Engineering and Assurance for the Life Cycle Logistician

Mr. Thomas Hurt
Director, Joint Federated Assurance Center (JFAC)
OUSD(R&E)

2019 Product Support Manager (PSM) Workshop
Joint Base Andrews, MD | May 15, 2019
84% of breaches exploit the vulnerabilities in the application, yet funding for IT defense vs. software assurance is 23 to 1.
Who Fixes the Most Vulnerabilities?

What is the percentage of known vulnerabilities remedied by each industry vertical, in order to reduce application-layer risk?

- **Manufacturing**: 81%
- **Financial Services**: 65%
- **Retail + Hospitality**: 60%
- **Technology**: 50%
- **Healthcare**: 43%
- **Government**: 27%

The data represents 208,670 application assessments submitted for analysis during the 18-month period from October 1, 2014 through March 31, 2015 by large and small companies, commercial software suppliers, open source projects and software outsource.

Contest: Need for Engineering-in Software Assurance Activities over the Software Development Life Cycle (SDLC)

Where Software Flaws Are Introduced

- Requirements Engineering: 70%
- System Design: 20%
- Software Architectural Design: 10%
- Component Software Design: 3.5%
- Code Development: 16%
- Unit Test: 50.5%
- Integration: 9%
- System Test: 21%
- Acceptance Test: 9%
- Operation: 21%

Where Software Flaws Are Found

Improved focus on engineering-in software assurance activities needed on the front end of the SDLC

Source: Carnegie Mellon University, Software Engineering Institute (Critical Code; NIST, NASA, INCOSE, and Aircraft Industry Studies), used with permission.
Tools Throughout the System Life Cycle
(especially Sustainment)

With the integration and automation of software assurance tools throughout the system life cycle, programs can make informed decisions on the identification and mitigation of risk.
**New System Tactical Use Threads**

How will a component actually be used?

- Engine Control SW (ECS) provides needed metrics
- Input: Engine performance data; Output: Needed alerts/response
- Read/write capabilities to data bus do needed functions

**Mission Threads**

What will my system do?

- Engine functionality will be controlled by ECS
- Engine Monitoring System will monitor engine performance
- Performance issues will be transmitted by data bus to control panel

**System Requirements**

What is required to make my mission successful?

- ECS has no known vulnerabilities
- Monitoring SW cannot be exploited to access ESC or data bus
- Secure Design/Architecture considerations for Data bus communication

---

**Sound Systems Engineering**

Distribution Statement A: Approved for public release. Distribution is unlimited.
JFAC Program Support Activities

• Processes
  • Ticket and Response Coordination
  • Software Assurance License Procurement and Distribution
  • FOC Planning and Execution

• Working Groups
  • Action Officer Working Group
  • Software Assurance Technical Working Group
    • SwA Portal content sub-group
  • Hardware Assurance Technical Working Group
    • Standards and best practice, field-programmable gate array (FPGA), supply chain risk management (SCRM), Technical Assessment, ASSESS and EDA assurance sub-groups

• Applications
  • JFAC Portal
  • Assurance Knowledge Base (AKB)
  • Cyber Integrator

• Products
  • Defense Acquisition University Software Assurance Course (CLE 081)
  • Security Classification Guide
  • SwA Contract Language Guidance
  • State-of-the-Art Resource (SOAR) for Software Vulnerability Detection
JFAC Assurance Knowledge Base Support for Sound Systems Engineering

**Development Artifacts**
- Assurance assessments
- SwA tool findings
- Vulnerability prioritization (consequence and likelihood)
- Deployed assurance countermeasure rationale
- Mitigations
- Regression test results

**Transition to Sustainment**
- Latent vulnerabilities and characteristics
- Mitigated vulnerabilities
- Decisions and rationale
- Vulnerability test results
- Bill of materials (BoM)
- Chain of custody

Metadata collected through the identification of tactical threads, mission threads, and systems requirements throughout development is critical to sustainment of software.
Is the Future Sustainable?

New Features/Components Added continuously

Everything is interconnected or networked (Internet of Things (IoT))

Technology continues to advance (methods of attack)

The addition of new components, changes to the network, and advancement of adversary technology creates a continuous cycle of redesign and patching to protect against unwanted access.
• Latent software vulnerabilities identified or exploited in sustainment are exponentially more expensive to fix.

• Acquisition of source code and documentation in the data rights package are expensive and ineffective steps for legacy DoD programs. Logistical data needs must be included in the development Request for Proposals (RFP).

• Sound systems engineering, implementation of SwA countermeasures, and transition of assurance rational into sustainment is critical to the protection of our weapons systems.

• JFAC needs your advocacy for development programs to use the AKB to store and retain assessment data collected throughout development, test, and deployment for use in sustainment.

• Select JFAC assessment data retention uses:
  • Vulnerability and mitigation rationale retention throughout the life cycle
  • Data mining (tracking, trending, intel, etc.)
  • Chain of custody
  • Bill of materials

https://jfac.navy.mil
For Additional Information

Mr. Tom Hurt
Director, JFAC / Deputy Director, Software Assurance
Strategic Technology Protection and Exploitation
Office of the Under Secretary of Defense
for Research and Engineering
571-372-6129
thomas.d.hurt.civ@mail.mil
Engineering Software Assurance into the Life Cycle

Operational Need:
- Business or Mission Analysis: Identify threat environment and opportunities for attack
- Stakeholder Needs & Req Definition: Define functional requirements for operation in a cyber contested environment
- System Req. Definition: Derive non-functional SwA requirements
- Architecture Definition: Develop secure architecture
- Design Definition: Design system with considerations for SwA
- System Analysis: Criticality Analysis, SwA Evaluation of COTS

Delivered Capability:
- Operational Need
- Design
- Product
- Validation
- Continuous Application Across the Acquisition Lifecycle
- Sustainment and Continuous Engineering: Monitor for 3rd party vulnerabilities, continued assessment & timely patching
- Operation: Implement operational monitoring and response
- Validation: Conduct third party SwA testing, validate security requirements/assumptions
- Transition: Transition data rights, ensure acquirer can rebuild & retest

Requirements
Integration:
- Full system regression testing
- Automated, reproducible build

Implementation:
- Warning flags & Coding standards
- Hardening measures
- Code reviews

Verification:
- Static source code weakness analysis
- Binary analysis
- Origin analysis
- Web app scanners & fuzzers
- Negative testing
- Automated test suite w/coverage
- Penetration testing

Support:
- Risk Management: Assurance case
- Configuration Management: Version control, access control, code signing
- Measures & Metrics

Note: Lifecycle processes typically occur simultaneously, not in sequence; see ISO/IEC 15288 & 12207

Distribution Statement A: Approved for public release. Distribution is unlimited.
JFAC Service Provider Capabilities

**Software and Hardware Assurance (SwA and HwA) Requirements Support:**
Identification of applicable SwA and HwA requirements from policy, standards, instructions, and guidance

**Knowledge Source:**
Identification of applicable SwA and HwA assessments and attack information from the AKB

**Subject Matter Experts (SMEs):**
System security engineering (SSE) support during lifecycle, e.g., secure architecture & design, criticality analysis techniques, supply chain assurance (SCRM), SETR criteria, sustainment support, etc.

**Third Party Assessment:**
Assistance in program evaluation and risk assessments, including bitstream analysis, hardware functional verification, static source code analysis, dynamic binary analysis, static binary analysis, web application analysis, database analysis, and mobile application analysis

**Program Protection Plan (PPP) & SSE Planning:**
Assistance with PPP development and the planning of SSE activities and countermeasures, to include SwA and HwA

**Contract Assistance:**
Assist programs with the development of SwA and HwA contract language for RFPs and CDRLs

**Metrics Assistance:**
Assist programs with the identification, benchmarking, and collection of SwA and HwA related metrics (contract, progress, TPMs, ...)

Distribution Statement A: Approved for public release. Distribution is unlimited.
Joint Federated Assurance Center (JFAC) Capabilities

Protecting force lethality and increasing resilience through software and hardware assurance

- Assured Design Methods
- Binary Software Analysis
- Physical/Functional Verification
- IC Component Markers
- Source Code Analysis
- Supply Chain Assessments
- Technology/Prototype Development and Transitions

- Federated laboratory capability of expertise and tools for vulnerability detection and analysis
- Support program offices with software and hardware assurance expertise and capabilities

https://jfac.navy.mil