Human Element in Software Supply Chain Integrity

Humans are incapable of securely storing high-quality cryptographic keys, and they have unacceptable speed and accuracy when performing cryptographic operations. (They are also large, expensive to maintain, difficult to manage, and they pollute the environment. It is astonishing that these devices continue to be manufactured and deployed. But they are sufficiently pervasive that we must design our protocols around their limitations.)

From "Network Security", by Kaufmann, Perlman and Speciner

Homeland Security

DoD-DHS-NIST Software Assurance Forum Software Supply Chain Integrity- People Behind Principle

Facilitator: Vehbi Tasar, (ISC)²

Mini-Keynote: Alan Paller, SANS Institute

Homeland Security

Software Supply Chain Integrity– People Behind the Principle

- **AS THE CONTROLS AND BEST PRACTICES REQUIRED TO ENSURE THE SOFTWARE SUPPLY CHAIN INTEGRITY EVOLVE, HOW WILL THE ROLE OF "PEOPLE" ADDRESSED?**
  - **GLOBAL SUPPLY CHAIN, OUTSOURCING - Alan Paller**
  - **ORGANIZATIONAL & CULTURAL ISSUES- Vehbi Tasar**
  - **POLICY & PERSONNEL TRAINING – David Stender**
  - **SECURE CODING AND USER INTERFACE – Robert Seacord**
  - **USING METRICS TO MITIGATE RISK – Shari Pfleeger**

Homeland Security

- **Director of Research for the SANS Institute**
- **Founded SANS Institute in 1989 that helped educate more than 80,000 people on technical security skills**
- **Edits NewsBites, the twice-a-week summary of the most important news stories in security**
- **Holds degrees from Cornell and MIT**
- **Authored hundreds of articles and two books.**

SANS

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

Speaker Bio: Vehbi Tasar

- **Responsible for exam development at (ISC)²**
- **Developed the CSSLP credential that was introduced in June, 2009.**
- **Prior to joining (ISC)², VP of engineering for Invio Software in Palo Alto, CA and Persystent Technologies in Tampa, FL.**
- **30 years of product development experience working for both small and large companies**
- **BS in electrical engineering, MS in Computer Science, doctorate in electrical engineering & CISSP and CSSLP**

(ISC)²®

- Differences in backgrounds, taxonomy and attitudes of different people roles in SDLC

- Ubiquitousness of SDLC- Software supply chain touches many people who are traditionally assumed to be outside of the SDLC

- Complexity of software and how it is interpreted by different roles in SDLC

(ISC)²®

- Differences in backgrounds, taxonomy and attitudes of different people roles in SDLC
  - IT Roles
  - SDLC Roles

(ISC)²®

- International Information Systems Security Certification Consortium (ISC)²
  - CISSP – 65,000 CISSP's in 135 countries
    - CISSP People Roles: Officer, Director, Manager, Leader, Supervisor, Analyst, Designer, Cryptologist, Cryptographer, Cryptanalyst, Architect, Engineer, Programmer, Instructor, Professor, Investigator, Consultant, Salesman, Representative
  - CSSLP – 859 CSSLP's in 45 countries
    - CSSLP People Roles: Software Developer, Software engineer, Architect, Product manager, Project manager, Software QA, QA tester, Business analyst, Professionals who manage these stakeholders

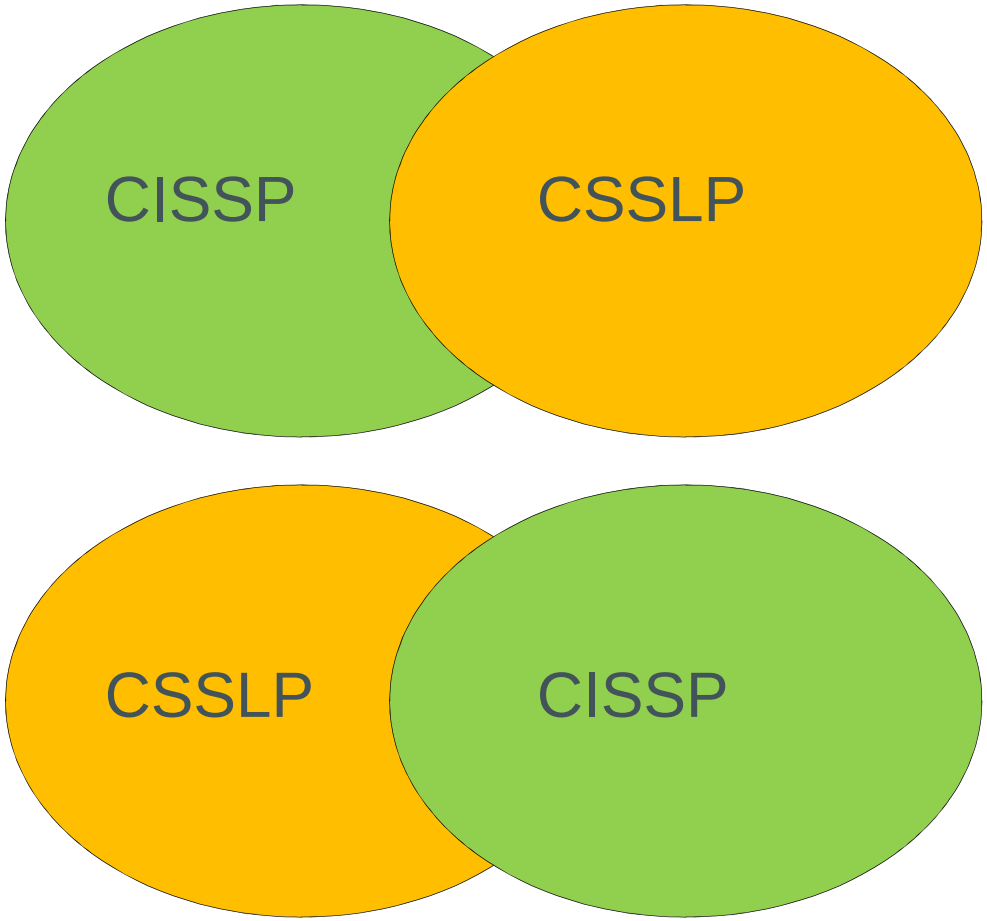
(ISC)²®

Cultural differences between roles

- The software supply chain integrity touches anyone who is directly or indirectly part of the software development lifecycle (SDLC), e.g.,
  - IT personnel, product managers, acceptance testers, developers, QA staff, compliance officers, managers, and users.
  - There are many in the SDLC community who see security as someone else's responsibility
  - There are many in the IT community who see software security as someone else's responsibility
  - One of the main tasks of assuring integrity in supply chain is to bridge the gaps between different cultures
    - IT personnel, software developers, testers, and pre and post sales engineers, etc.

(ISC)²®

Knowledge Gaps in Security



(ISC)²®

Name of the Security Principle	Description	CSSLP domain
Chain of Custody	Secure the source code during each change and handoff for its lifetime	Secure Software Implementation and Coding
Least Privilege Access	Access data with only the privileges needed to do the job	Secure Software Concepts& Secure Software Design
Separation of Duties	Do not allow unilateral change or control of data	Secure Software Concepts& Secure Software Design

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

Software Supply Chain Integrity Controls

Name of the Security Principle	CISSP domain	CSSLP domain
Chain of Custody	Physical Security & Access Control & Legal Regulations, etc. & Application Development Security	Secure Software Implementation and Coding
Least Privilege Access	Access Control & Operations Security & Physical Security	Secure Software Concepts& Secure Software Design
Separation of Duties	Operations Security & Access Control	Secure Software Concepts& Secure Software Design

Software Supply Chain Integrity Controls

Name of the Security Principle	Description	CSSLP domain
Tamper Resistance & Evidence	Obstruct attempts to tamper, and when they occur, make sure that they are evident and reversible	Secure Software Implementation and Coding
Persystent Protection	Protect critical data independent of its development location	Secure Software Design & Software Acceptance & Software Deployment, Operations, Maintenance and Disposal

Name of the Security Principle	CISSP domain	CSSLP domain
Tamper Resistance & Evidence	Legal Regulations, Investigations and Compliance & Physical Security	Secure Software Implementation and Coding
Persystent Protection	Operations Security & Cryptography & Security Architecture and Design & Business Continuity and Disaster Recovery Plan	Secure Software Design & Software Acceptance & Software Deployment, Operations, Maintenance and Disposal

Name of the Security Principle	Description	CSSLP domain
Compliance Management	Confirm the success of the protections continually and independently	Secure Software Requirements & Software Acceptance & Software Deployment, Operations, Maintenance and Disposal
Code Testing and Verification	Apply the methods for code inspection and detect suspicious code	Secure Software Implementation/Coding & Secure Software Testing & Software Acceptance

Software Supply Chain Integrity Controls

Name of the Security Principle	CISSP domain	CSSLP domain
Compliance Management	Legal Regulations, Investigations and Compliance & Information Security Governance and Risk Management	Secure Software Requirements & Software Acceptance & Software Deployment, Operations, Maintenance and Disposal
Code Testing and Verification	Application Development Security	Secure Software Implementation/Coding & Secure Software Testing & Software Acceptance

- Complexity of software and how it is interpreted by different roles in SDLC
  - Mizuho Securities Case-
    - Mizuho Securities– User of the software
    - Tokyo Stock Exchange- Owner of the software
    - Fujitsu- Developer of the software

(ISC)²®

- Complexity of software and how it is interpreted by different roles in SDLC
  - Supply chain integrity vulnerabilities revealed in Mizuho Case
    - Human interface – between requirements and design
    - QA process- between requirements and testing
    - Communication - between user and developer through the subcontracting chain
    - Product liability – between software engineer and legal system
    - Ethical considerations- between software engineer and society

(ISC)²®

- In the new world of Internet, Web 2.0 applications, cloud applications, security is a fundamental responsibility of everyone.
- Security of the software supply chain can be improved by helping CISSP and CSSLP roles learn from each other's culture and experience.

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

Speaker Bio: David Stender

- **Associate Chief Information Officer for Cybersecurity and Chief Information Security Officer, Internal Revenue Service**
- **More than 26 years of government, military, and business experience in developing and implementing security policy**
- **Holds CISSP, CSSLP and CAP credentials**
- **B.S., General Engineering and Political Science, U.S. Naval Academy; M.S., Telecommunications with IA emphasis, University of Maryland University College**

- **Senior Vulnerability Analyst, Secure Coding Team Lead at CERT/SEI**
- **Lead the secure coding initiative at CERT, including the development of secure coding standards for the C, C++, and Java programming languages**
- **Over 25 years of software development experience in industry, defense, and research**
- **Author of four books in SEI series including Secure Coding in C and C++ and the CERT C Secure Coding Standard**
- **BS in Computer Science from Rensselaer Polytechnic Institute**

CERT

Software Supply Chain Integrity: The People Behind the Principle

Secure Coding and User Interface

Robert C. Seacord

CERT

a) Trust the programmer.
b) Don't prevent the programmer from doing what needs to be done.
c) Keep the language small and simple.
d) Provide only one way to do an operation.
e) Make it fast, even if it is not guaranteed to be portable.
f) Make support for safety and security demonstrable (1)

(1) New for C1X

Not on the list:

a) The language should be easy to learn and easy
b) Always do the least surprising thing.

CERT

- The C Standard defines **undefined behavior** as:
- *Behavior, upon use of a nonportable or erroneous program construct or of erroneous data, for which the standard imposes no requirements. An example of undefined behavior is the behavior on integer overflow.*

CERT

Undefined behaviors are identified in the standard:

- If a "shall" or "shall not" requirement is violated, and that requirement appears outside of a constraint, the behavior is undefined.
- Undefined behavior is otherwise indicated in this International Standard by the words "undefined behavior"
- by the omission of any explicit definition of behavior.

There is no difference in emphasis among these three; they all describe "behavior that is undefined".

C99 Annex J.2, "Undefined behavior," contains a list of explicit undefined behaviors in C99.

CERT

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

Undefined Behaviors in C

Behaviors are classified as "undefined" by the standards committees to:

- give the implementer license not to catch certain program errors that are difficult to diagnose;
- avoid defining obscure corner cases which would favor one implementation strategy over another;
- identify areas of possible conforming language extension: the implementer may augment the language by providing a definition of the officially undefined behavior.

Implementations may

- ignore undefined behavior completely with unpredictable results
- behave in a documented manner characteristic of the environment (with or without issuing a diagnostic)
- terminate a translation or execution (with issuing a diagnostic).

CERT

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

Fun With Integers

- `char x, y;`
- `x = -128;`
- `y = -x;`

- `if (x == y) puts("1");`
- `if ((x - y) == 0) puts("2");`
- `if ((x + y) == 2 * x) puts("3");`
- `if (((char)(-x) + x) != 0) puts("4");`
- `if (x != -y) puts("5");`

CERT

Licensed to:
- Computer Associates
- Siemens
- SANS

SEI Secure Coding Course

University courses
- CMU
- Purdue
- University of Florida
- Santa Clara University
- St. John Fisher College

Secure Design Patterns

Influence International Standard Bodies

ISO IEC

Tool Test Suite

Secure Coding in C and C++ — Robert C. Seacord

The CERT C Secure Coding Standard — Robert C. Seacord

Adoption by Analyzer Tools

Application Conformance Testing

Adoption by software developers
- Lockheed Martin Aeronautics
- General Atomics

python™ C++ Java

LOCKHEED MARTIN — We never forget who we're working for™

GENERAL ATOMICS ELECTRONIC SYSTEMS

CERT

Breadth of impact

2003 Time 2010

Purpose: Study the problem of producing analyzable secure coding guidelines for C99 and C1x

First meeting held on October 27, 2009

Meetings will be held the first and third Wednesday of each month by teleconference

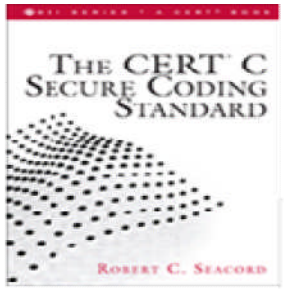
Thomas Plum is the inaugural chair

Robert Seacord is the project editor

CSCG SG Wiki: http://wiki.dinkumware.com/twiki/bin/view/CSCG/

Mailing list : wg14-cscg-l@cert.org

CERT



221 Guidelines

Security Manager

static analysis tools, unit tests, and fuzz testing

Deploy

Source Code

CERT

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

CERT SCALe (Source Code Analysis Lab)

Satisfy demand for source code assessments for both government and industry organizations

Assess source code against one or more secure coding standards.

Provided a detailed report of findings

Assist customers in developing certifiably conforming systems

CERT

DoD-DHS-NIST Software Assurance Forum

Using Metrics to Mitigate Risk

Presented By Shari Lawrence Pfleeger, Senior Information Scientist

RAND CORPORATION

Homeland Security

- 1996: Ariane 5 rocket exploded because of software overflow error in the inertial reference system. European Software Agency reused IRS from Ariane 4 rocket to save money. Estimated loss: $500 million.
- 1999: Mars Climate Orbiter didn't work because Lockheed Martin supplied NASA with imperial instead of metric units. Estimated loss: $125 million.
- 2003-4: FBI Trilogy program, based on Service Oriented Architecture, cancelled after many years of development. Estimated loss: $174 million.

RAND CORPORATION

Homeland Security

- 1996: Ariane 5 rocket exploded because of software overflow error in the inertial reference system. European Software Agency reused IRS from Ariane 4 rocket to save money. Estimated loss: $500 million.
  - Static code analysis could have identified potential buffer overflow.
  - Analysis of underlying assumptions would have helped, too.

RAND CORPORATION

Homeland Security

- 1999: Mars Climate Orbiter didn't work because Lockheed Martin supplied NASA with imperial instead of metric units. Estimated loss: $125 million.
  - Measurement-guided design review, code review or testing could have identified problems before fielding.

RAND CORPORATION

Homeland Security

- 2003-4: FBI Trilogy program, based on Service Oriented Architecture, cancelled after many years of development. Estimated loss: $174 million.
  - Static code analysis by Aerospace Corporation revealed disproportionate number of tiny (2-, 3-, 4-line) modules, suggesting that interfaces were unnecessarily complex.

RAND CORPORATION

Homeland Security

- Include measurement at all stages of development.
- Include funding for thorough measurement and review at each stage of development.
- Use measurement screen with suppliers, and don't accept delivery until measurement requirements are met.

RAND CORPORATION

Homeland Security