physical systems or for a Credit Card transaction processing company. These domains will facilitate the exploration of security needs throughout the SDLC.
Online Courses	The Adaptive Cyber-Security Training Online (**ACT-Online**) courses are available on the TEEX Domestic Preparedness Campus. Ten courses address three discipline- specific tracks. The targets are everyday non-technical computer users, technical IT professionals, business managers and professionals. These courses are offered at no cost and students earn a DHS/FEMA Certificate of completion along with Continuing Education Units (CEU) at the completion of each course. http://www.teexwmdcampus.com/index.k2
	The **CERT Virtual Training Environment (VTE)** combines the components of traditional classroom training with the convenience of web-based training. Over 200 hours of course material focused around the technical, policy, and management implications of information security – including preparatory courses for commercial certifications, core skills courses, role-based courses for managers and technical staff, and vendor-developed courses. Open access is provided to individual DoD personnel (Active Duty, DoD Civilian and contractors) and members of the Federal Civilian Workforce through specific sponsorships from DISA, and DHS in conjunction with the Department of State Foreign Service Institute. Sponsored accounts can be requested at www.vte.cert.org. Public access to many of the materials is provided through the VTE Library at https://www.vte.cert.org/vteweb/Library/Library.aspx

Tools

Tools and web resources that can be used in class to provide hands-on experience with SwA Concepts.

Table 4 – Tools and web resources for hands-on classroom experience with SWA Concepts

Tool Name	Tool Description	Possible Classroom Uses
ArgoUML	ArgoUML is the leading open source UML modeling tool and includes support for all standard UML 1.4 diagrams. It runs on any Java platform.	Misuse cases, security focused UML class diagrams and other documentation for class assignments and projects.
ASCE	ASCE supports the key assurance case notations: Goal Structuring Notation and Claims-Arguments-Evidence. Academic license available upon request at http://www.adelard.com/web/hnav/ASCE/index.html	Assurance case documentation for class assignments and projects, Demonstration of worked examples used on real projects.
Burp Suite	Burp Suite is an integrated platform for attacking web applications. Located at http://www.portswigger.net/suite/	Burp Suite allows the combination of manual and automated techniques to enumerate, analyze, scan, attack and exploit web applications.
CERT Secure Coding Standards	Secure coding standards for commonly used programming languages such as C, C++ and Java. Located at https://www.securecoding.cert.org	Online reference; examples of coding do's and dont's.
FindBugs™	A program which uses static analysis to look for bugs in Java code at http://findbugs.sourceforge.net/	Scan java code repositories for bugs; Introduction to static code checking activities.
Microsoft SDL Threat Modeling Tool	The Microsoft SDL Threat Modeling Tool allows software architects to identify and mitigate potential security issues early, when they are relatively easy and cost-effective to resolve. Located at http://www.microsoft.com/security/sdl/getstarted/threatmodeling.aspx	Conduct student group workshops to discuss threats to various design alternatives, while suggesting possible mitigation strategies.

Table 4 – Tools and web resources for hands-on classroom experience with SWA Concepts		
Tool Name	**Tool Description**	**Possible Classroom Uses**
Olly Debug	OllyDbg is a 32-bit assembly level debugger for Microsoft Windows. Located at www.ollydbg.de/	Emphasize binary code analysis and is particularly useful in cases where source is unavailable. Explain Buffer Overflows.
Pharos	Pharos is an open source proxy that traps all HTTP and HTTPS data between server and client, including cookies and form fields, which can be intercepted and modified. Located at http://parosproxy.org/index.shtml	Pharos can be used as an introduction to web application security assessment.
SAMATE Reference Dataset	The purpose of the SAMATE Reference Dataset (SRD) is to provide users, researchers, and software security assurance tool developers with a set of known security flaws. This will allow end users to evaluate tools and tool developers to test their methods. Located at http://samate.nist.gov/index.php/Main_Page.html.	A reference data set can be used in class to reflect upon known flaws in software.
SDMetrics	Analyze the structural properties of UML models using object-oriented measures of design size, coupling, and complexity. Located at http://www.sdmetrics.com/	Examine object-oriented metrics and measures for design and source code artifacts.
Splint	Splint is a tool for statically checking C programs for security vulnerabilities and coding mistakes. Located at http://www.splint.org/	Static analysis code checking activities.
Valgrind	Valgrind is an instrumentation framework for building dynamic analysis tools. Located at http://valgrind.org/	Demonstrate dynamic analysis techniques to detect memory management and threading bugs, as well as detailed program profiling.
Vine	Provides an intermediate language that x86 code can be translated to for Static analysis. Located at http://bitblaze.cs.berkeley.edu/vine.html	Identify data flows analysis and conduct binary analysis.
Web Resources		
Google Code	http://google-gruyere.appspot.com	Web application exploits and defenses

Books

Table 5 – A List of SwA focused Books for Use in Education and Training		
Topic	**Title and Publisher**	**Summary and Possible Use**
Software Assurance in SDLC	**Secure Coding: Principles and Practices**, Mark G. Graff and Kenneth R. van Wyk, O'Reilly, 2003	A practical approach to integrating SwA topics into the SDLC. Great for assignment of additional readings that complement classroom materials. http://www.securecoding.org/
Information Security	**Building a Secure Computer System**, Morrie Gasser, 1988	Good reading for Information Security basics.
Activities to improve SwA during the SDLC	**Software Security: Building Security In**, Gary McGraw, Addison-Wesley Professional, 2006.	Introduction to Software Security Touchpoints during software development. Possible use as a textbook or additional reference material.
	The Security Development Lifecycle: SDL: A Process for Developing Demonstrably More Secure Software, Michael Howard, Steve Lipner, Microsoft Press, 2006	Adaptation of Microsoft's Security Development Lifecycle (SDL) with case-studies on several Microsoft products.
	Secure and Resilient Software Development, Mark S. Merkow, Lakshmikanth Raghavan, Auerbach Publications, 2010	A practitioner's perspective on enterprise assurance programs.
	Building Secure Software: How to Avoid Security Problems the Right Way, John Viega and Gary McGraw, Addison Wesley, 2002	Software Assurance principles and guidelines and Implementation level issues Possible use as a textbook or additional reference material
	Secure Programming for Linux and Unix	Unix systems-specific guidelines for C, C++

Standards of Practice

Table 6– Domain-specific SwA standards used in practice		
Standard	**Source**	**Purpose**
Assurance Process Reference Model (PRM)	Presentation: https://buildsecurityin.us-cert.gov/swa/downloads/ACSAC2010BartolMoss12-05-2010.pdf Self Assessment: https://buildsecurityin.us-cert.gov/swa/downloads/20100922_PRM_Practice_List_2page.pdf	The Assurance PRM can be used to help organizations conduct a gap analysis of existing practices. The results of a gap analysis can be used to prioritize and track SwA implementation efforts. The Assurance PRM addresses assurance from executive to developer.
BSIMM2: The Building Security In Maturity Model	http://bsimm2.com/	Pronounced "bee simm" was created by observing and analyzing real-world data from thirty leading software security initiatives. The BSIMM can help you determine how your organization compares to other real-world software security initiatives and what steps can be taken to make your approach more effective.
CERT Resilience Management Model	http://www.cert.org/resilience/rmm.html	It has two primary objectives: 1. Establish the convergence of operational risk and resilience management activities such as security, business continuity, and aspects of IT operations management into a single model. 2. Apply a process improvement approach to operational resilience management through the definition and application of a capability level scale that expresses increasing levels of process improvement.

Workforce Credentials

Table 7– Certification and Training Opportunities		
Certification Authority	**SwA Relevant Certificates**	**Resources**
EC-Council	EC-Council Certified Secure Programmer (**ECSP**) (Technologies Covered: C/C++, Java, .Net, PHP, SQL)	http://www.eccouncil.org/certification.htm
	Certified Secure Application Developer (**CSAD**)	
	Certified Ethical Hacker (**CEH**)	
	Licensed Penetration Tester (LPT)	
GIAC - Global Information Assurance Certification	GIAC Secure Software Programmer - .NET (**GSSP-NET**)	http://www.giac.org/certifications/
	GIAC Secure Software Programmer - Java (**GSSP-JAVA**)	
	GIAC Web Application Penetration Tester (**GWAPT**)	
	GIAC Certified Penetration Tester (**GPEN**)	
IEEE Computer Society	Certified Software Development Professional (**CSDP**)	http://www.computer.org/portal/web/certification
(ISC)²	Certified Secure Software Lifecycle Professional (**CSSLP**)	http://www.isc2.org/csslp-certification.aspx

Job Roles

- **What kind of jobs can I get ?**
  - Jobs and career planning
    - http://www.sans.org/20coolestcareers

#18 - Security-savvy Software Developer*

"Kool, because this is VERY rare."

Job Description

The security-savvy software developer leads all developers in the creation of secure software, implementing secure programming techniques that are free from logical design and technical implementation flaws. This expert is ultimately responsible for ensuring customer software is free from vulnerabilities that can be exploited by an attacker.

#2 - System, Network, and/or Web Penetration Tester* - Top Gun Job

"You can be a hacker, but do it legally and get paid a lot of money!"

Role Descriptions

- **Cyber Software Assurance Developer/Integrator**
  - Experience with applying security activities within SDLC
  - Experience with security, including CSSLP, CISSP and SANS secure programming assessments
  - Experience with security standards, including SSE-CMM, NIST SPs, ISO 15408
  - Common Criteria, or client-specific software assurance guides. (Also see the section on "Standards of Practice")

- **Software Assurance Engineer**
  - Provide technical leadership in all aspects of software assurance and computer systems engineering support
  - Lead and actively participate in the evaluation and analyses of activities related to all phases of the secure software life cycle from initial planning, requirements definition, design and development through integrated system testing and sustaining operations.
  - Responsibilities will also include the support of a wide range of technical and programmatic activities for program offices, including leading the review and assessment of software system architecture; system requirements and their allocation to lower level specifications; design, code and test activities; trade studies; COTS/GOTS products; reuse software; test tools; simulators; software verification and validation (V&V); and system test and integration. Support independent review efforts in analyzing and assessing system software and related development and testing activities.

» Information Security Systems & Software Development Professional (ISSSDP)

- The Information Security Systems and Software Development Professional is responsible for secure design, development, testing, integration, implementation, sustainment, and/or documentation of software applications (web based and non-web) following formal secure systems development lifecycle processes and using security engineering principles.
- The following professional requirements form part of a broad Federal Government effort to identify and describe roles.

Table 8– Credentials, Competencies and Skills by Performance Level

	Entry Level	Intermediate Level	Advanced Level
Software Development Written & Oral Communication Creative Problem Solving Information Security/Assurance Critical Thinking and Analytical Skills	Yes	Yes	Yes
Software Engineering Project/Program Management Leadership & People Management	No	Yes	Yes
Suggested Credentials in: ▪ Computer science/engineering ▪ Database/information management ▪ Information assurance/security ▪ Software assurance/security ▪ Information systems management	Associate's Degree from an accredited program	Bachelor's Degree from an accredited program	Master's Degree from an accredited program plus 5 years' experience

Got Content?

- The pocket guide is a “work in progress”
- Plenty of opportunity to contribute content
- Join the Effort !
  - Your comments, suggestions, criticism/praise are all very welcome

Where to find the PocketGuide?

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

- And many others…

SwA Communities
SwA Forums & Working Groups
Workforce Education & Training
Processes & Practices
Technology, Tools & Product Eval.
Acquisition & Outsourcing
Measurement
Business Case
Malware
SwA Market Place
SwA Landscape
SwA Ecosystem
Making Security Measurable
Build Security In

Homeland Security

BUILDING SECURITY IN

Software Assurance Pocket Guide Series

The SwA Pocket Guide Series comprises free, downloadable documents on software assurance in acquisition and outsourcing, software assurance in development, the software assurance life cycle, and software assurance measurement and information needs. SwA Pocket Guides are developed collaboratively by the SwA Forum and Working Groups, which function as a stakeholder community that welcomes additional participation in advancing and refining software security. Your input on these documents is welcome; please use the feedback form. For general inquiries, please email Software.Assurance@dhs.gov.

- **SwA in Acquisition and Outsourcing**
  - Software Assurance in Acquisition and Contract Language
  - Software Supply Chain Risk Management and Due Diligence
- **SwA in Development**
  - 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
- **SwA Life Cycle**
  - Software Assurance in Education, Training & Certification
- **Future SwA Pocket Guides**

▸ **SwA in Acquisition and Outsourcing**

▸ **SwA in Development**

▾ **SwA Life Cycle**

Software Assurance in Education, Training & Certification
Life Cycle Support Volume I – (Version 2.1, March 1, 2011)
Current events related to cybersecurity encourage a fundamental shift in the way we think about educating and training a workforce prepared to address security issues in all phases of a software system. Software assurance education and training is aimed to ensure adequate coverage of requisite knowledge areas in contributing disciplines such as software engineering (including its many subdisciplines), systems engineering, project management, etc., to identify and acquire competencies associated with secure software. The primary audiences for this pocket guide are educators and trainers who can use this guide to identify resources to supplement their efforts as well as identify strategies to inject software assurance related topics in the existing education and training programs.

8.5" x 11" version PDF File

▸ **Future SwA Pocket Guides**

Homeland Security

SwA in Acquisition and Outsourcing

SwA in Development

Key Practices for Mitigating the Most Egregious Exploitable Software Weaknesses

Development Volume II – (Version 1.3 May 24, 2009)

Common Weakness Enumeration (CWE) provides a standard means for understanding residual risks; enabling more informed decision-making by suppliers and consumers about the security of software. The 2009 CWE/SANS Top 25 Most Dangerous Programming Errors is a list of the most egregious programming errors that can lead to serious exploitable software vulnerabilities. The main goal for the Top 25 list is to stop vulnerabilities at the source by educating programmers on how to eliminate the most egregious programming errors before software is shipped. The list could be used as a tool for education and awareness that helps programmers prevent the kinds of vulnerabilities that plague the software industry. Software consumers could use the same list to help them to ask for more secure software, while software managers and CIOs could use the Top 25 list as a measuring stick of progress in their efforts to secure their software. This volume of pocket guide links the CWE's with the common attacks that exploit these weaknesses, the resulting mission and business risks. It provides recommended practices for preventing the exploits in software.

8.5" x 11" version PDF File

Software Security Testing

Development Volume III – (Version 0.7 May 10, 2010)

Software security testing validates the secure implementation of a product thus reducing the likelihood of security flaws being released and discovered by customers or malicious users. The goal is not to "test in security," but to validate the robustness and security of the software products prior to making them available to customers and to prevent security vulnerabilities from ever entering the software. This volume of the pocket guide describes the most effective security testing techniques, their strengths and weaknesses, and when to apply them during the Software Development Life Cycle.

8.5" x 11" version PDF File

Requirements and Analysis for Secure Software

Development Volume IV – (Version 1.0, October 5, 2009)

Comprehensive requirements are critical for successful system development, but all too often, requirements fail to explicitly consider security. As a result, systems meet the functionality but are rarely safe and consequently are the target of attacks. Systems which carefully document security requirements reduce the likelihood of successful attacks. Security requirements include functions that implement a security policy such as areas of access control, identification, authentication and

Malware
SwA Market Place
SwA Landscape
SwA Ecosystem
Making Security Measurable
Build Security In

Homeland Security

BUILDING SECURITY IN

- SwA in Development
  - 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
- SwA Life Cycle
  - Software Assurance in Education, Training & Certification
- Future SwA Pocket Guides

SwA in Acquisition and Outsourcing

Software Assurance in Acquisition and Contract Language
Acquisition and Outsourcing Volume I – (Version 1.1 July 31, 2009)
Integrating software security in the acquisition life cycle promotes the acquisition of secure software. This volume of the pocket guide includes sample SwA Request for Proposal (RFP)/Contract language. Buyers and evaluators of software and suppliers can gain security risk-based insight. They can put suppliers on notice that consumers are concerned about software security and the risks to their organizations that are attributable to exploitable software.

8.5" x 11" version PDF File

Software Supply Chain Risk Management and Due Diligence
Acquisition and Outsourcing Volume II – (Version 1.2 June 16, 2009)
Software security enhanced due-diligence is a critical element of software supply chain risk management. The focus of the volume is to increase awareness for the need to include software assurance and identify best practices in the acquisition of software. Due-diligence involves taking reasonable steps to ensure that software or a software-intensive system not only meets functional and technical requirements, but also addresses software assurance concerns. Buyers and evaluators of software and services can gain security risk-based insight. They can put suppliers on notice that consumers are concerned about software security and the risks to their organizations that are attributable to exploitable software.

8.5" x 11" version PDF File

SwA in Development

SwA Life Cycle

Future SwA Pocket Guides

Find me

- Robin A. Gandhi, Ph.D.
  Assistant Professor of Information Assurance
  University of Nebraska at Omaha

  rgandhi@unomaha.edu

  Voice: (402) 554 3363, Fax: (402) 554-3284

  http://faculty.ist.unomaha.edu/rgandhi

Acknowledgement

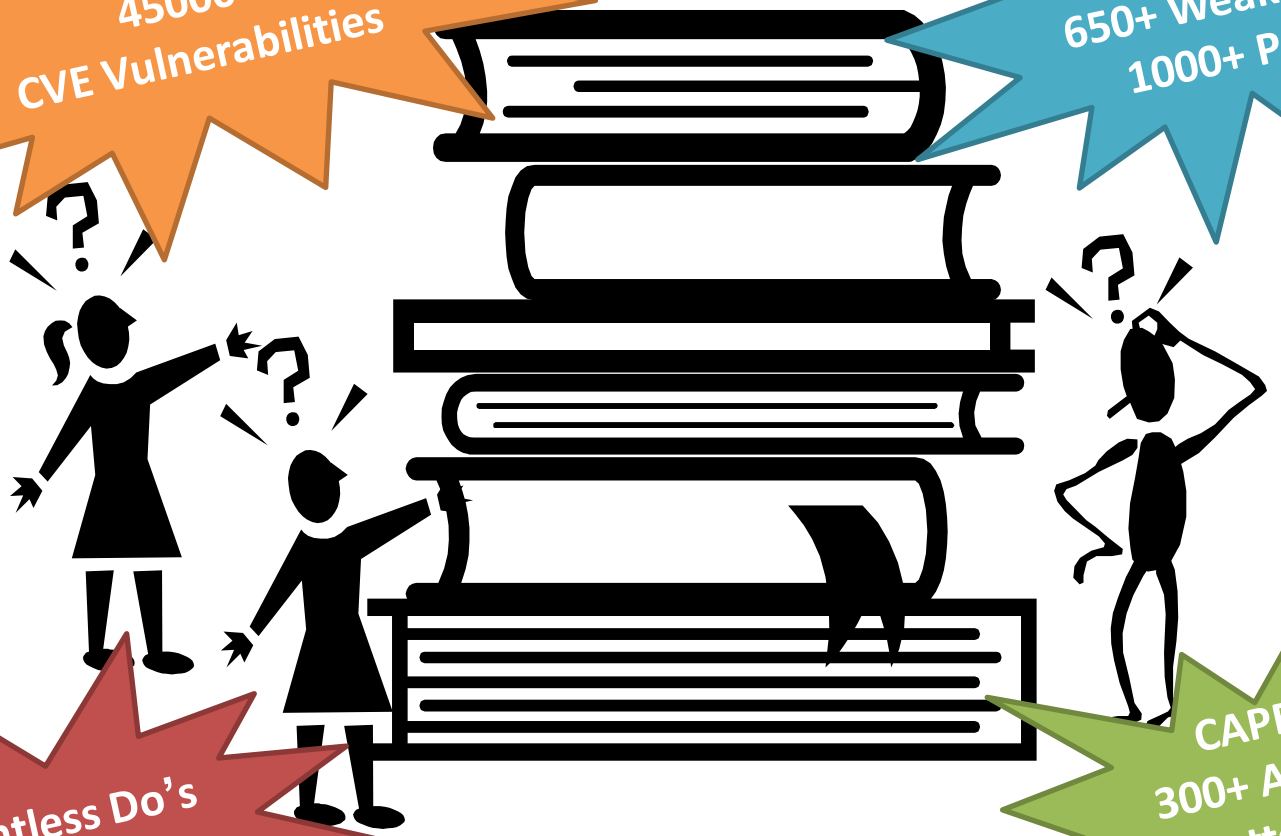
- Joe Jarzombek for giving me the opportunity to lead this effort
- Members of the SwA WG on Education and Training for insightful comments, reviews and content (Dan, Carol, Nancy, Art)
- Susan Morris, Walter Houser, Dominick Chiriyan
- And many others…

Bonus Slides

Why Johnny Can't write secure code?

- *Johnny, avoid these weaknesses…. Period!*
  - Common Weaknesses Enumeration (CWE)
- *Johnny…learn from your mistakes*
  - Common Vulnerabilities and Exposures (CVE)
- *Johnny…these are the ways of the bad guys*
  - Common Attack Patterns Enumeration and Classification (CAPEC)
- *Johnny…these are ways to develop secure code*
  - CERT secure coding guidelines

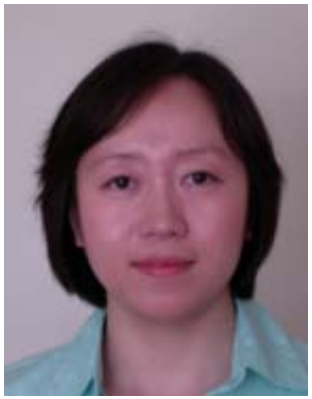
Poor Johnny !


45000+
CVE Vulnerabilities
CWE
650+ Weaknesses
1000+ Pages
Countless Do's
and Don'ts
CAPEC
300+ Attack
Patterns

Using Semantic Templates to Study Vulnerabilities Recorded in Large Software Repositories

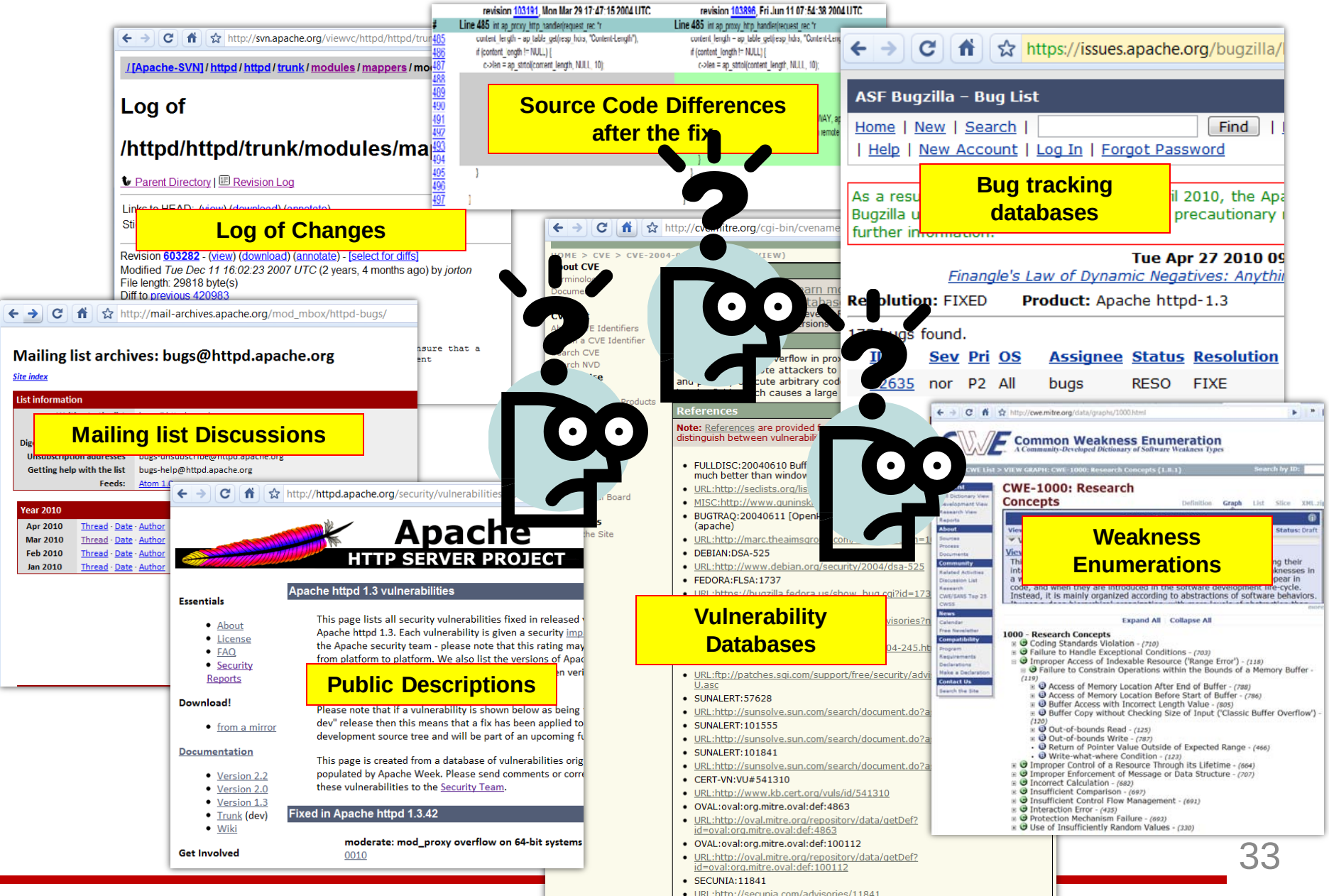
Me

Harvey Siy

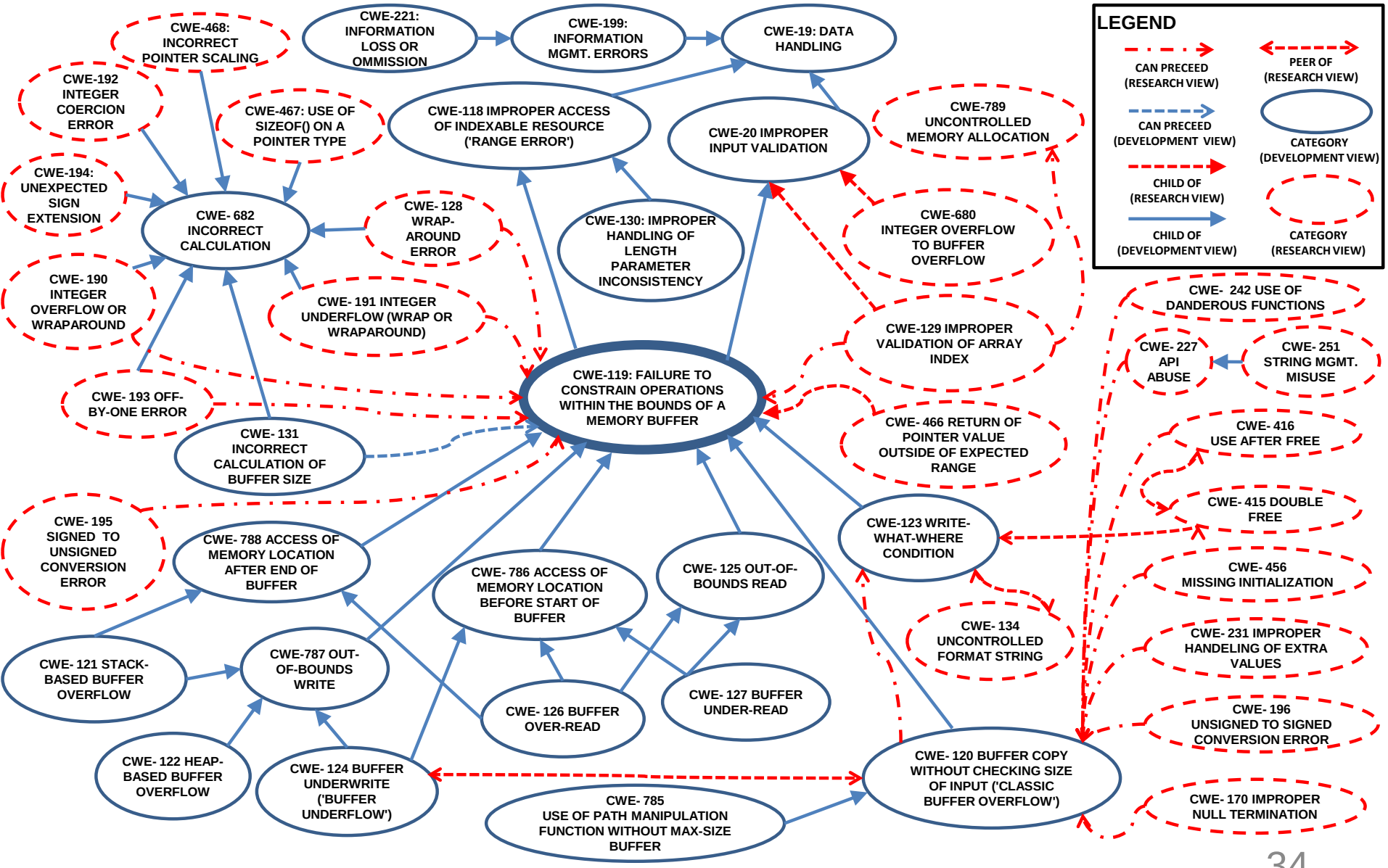


Yan Wu

The Paradox we face !



Concept Extraction



Tangling of information in the CWE

- **CWE-119: Failure to Constrain Operations within the Bounds of a *Memory Buffer***
  - The software performs operations on a ***memory buffer***, but it can read from or write to a memory location that is outside of the intended boundary of the ***buffer***.
  - Certain languages allow direct addressing of memory locations and do not automatically ensure that these locations are valid for the memory buffer that is being referenced. This can cause read or write operations to be performed on memory locations that may be associated with other variables, data structures, or internal program data. As a result, an attacker may be able to execute arbitrary code, alter the intended control flow, read sensitive information, or cause the system to crash.

LEGEND

Software Fault

Weakness

Resource/Location

Consequence

Tangling of information in the CWE

- CWE-120: ***Buffer*** **Copy without Checking Size of Input** ('Classic **Buffer Overflow**')
  - The program copies an input ***buffer*** to an output ***buffer*** without verifying that the size of the input ***buffer*** is less than the size of the output ***buffer***, leading to a **buffer overflow**.
  - A **buffer overflow** condition exists when a program attempts to put more data in a ***buffer*** than it can hold, or when a program attempts to put data in a ***memory area*** outside of the boundaries of a ***buffer***.
  - **Buffer overflows** often can be used to **execute arbitrary code**…
  - **Buffer overflows** generally **lead to crashes**

LEGEND

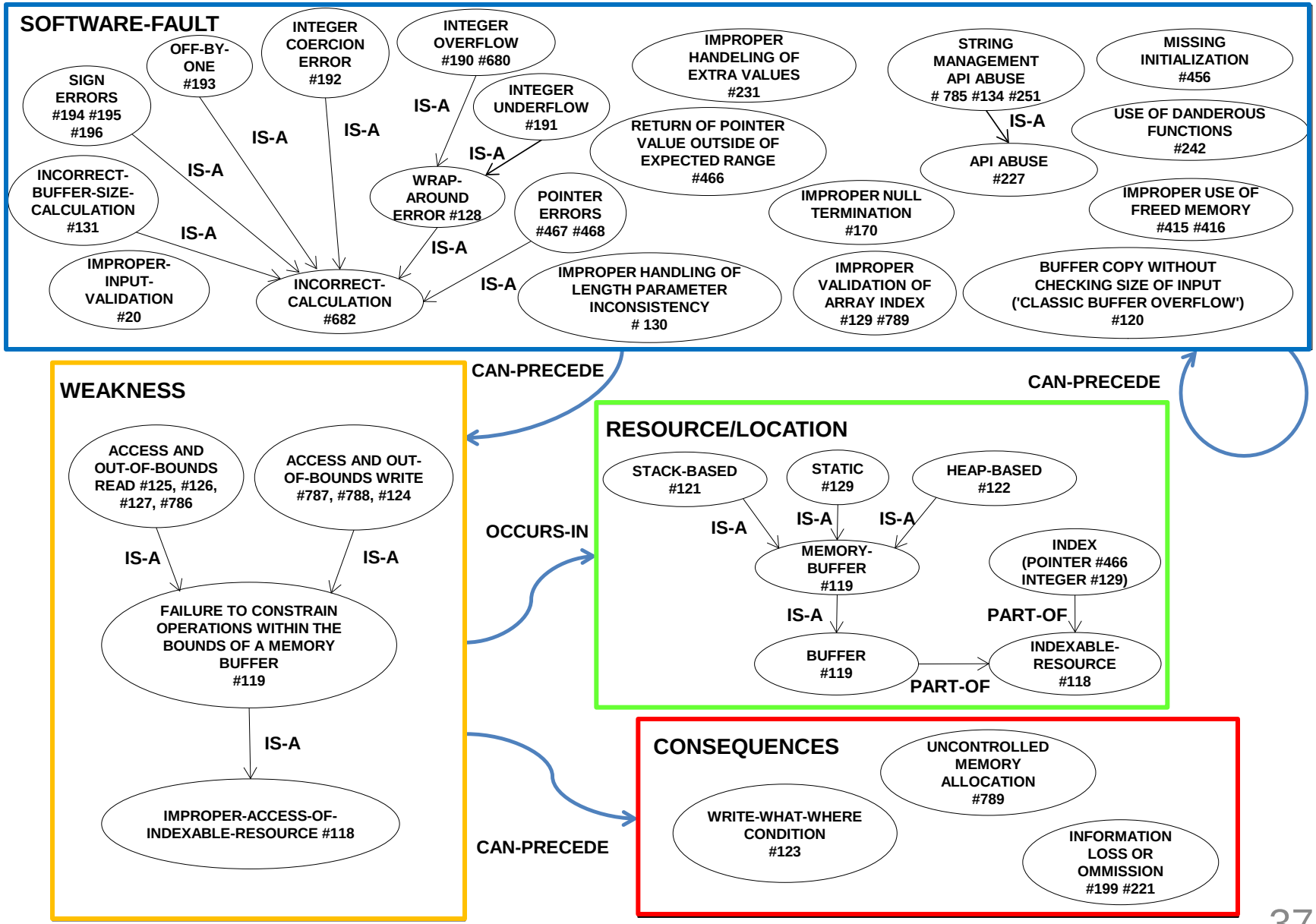
Software Fault

Weakness

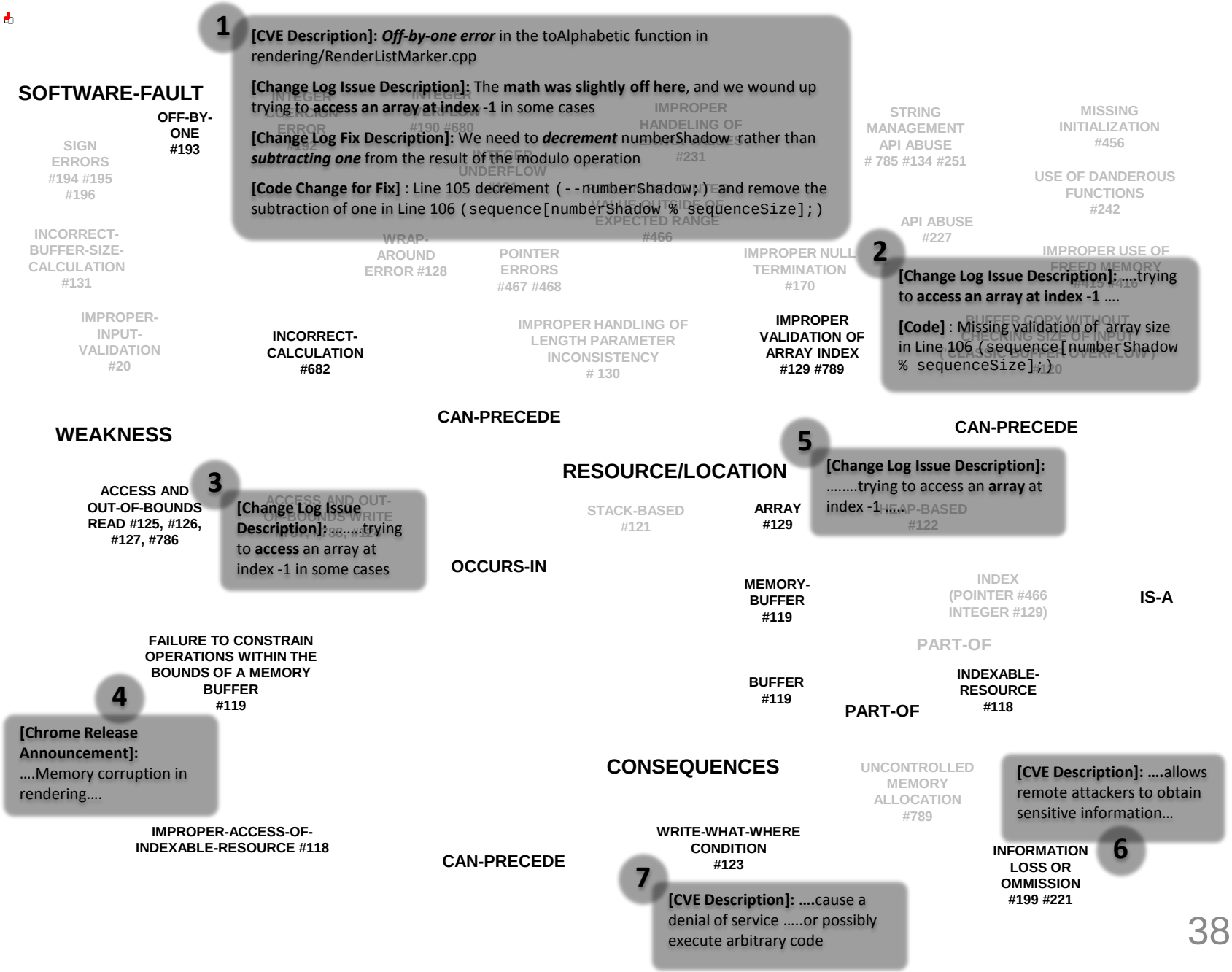
Resource/Location

Consequence

Buffer Overflow



Buffer Overflow Semantic template CVE-2010-1773



Experiment

- The scenario…
  - A newbie programmer or occasional contributor to open source project
    - How much effort does it take to study a vulnerability and summarize lessons learned?
- 30 Computer Science students from a senior-level undergraduate Software Engineering course.
  - None to more than 5 years
  - No prior knowledge of semantic templates

Experiment

- $H1_0$:
  - There is no reduction in completion time for subjects who use semantic templates compared to those who do not.
- $H2_0$:
  - There is no improvement in accuracy of understanding of vulnerabilities for subjects who use semantic templates compared to those who do not.

Variables

- The experiment manipulated these *independent variables*:
  - **Group** - refers to the group assigned (1 or 2).
  - **Round** - refers to the experiment round (1 or 2).
- **Vulnerability ID** - the vulnerability under study (1-1, 1-2, 1-3, 2-1, 2-2, 2-3).
  - These self-reported *subject variables* were collected:
    - **Programming skill level**
    - **Reading comprehension and writing skill levels** - ability to read and write technical English documents.

Variables

- Dependent variables :
  - **Time to complete assignment**
  - **CWE identification accuracy**
  - **Fault identification accuracy**
    - a score (scale of 1-5) on the accuracy of the identification of the software fault that led to the vulnerability
  - **Failure identification accuracy**
    - a score (scale of 1-5) on the accuracy of the description of the nature of the vulnerability (the manifested problem, the resources impacted and the consequences)

Initial Results and Findings

Table 1: p-values of one-tailed t-tests for Time data

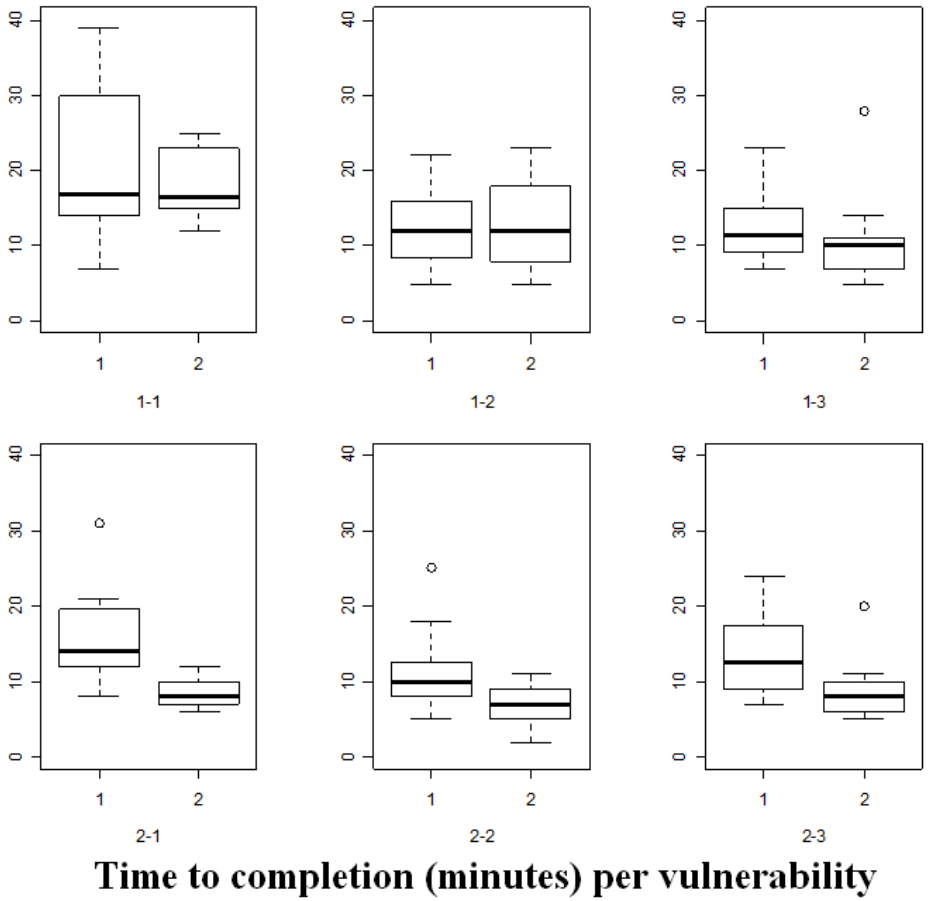
Round 1	(1-1) 0.3627	(1-2) 0.5855	(1-3) 0.1516
Round 2	(2-1) **0.0001**	(2-2) **0.0030**	(2-3) **0.0015**

p-values of one-tailed t-tests for CWE precision

Round 1	(1-1) 0.9281	(1-2) 0.9957	(1-3) 0.5344
Round 2	(2-1) 0.1840	(2-2) 0.6023	(2-3) 0.0891

Table 1: p-values of one-tailed t-tests for CWE recall

Round 1	(1-1) 0.0683	(1-2) 0.9481	(1-3) 0.2286
Round 2	(2-1) **0.0141**	(2-2) **0.0093**	(2-3) **0.0021**



Time to completion (minutes) per vulnerability

Future Work

- Integrate with existing static and dynamic analysis tools to enhance reporting capabilities
  - Provide layers of guidance to a developer upon detection of a software flaw
  - Organize and retrieve knowledge of past vulnerabilities
  - Verify patch submissions
- Investigate project/developer specific coding errors and vulnerability fix patterns
- Other usage scenarios in the SDLC

Acknowledgement

- This research is funded in part by Department of Defense (DoD)/Air Force Office of Scientific Research (AFOSR), NSF Award Number FA9550-07-1-0499, under the title "**High Assurance Software**"

Thank you for your Attention