Cybersecurity Technical Risk Indicators:

A Measure of Technical Debt in Software Supply Chain Risk Management

Joe Jarzombek, USAF Lt Col (Retired), CSSLP, PMP
Global Manager, Software Supply Chain Solutions
Synopsys Software Integrity Group

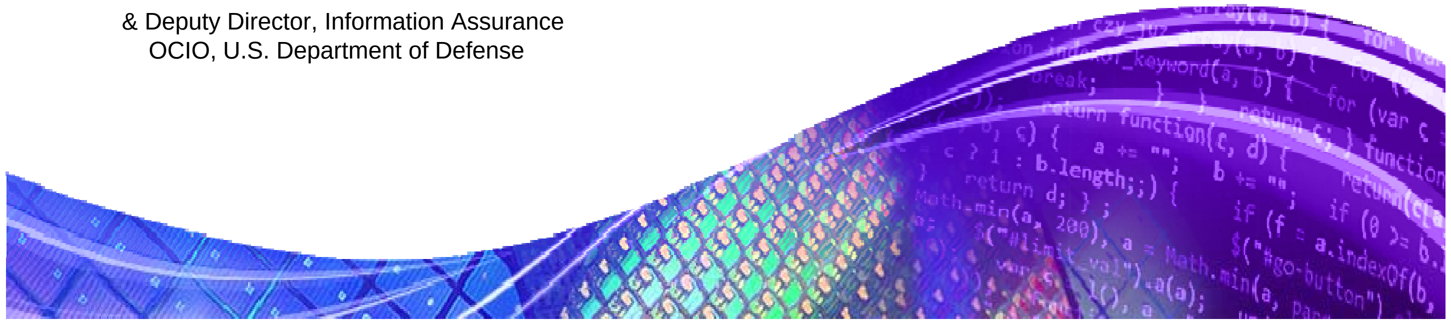
Joe.Jarzombek@synopsys.com
+1 (703) 627-4644

Previously

Director, Software and Supply Chain Assurance
U.S. Department of Homeland Security

& Deputy Director, Information Assurance
OCIO, U.S. Department of Defense

PSM Users Group 14 June 2017



Indicators of Software Risk – A Form of Technical Debt

Code Analysis & Bill of Materials Results with Policy Element Count

- OWASP Top 10 Issues (CWEs & CVEs)
- CWE/SANS Top 25 Issues (CWEs)
- CWE/SANS On the Cusp (26-41) Issues
- Issues with CWE IDs
- Other issues (weaknesses without IDs)
- Known Vulnerabilities (from CVEs)
- Critical Vulnerabilities (7+ on CVSS)
- Types of Licenses
- Components with unconfirmed pedigree

Links urgency of CVE patch and/or CWE Negative Technical Impacts with Business Risks

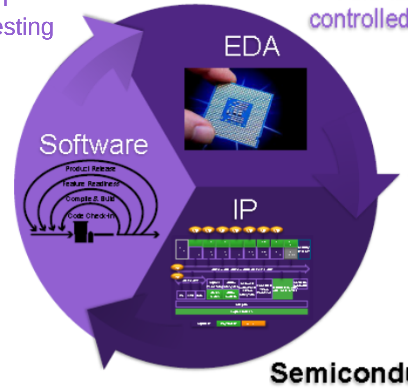
Technical Risk Indicators (Count of Elements,) that if left unmitigated, represent or could contribute to:

- Denial of Service
- Unauthorized Bypass of Protection Mechanism
- Unauthorized Gain of Privileges /Assumption of Identity
- Execution of Unauthorized Code or Command
- Unauthorized Alteration of Execution Logic
- Unauthorized Modification of data, files, directories or memory
- Information leakage or unauthorized reading of data, files, directories or memory
- Hiding of Activities
- Degradation of Quality
- Unexpected State or other Technical Risk

Who is Synopsys?

Software Integrity
Gartner "Leader" in Application Security Testing

Electronic Design Automation
Anti-tampering, controlled distribution



Semiconductor IP
Standards, testing

Top 20 Global Software Companies

1	Microsoft
2	Oracle
3	SAP
4	Symantec
5	VMware
6	Salesforce
7	Intuit
8	CA Tech
9	Adobe
10	Teradata
11	Amdocs
12	Cerner
13	Citrix
14	Autodesk
15	**Synopsys**
16	Sage Group
17	Akamai Tech
18	Nuance
19	Open Text
20	F5 Networks

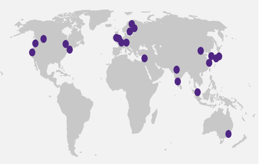
Financial Snapshot

2015 Revenue:
$2.242B

3-Yr Backlog:
$3.6B

FY15 Operating Cash Flow:
$495M

Global Reach



Engineering Culture

Total Employees:	**~11,000**
Engineers:	**~50%**
Software Integrity Group:	**~1000**

'Building Security in' from Silicon to Software

An ever-more connected world . . .

Goods & Services

- Track materials
- Speed distribution
- Product feedback

People

- Wellness monitoring
- Medical case management
- Social needs

Communities

- Traffic status
- Pollution alerts
- Infrastructure checks

Environment

- Pollution checks
- Resource status
- Water monitoring

Homes

- Utilities control
- Security monitoring
- Structure integrity

Technology complexity creates vulnerabilities.
Interdependencies and supply chain risks abound.

Cyber Risks and Consequences in IoT Solutions
Creating More Attack Vectors

- **Edge Devices (including Applications, Sensors, Actuators, Gateways & Aggregation)**
  - Device Impersonation and Counterfeiting
  - Device Hacking
  - Snooping, Tampering, Disruption, Damage
- **IoT Platform (Data Ingestion/Analytics, Policy/Orchestration, Device/Platform Mgmt)**
  - Platform Hacking
  - Data Snooping & Tampering
  - Sabotaging Automation & Devices
- **Enterprise (Business/Mission Applications, Business Processes, etc)**
  - Business/Mission Disruption
  - Espionage & Fraud
  - Financial Waste

Growing Concern with Internet of Things (IoT)

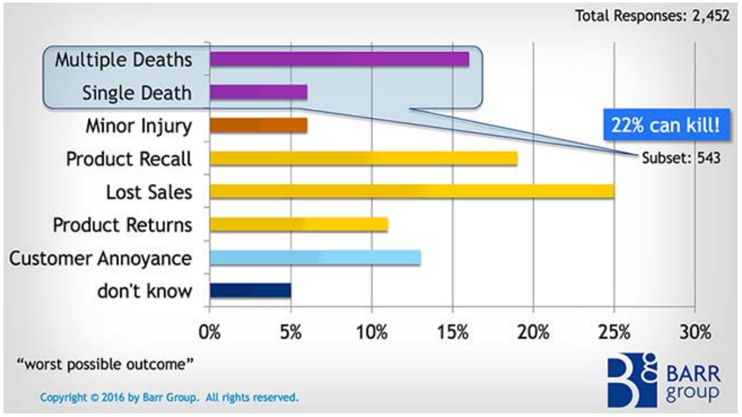
- Lax security without liability for the growing number of IoT embedded devices in appliances, industrial applications, vehicles, smart homes, smart cities, healthcare, medical devices, etc.

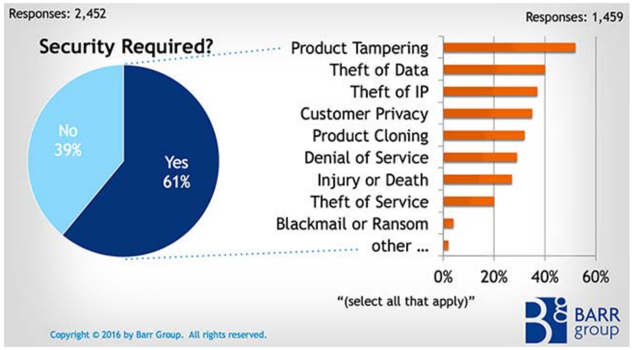
  - Sloppy manufacturing ‘hygiene’ is compromising privacy, safety and security – incurring risks for faster time to market
  - IoT risks provide more source vectors for financial exploitation
  - IoT risks include virtual harm to physical harm
    - Cyber exploitation with physical consequences;
    - Increased risk of bodily harm from hacked devices

Safety/Security Risks with IOT embedded systems

Engineering Community concerns:

- Poorly designed embedded devices can kill;
- Security is not taken seriously enough;
- Proactive techniques for increasing safety and security are used less often than they should be.

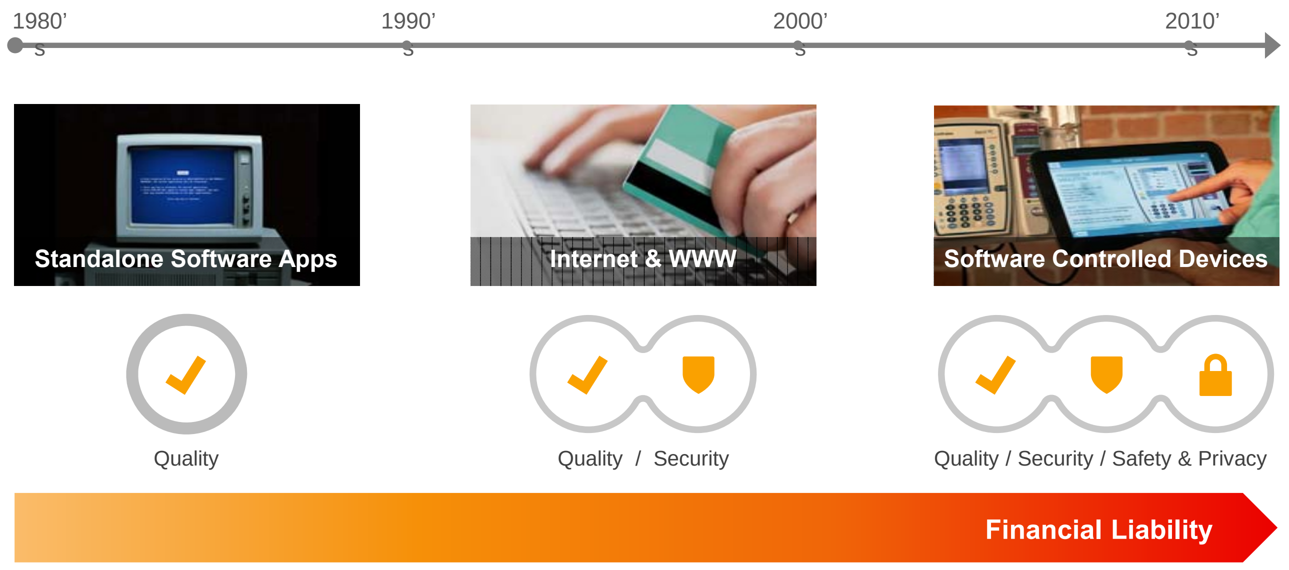




Barr Group: "Industry is not taking safety & security seriously enough"

Based on results of survey of more than 2400 engineers worldwide to better understand the state of safety- and security-aware embedded systems design around the world (Feb 2016).

Shifting Business Concerns: Increased Software Liability



SYNOPSYS®
Silicon to Software™

Software Security Enumerations and Definitions

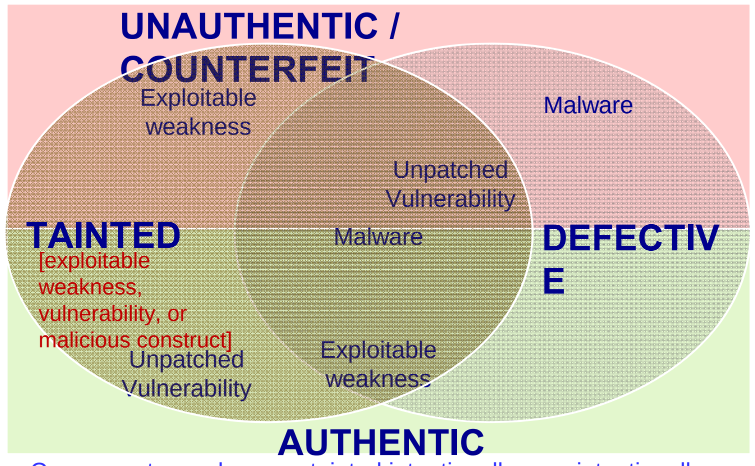
Enabling Standards-based Security Automation & Information Sharing

Software Supply Chain Assurance Focus on Components

Mitigating risks attributable to tainted, exploitable non-conforming constructs in ICT/IoT software

"Tainted" products are corrupted with malware, and/or exploitable weaknesses & vulnerabilities that put enterprises and users at risk

- Enable 'scalable' detection, reporting and mitigation of tainted software components in ICT/IoT
  - Leverage related existing standardization efforts
  - Leverage taxonomies, schema & structured representations with defined observables & indicators for conveying information:
    - o Tainted constructs:
      - ▪ Malicious logic/malware (MAEC) ITU-T X.1546
      - ▪ Exploitable Weaknesses (CWE) ITU-T X.1524
      - ▪ Vulnerabilities (CVE) ITU-T X.1520
    - o Attack Patterns (CAPEC) ITU-T X.1544
- Leverage catalogued diagnostic methods, controls, countermeasures, & mitigation practices
- Use publicly reported weaknesses and vulnerabilities with patches accessible via National Vulnerability Database (NVD) hosted by NIST



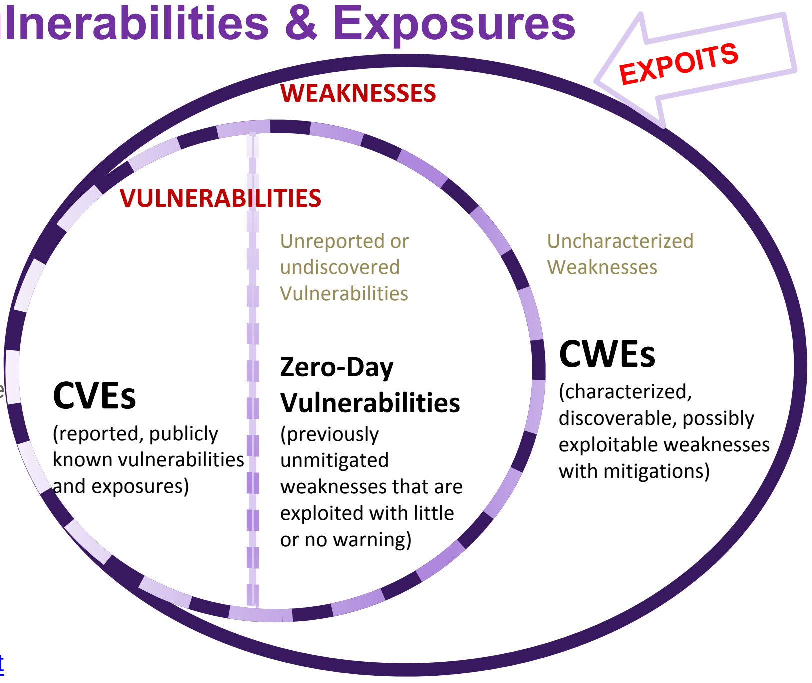
Components can become tainted intentionally or unintentionally throughout the supply chain, SDLC, and in Ops & sustainment

*Text demonstrates *examples* of overlap

International uptake in security automation standards via ITU-T CYBEX 1500 series

Exploits, Weaknesses, Vulnerabilities & Exposures

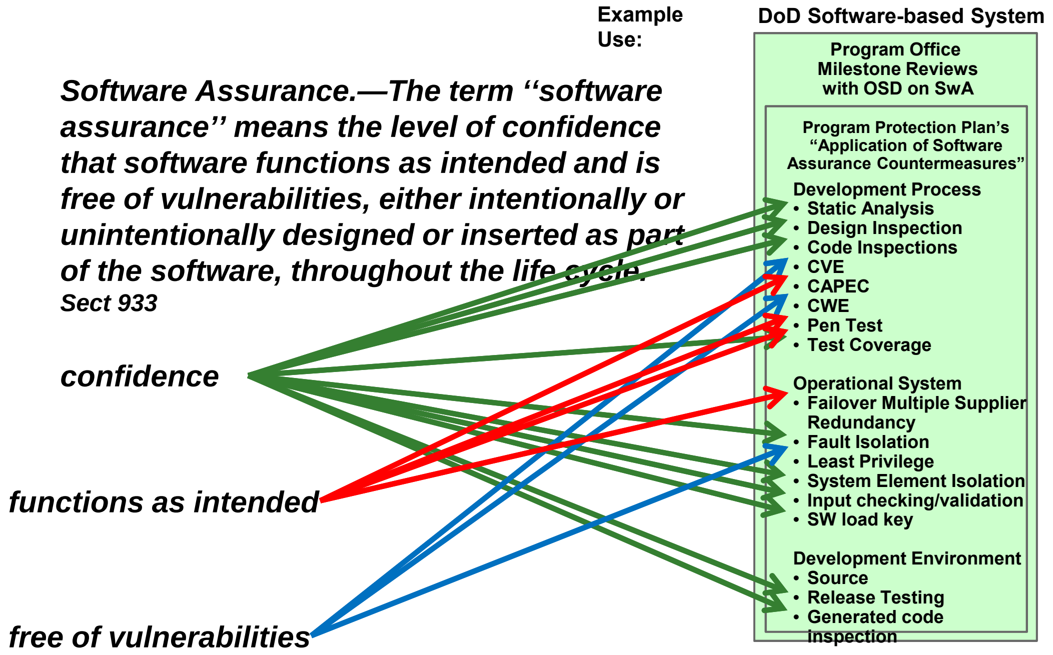
- ***The existence of an exploit designed to take advantage of a weakness (or multiple weaknesses) and achieve a negative technical impact is what makes a weakness a vulnerability.***
- **Weakness:** mistake or flaw condition in ICT/IoT architecture, design, code, or process that, if left unaddressed, could under the proper conditions contribute to a cyber-enabled capability being vulnerable to exploitation; represents potential source vectors for zero-day exploits -- Common Weakness Enumeration (CWE) https://cwe.mitre.org/
- **Vulnerability:** mistake in software that can be directly used by a hacker to gain access to a system or network; **Exposure:** configuration issue of a mistake in logic that allows unauthorized access or exploitation – Common Vulnerability and Exposure (CVE) https://cve.mitre.org/
- **Exploit:** action that takes advantage of weakness(es) to achieve a negative technical impact -- attack approaches from the set of known exploits are used in the Common Attack Pattern Enumeration and Classification (CAPEC) https://capec.mitre.org


EXPOITS
WEAKNESSES
VULNERABILITIES
Unreported or undiscovered Vulnerabilities
Uncharacterized Weaknesses
CVEs
(reported, publicly known vulnerabilities and exposures)
Zero-Day Vulnerabilities
(previously unmitigated weaknesses that are exploited with little or no warning)
CWEs
(characterized, discoverable, possibly exploitable weaknesses with mitigations)

Part of the ITU-T CYBEX 1500 series & USG SCAP

CNSS & Public Law 113-239 “Section 933 - Software Assurance”

Software Assurance.—The term ‘‘software assurance’’ means the level of confidence that software functions as intended and is free of vulnerabilities, either intentionally or unintentionally designed or inserted as part of the software, throughout the life cycle.
Sect 933



Example Use:

DoD Software-based System

Program Office Milestone Reviews with OSD on SwA

Program Protection Plan’s “Application of Software Assurance Countermeasures”

Development Process
- Static Analysis
- Design Inspection
- Code Inspections
- CVE
- CAPEC
- CWE
- Pen Test
- Test Coverage

Operational System
- Failover Multiple Supplier Redundancy
- Fault Isolation
- Least Privilege
- System Element Isolation
- Input checking/validation
- SW load key

Development Environment
- Source
- Release Testing
- Generated code inspection

DoD Program Protection Plan (PPP) Software Assurance Methods

Countermeasure Selection

Development Process
Apply assurance activities to the procedures and structure imposed on software development

Operational System
Implement countermeasures to the design and acquisition of end-item software products and their interfaces

Development Environment
Apply assurance activities to the environment and tools for developing, testing, and integrating software code and interfaces

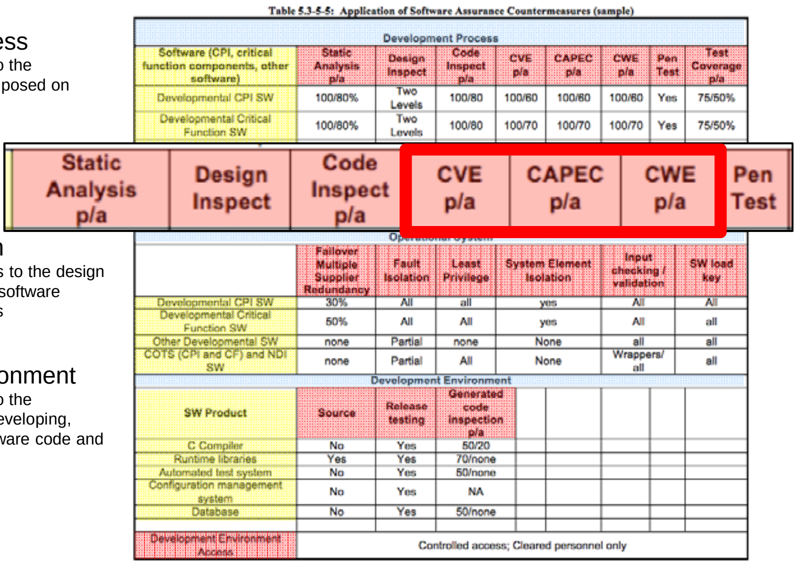


Table 5.3-5-5: Application of Software Assurance Countermeasures (sample)

Development Process								
Software (CPI, critical function components, other software)	Static Analysis p/a	Design Inspect	Code Inspect p/a	CVE p/a	CAPEC p/a	CWE p/a	Pen Test	Test Coverage p/a
Developmental CPI SW	100/80%	Two Levels	100/80	100/60	100/60	100/60	Yes	75/50%
Developmental Critical Function SW	100/80%	Two Levels	100/80	100/70	100/70	100/70	Yes	75/50%

Static Analysis p/a	Design Inspect	Code Inspect p/a	CVE p/a	CAPEC p/a	CWE p/a	Pen Test

Operational System						
	Failover Multiple Supplier Redundancy	Fault Isolation	Least Privilege	System Element Isolation	Input checking / validation	SW load key
Developmental CPI SW	30%	All	all	yes	All	All
Developmental Critical Function SW	50%	All	All	yes	All	all
Other Developmental SW	none	Partial	none	None	all	all
COTS (CPI and CF) and NDI SW	none	Partial	All	None	Wrappers/ all	all

Development Environment			
SW Product	Source	Release testing	Generated code inspection p/a
C Compiler	No	Yes	50/20
Runtime libraries	Yes	Yes	70/none
Automated test system	No	Yes	50/none
Configuration management system	No	Yes	NA
Database	No	Yes	50/none
Development Environment Access	Controlled access; Cleared personnel only		

Additional Guidance in PPP Outline and Guidance

SS/KPP Cyber Survivability

Vol II "Risk-Managed Performance Measures for System Survivability: Ten Cyber Survivability Attributes"

- CSA-10 – Actively Manage System Configurations to Counter Vulnerabilities at Tactically Relevant Speeds
- CSA-06 – Minimize and Harden Attack Surfaces (MHAS)
  - Technical Discussion: Exposed attack surfaces are hardened through use of more secure component and processes, e.g., secure operating systems, trusted or assured hardware and software components, etc. (eliminate vulnerabilities to reduce attack surfaces)
  - Resist Attack Effects (Harden): Vulnerabilities are reduced through hardware and software assurance processes, verification, and testing (through use of appropriate attack trees), and other supply chain and engineering practices.

VDC Research
Insights for the Connected World

Skyrocketing Costs of Aerospace & Defense Systems Failure Fuel Security-Focused Design Practices

June 2017

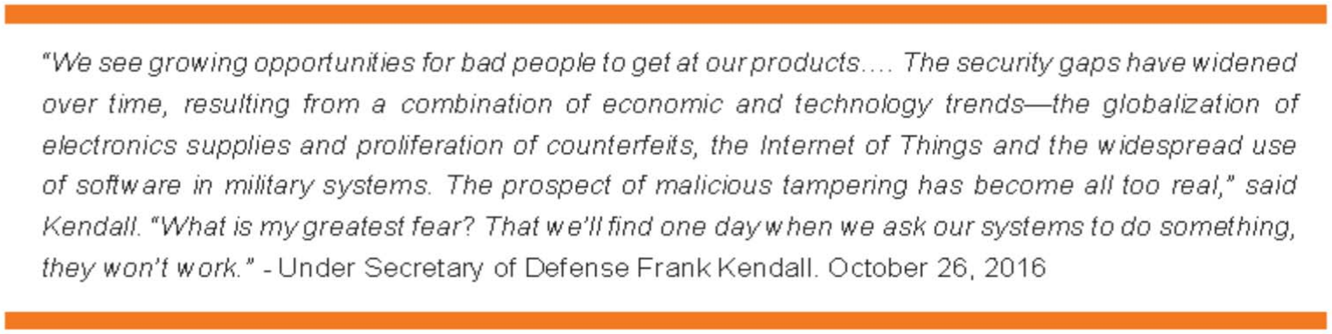
License to Distribute: Synopsys
By Chris Rommel, Executive Vice President

With today's proliferation of asymmetric cyberattack and exploitation, any claims of system safety or reliability must include considerations for the security of software that enables and controls system functionality.

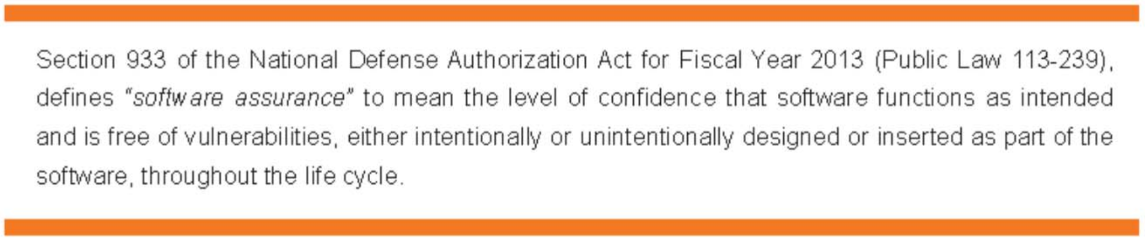
To safeguard one's own strategic interests, all ecosystem constituents must reevaluate both their own development security assurance processes as well as those of their partners and suppliers.

[VDC A&D Report]

Skyrocketing Costs of Aerospace & Defense Systems Failures Fuel Security-Focused Design Practices

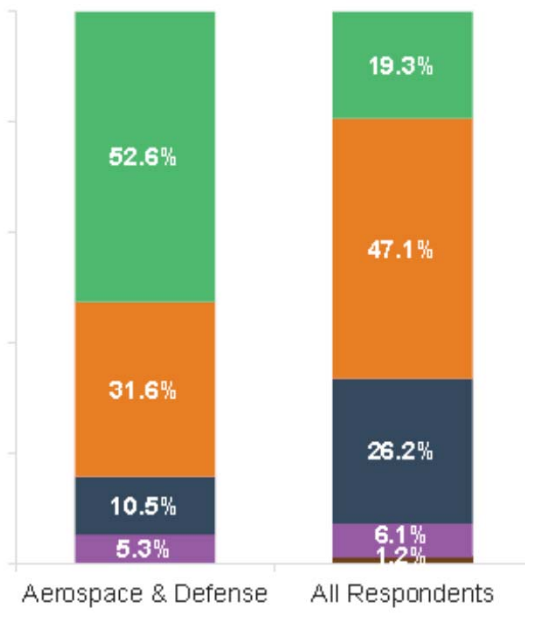


"We see growing opportunities for bad people to get at our products…. The security gaps have widened over time, resulting from a combination of economic and technology trends—the globalization of electronics supplies and proliferation of counterfeits, the Internet of Things and the widespread use of software in military systems. The prospect of malicious tampering has become all too real," said Kendall. "What is my greatest fear? That we'll find one day when we ask our systems to do something, they won't work." - Under Secretary of Defense Frank Kendall. October 26, 2016



Section 933 of the National Defense Authorization Act for Fiscal Year 2013 (Public Law 113-239), defines *"software assurance"* to mean the level of confidence that software functions as intended and is free of vulnerabilities, either intentionally or unintentionally designed or inserted as part of the software, throughout the life cycle.

Skyrocketing Costs of Aerospace & Defense Systems Failures Fuel Security-Focused Design Practices



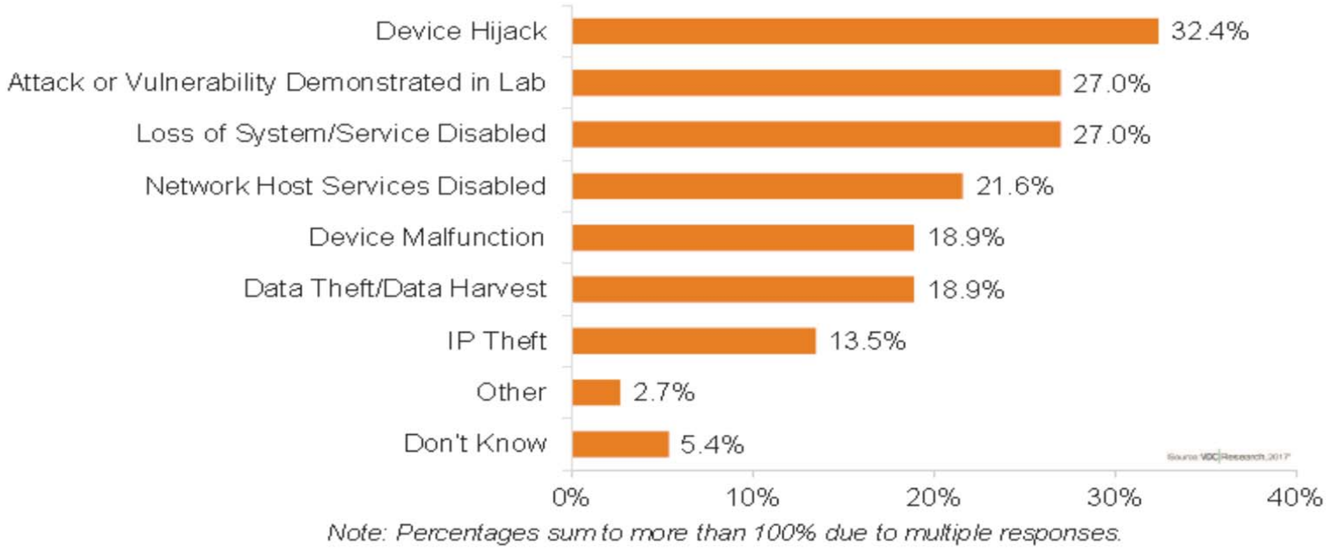
Reason Security is Not Important for Current Project/Program (% of respondents)

- 🟩 Device Is Unconnected
- 🟧 Embedded Data/Functionality Deemed Insensitive
- ■ Security Expected to be Added by End Users
- 🟪 Other
- 🟫 Don't know

Source: VDC Research 2017

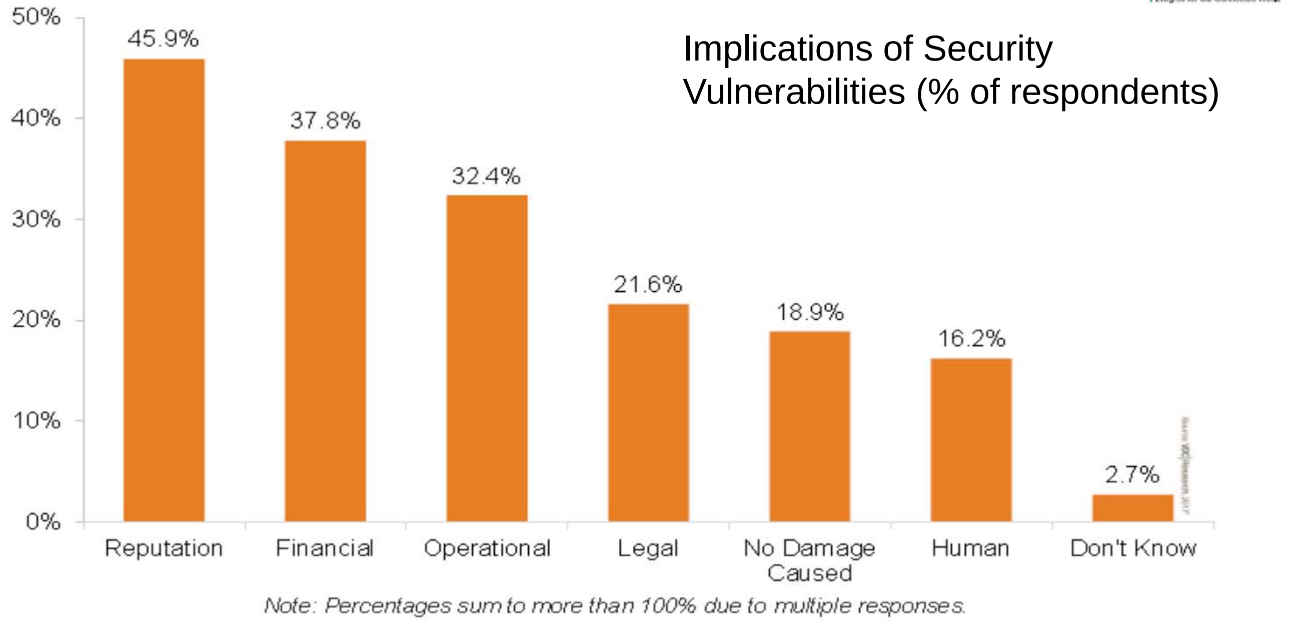
Skyrocketing Costs of Aerospace & Defense Systems Failures Fuel Security-Focused Design Practices

Results of Security Vulnerabilities (% of all respondents)



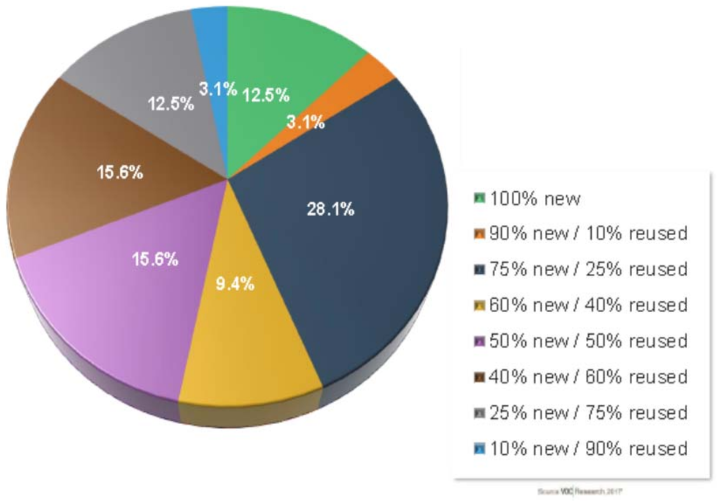
Note: Percentages sum to more than 100% due to multiple responses.

Skyrocketing Costs of Aerospace & Defense Systems Failures Fuel Security-Focused Design Practices

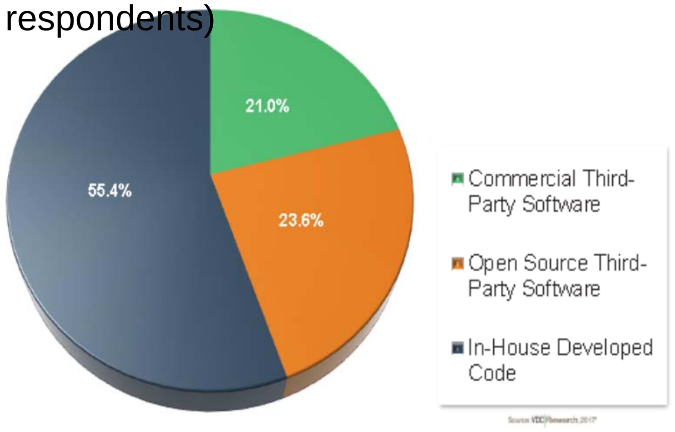


Skyrocketing Costs of Aerospace & Defense Systems Failures Fuel Security-Focused Design Practices

Code Reuse as % of Total In-House Code (% of A&D respondents)



% of Total Software Code in Final Design by Origin (Avg of A&D respondents)



Software Supply Chain Management

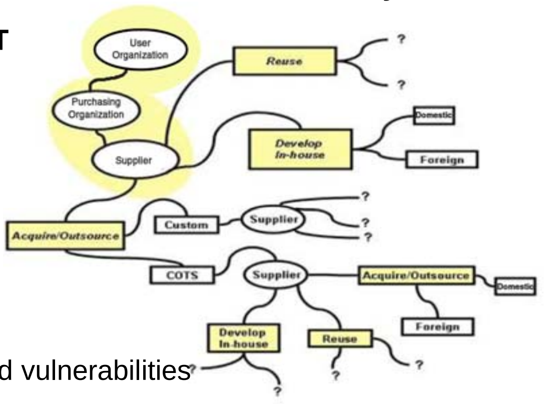
Enabling Enterprise Control of Risks Attributable to Exploitable Software

Software Supply Chain Risk Management Imperative

Increased risk from supply chain due to:

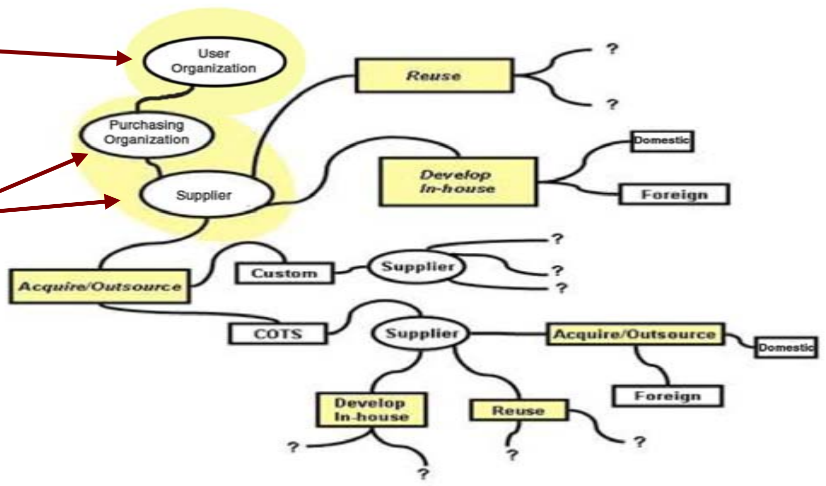
- **Increasing dependence on commercial ICT/IoT for enterprise business/mission critical systems**
- **Increasing reliance on globally-sourced software for ICT/IoT**
  - Varying levels of development/outsourcing controls
  - Lack of transparency in process chain of custody
  - Varying levels of acquisition 'due-diligence"
- **Residual risk passed to end-user enterprise**
  - Defective and Unauthentic/Counterfeit products
  - Tainted products with malware, exploitable weaknesses and vulnerabilities
- **Growing technological sophistication among adversaries**
  - Internet enables adversaries to probe, penetrate, and attack remotely
  - Supply chain attacks can exploit products and processes throughout the lifecycle

Software in the supply chain is often the vector of attack



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

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

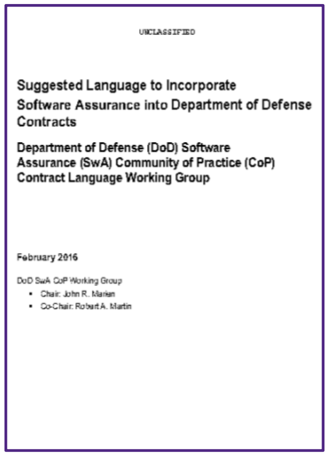


Who makes risk decisions?
Who determines ‘fitness for use’ for ‘technically acceptable’ criteria?
Who “owns” residual risk from tainted/counterfeit products?

* “Tainted” products are those that are corrupted with malware, or exploitable weaknesses & vulnerabilities

Putting Software Assurance into Defense Contracts

- US DoD SwA Community of Practice guidance provides help to defense agencies and military services to put Software Assurance on Contact.
- This document suggests language that may be tailored for use in Request for Proposal (RFP) packages and contracts to provide a government program office insights into the software development activities of its contractors and to provide assurance regarding developed software and its ability to meet mission needs.
- This language will generally appear in Sections C, I, L, and M of the standard format RFP and contract.


UNCLASSIFIED

Suggested Language to Incorporate Software Assurance into Department of Defense Contracts

Department of Defense (DoD) Software Assurance (SwA) Community of Practice (CoP) Contract Language Working Group

February 2016

DoD SwA CoP Working Group
- Chair: John R. Marien
- Co-Chair: Robert A. Martin

*Common industry practice and Section 933 of the National Defense Authorization Act for Fiscal Year 2013 (Public Law 113-239), define "software assurance" to mean the level of confidence that software functions as intended and is free of vulnerabilities, either intentionally or unintentionally designed or inserted as part of the software, throughout the life cycle.

Majority of Breaches Attributable to Exploitable Software

Data Breaches make headlines – the cause of them rarely do

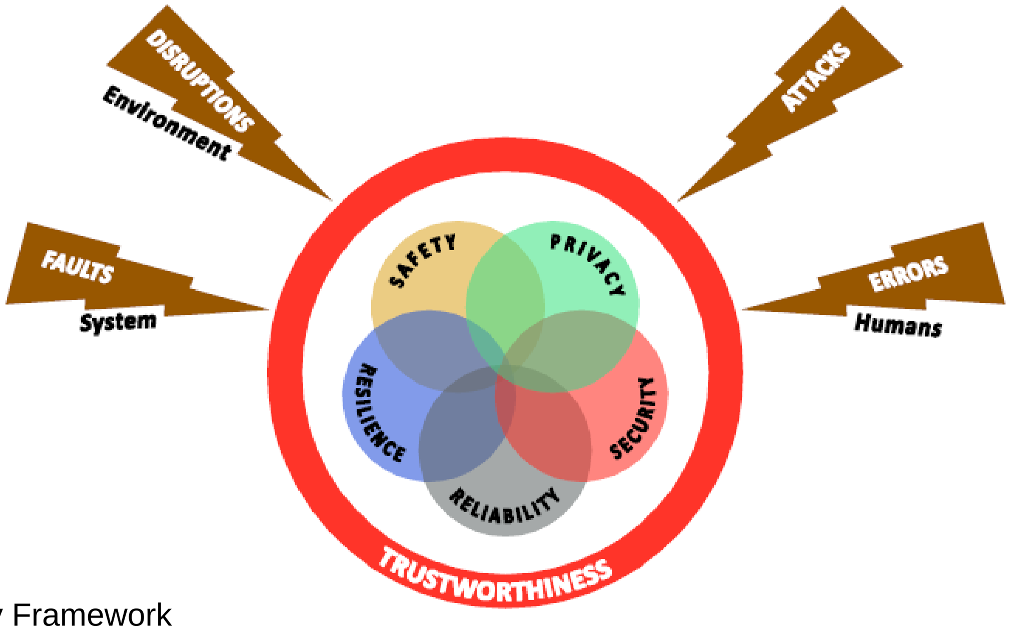
- ✓ **Over 70 % of security breaches happen at the Application** (Gartner)
- ✓ 92% of vulnerabilities are in application layer (NIST)
- ✓ **Up to 80% of of Data Breaches originate in the Supply Chain** (SANs Institute)
- ✓ More than 80% of Enterprises depend on third-party code (Gartner)
- ✓ **90% of a typical application is comprised of third-party / OSS components** (SANS)
- ✓ Most developers lack sufficient security training (Gartner)
- ✓ **Web Application Attacks are the #1 source of data breaches** (Verizon DBIR 2016)

Data breaches exploit vulnerabilities and weaknesses in applications -- root causes in unsecure software

This is a Software Supply Chain Issue

Trustworthiness of an IIoT System

Trustworthiness is the degree of confidence one has that the system performs as expected in respect to *all* the key system characteristics in the face of environmental disruptions, human errors, system faults and attacks.



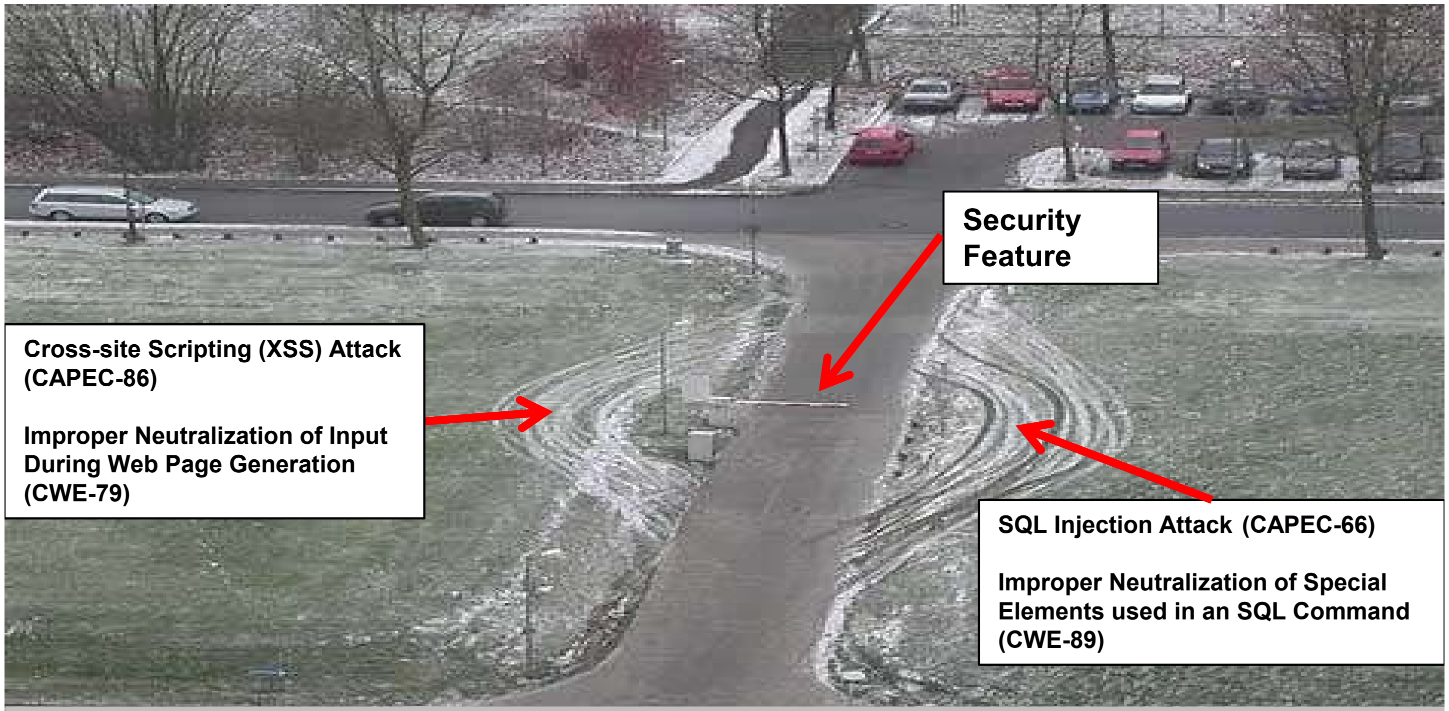
Source: IIC IIoT Security Framework

Enterprises Have Used Reactive Technologies to Defend…

They are good; designed for known threats. What about broader risks to enterprises and users?

eset
AhnLab
McAfee
avast!
PANDA
G DATA
Quick Heal
Norton from symantec
RISING
F-SECURE
TREND MICRO
webroot
ca
bitdefender
Sunbelt Software
KASPERSKY
AVG
INTRUSION DETECTION & PREVENTION
Carl Endorf, Eugene Schultz, and Jim Mellander
VPN!
Certified
CRYPTO DATA
Vulnerability Scanner Overview
Load Balancer
SIEM
Security Information and Event Managment

Enterprises cannot stop the threats; yet can control their attack vectors/surfaces


Security Feature
Cross-site Scripting (XSS) Attack (CAPEC-86)
Improper Neutralization of Input During Web Page Generation (CWE-79)
SQL Injection Attack (CAPEC-66)
Improper Neutralization of Special Elements used in an SQL Command (CWE-89)

Exploitable Software Weaknesses (CWEs) are exploit targets/vectors for future Zero-Day Attacks

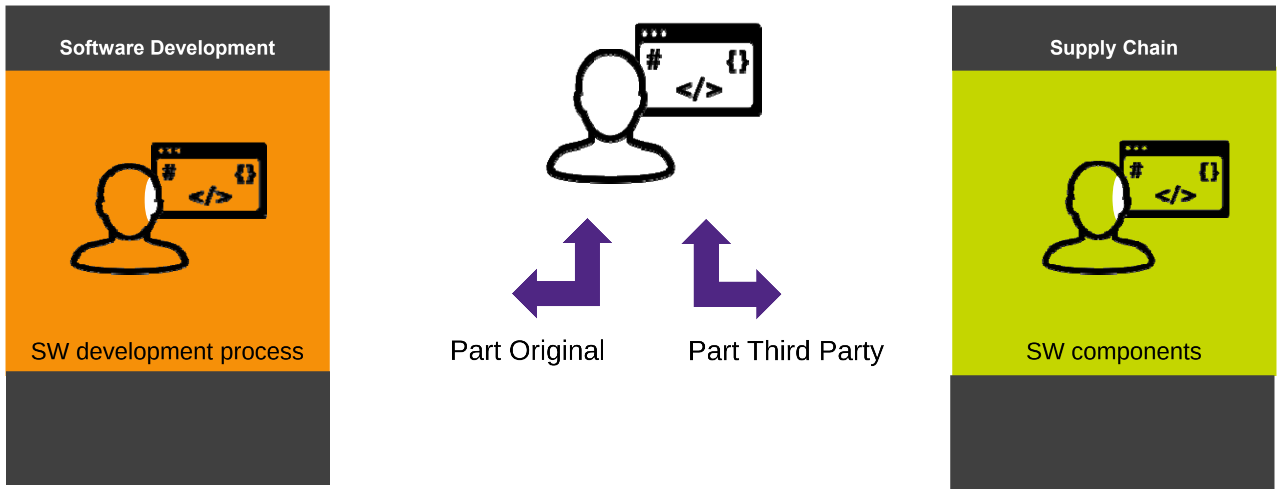
Products on “Whitelisted” Approved Products List or “Assessed & Cleared” Products List should be Tested for…

- **Exploitable Weaknesses (CWEs, ITU-T X.1524)**
  - If suppliers do not mitigate exploitable weaknesses or flaws in products (which are difficult for users to mitigate), then those weaknesses represent vectors of future of exploitation and ‘zero day’ vulnerabilities.
- **Known Vulnerabilities (CVEs, ITU-T X.1520)**
  - If suppliers cannot mitigate known vulnerabilities prior to delivery and use, then what level of confidence can anyone have that patching and reconfiguring will be sufficient or timely to mitigate exploitation?
- **Malware (MAEC, ITU-T X.1546)**
  - If suppliers do not check that the software they deliver does not have malware (typically signature-based), then users and using enterprises are at risk of whitelisting the malware.

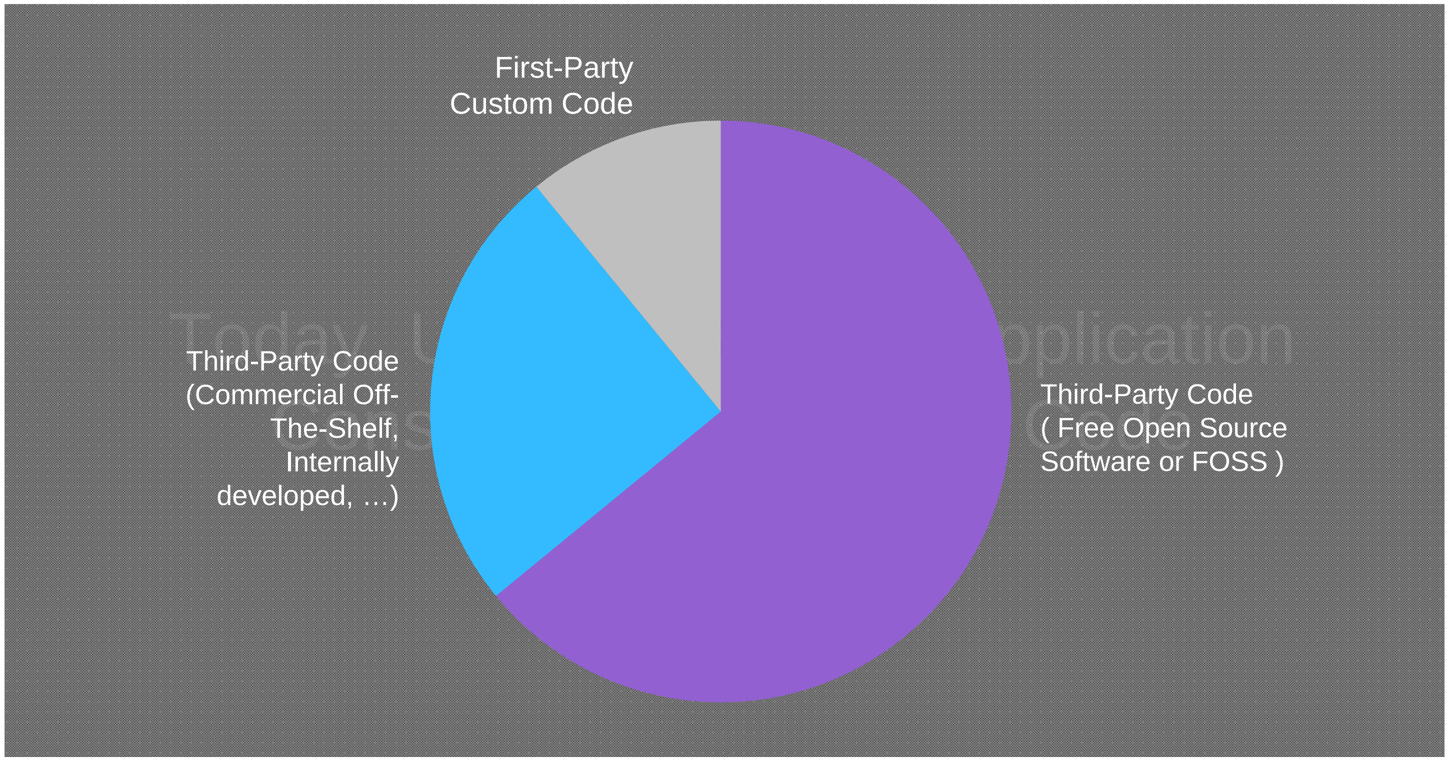
Software Testing

Enabling Insight into Risks Attributable to Exploitable Software

Software Today Is Assembled



Today, Up to 90% of an Application Consists of Third-Party Code


First-Party Custom Code
Third-Party Code (Commercial Off-The-Shelf, Internally developed, …)
Third-Party Code (Free Open Source Software or FOSS)

Do you trust what’s in your Third-Party Code?

Why test your software?

- **Software is buggy, only question is how many bad and exploitable known or unknown bugs are out there?**
- **Hackers use binary analysis and fuzzing techniques to find vulnerabilities**
  - Found vulnerabilities are used to develop exploits or launch DOS attacks
- **Any software processing input can be attacked: network interfaces, device drivers, user interface, etc…..**

Types of Automated Tools/Testing

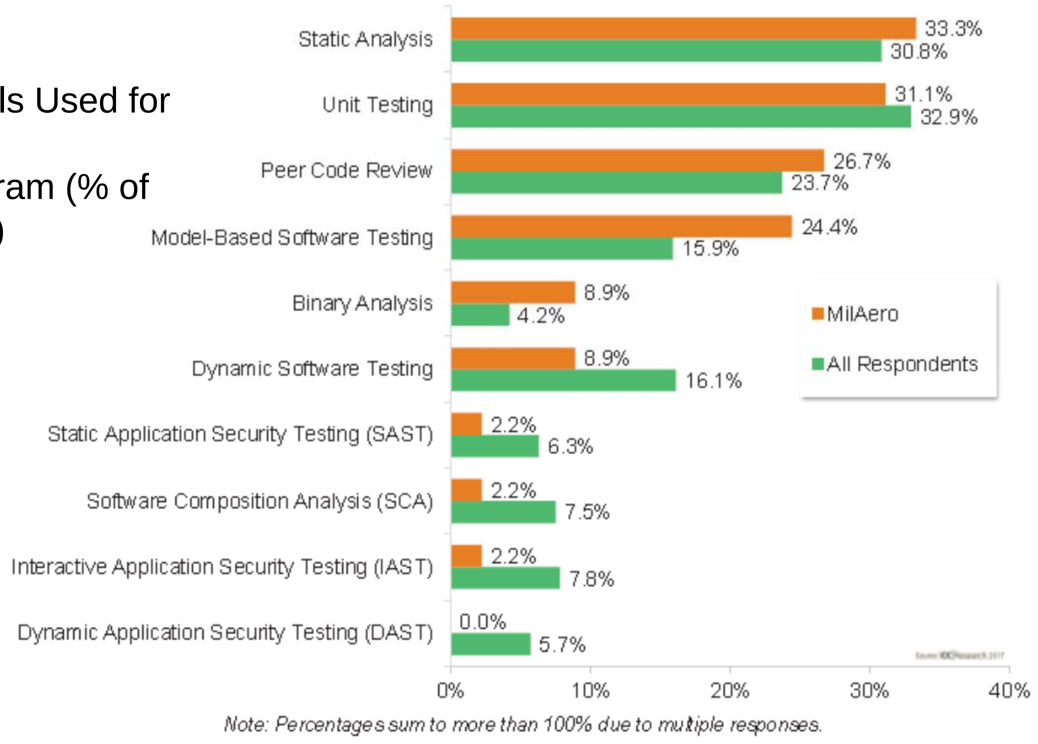
What They Find; How They Support Origin Analysis & Risk Management

- **Dynamic Runtime Analysis** – Finds security issues during runtime, which can be categorized as CWE’s
  - **Malformed input testing** (fuzz testing, DoS testing) – Finds zero-days and robustness issues through negative testing.
  - **Behavioral analysis** – Finds exploitable weaknesses by analyzing how the code behaves during “normal” runtime.
- **Software Composition Analysis** – Finds known weaknesses and categorizes them as CVE’s and other issues.
- **Static Code Analysis** – Finds defects in source code and categorizes them as CWE’s.
- **Known Malware Testing** – Finds known malware (e.g. viruses and other rogue code).

These tests can be used to enumerate CVE’s, CWE’s, and malware which can be further categorized into prioritized lists.

VDC Research
Insights for the Connected World

Types of Tools Used for Current Project/Program (% of respondents)

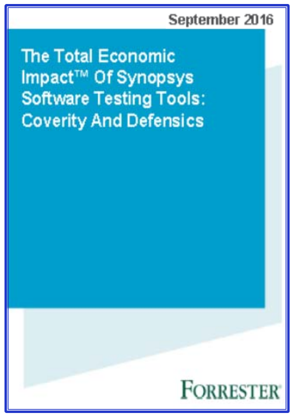


Note: Percentages sum to more than 100% due to multiple responses.

SYNOPSYS®

Total Economic Impact of Synopsys Software Testing Tools

Forrester Case Study – Useful Framework


September 2016

The Total Economic Impact™ Of Synopsys Software Testing Tools: Coverity And Defensics

Using Coverity and Defensics in the development lifecycle…

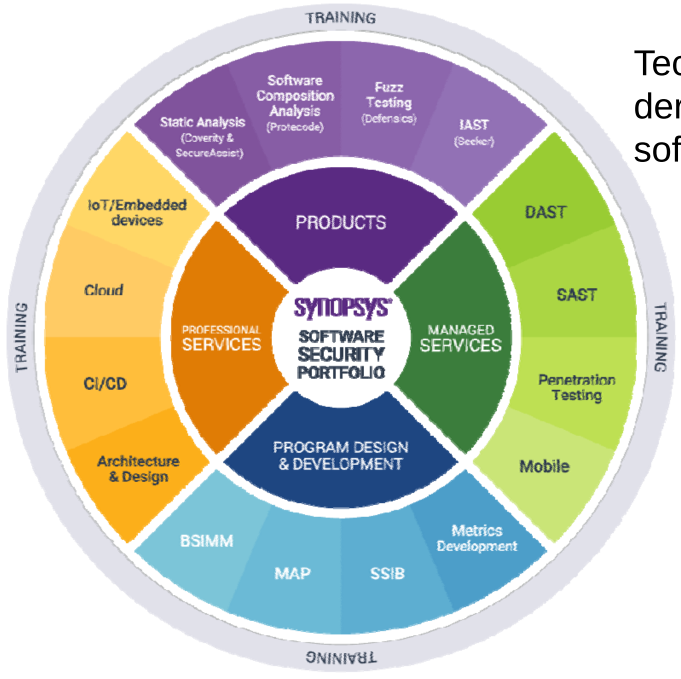
- **Improved product quality and security**
  - Avoided remediation expenses in 8 code bases of 1.5M LoC each; saving $3.86M (NPV)
  - Lowered defect density within its code base…
    prevented future costs of allowing error-prone code to be reused.
- **Reduced time to market**
  - Using fuzz testing and static analysis, reduced product release cycle from 12 to 8 months; enabling company to redirect resources toward other productive activities.
  - Decreased time to detect and remediate defects/vulnerabilities;
- **Prevented high-profile breaches**
  - Lowered future risk exposure attributable to exploitable software
- **Mitigated costly post-deployment malfunctions**
  - Required 2 times fewer labor hours than in post-release phase

Numerical Data
ROI: **136%** **//** Total NPV: **$5.46m**
Cost to find & fix bugs: ↓**2x-10x**
Time to release new products: ↓**4mo**

Access full report at http://software.synopsys.com/register-for-coveritydefensicsTEIstudy.html

Synopsys Comprehensive Software Integrity Portfolio

The most comprehensive solution for integrating security and quality into your SDLC and supply chain



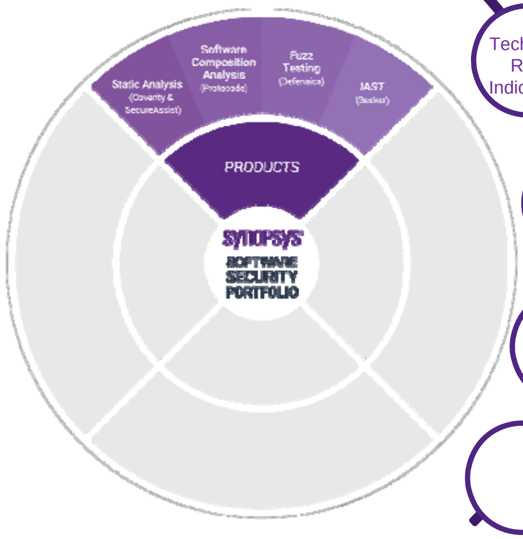
Technical Risk Indicators derived from output of software security tools

Glossary

- Interactive Application Security Testing (**IAST**)
- Dynamic Application Security Testing (**DAST**)
- Static Application Security Testing (**SAST**)
- Building Security In Maturity Model (**BSIMM**)
- Maturity Action Plan (**MAP**)
- Software Security Initiative In-a-Box (**SSIB**)
- Continuous Integration/ Continuous Delivery and Deployment (**CI/CD**)

Managing Risk Across your SDLC with Time-Proven, Industry-Leading Products:

Build in security and quality through automation at every step during development and across the supply chain


Static Analysis
(Coverity & SecureAssist)
Software Composition Analysis
(Protecode)
Fuzz Testing
(Defensics)
IAST
(Seeker)
PRODUCTS
SYNOPSYS
SOFTWARE SECURITY PORTFOLIO

Technical Risk Indicators

Static Analysis
Catch quality defects & security weaknesses in code as written with high accuracy

Technical Risk Indicators

Software Composition Analysis
Discover license compliance issues and known vulnerabilities from binary, open source, and third party code

Fuzz Testing
Bombards system with malformed inputs to trigger unknown vulnerabilities

Interactive Application Security Testing
Simulate actual exploits in the application, verifying results, eliminating false positives

Technical Risk Indicators –
Linking Software Vulnerabilities & Weaknesses with Business/Mission Risks

- ITU-T X.1521 **Common Vulnerability Scoring System (CVSS)** https://www.first.org/cvss is an open framework for communicating the characteristics and severity of software vulnerabilities; providing a standardized method for rating IT vulnerabilities and determining the urgency of response.
- ITU-T X.1525 **Common Weakness Scoring System (CWSS)** https://cwe.mitre.org/cwss/cwss_v1.0.1.html provides a mechanism for prioritizing software weaknesses in a consistent, flexible, open manner.
- **Technical Impact:** derived from CWSS Base Finding metric group that captures the inherent risk of the weakness, a technical impact represents the potential result that can be produced by the weakness, assuming that the weakness can be successfully reached and exploited.
- **Software Security Risk:** using software with known vulnerabilities and weaknesses.
- **Technical Risk Indicator:** derived from vulnerability rating and/or weakness technical impact, an indicator of technical security risk that, if unpatched or unmitigated, could represent a source vector for attack, and if exploited could result in negative business/mission consequences.

Indicators of Software Risk – A Form of Technical Debt

Code Analysis & Bill of Materials Results with Policy Element Count

- OWASP Top 10 Issues (CWEs & CVEs)
- CWE/SANS Top 25 Issues (CWEs)
- CWE/SANS On the Cusp (26-41) Issues
- Issues with CWE IDs
- Other issues (weaknesses without IDs)
- Known Vulnerabilities (from CVEs)
- Critical Vulnerabilities (7+ on CVSS)
- Types of Licenses
- Components with unconfirmed pedigree

Links urgency of CVE patch and/or CWE Negative Technical Impacts with Business Risks

Technical Risk Indicators (Count of Elements,) that if left unmitigated, represent or could contribute to:

- Denial of Service
- Unauthorized Bypass of Protection Mechanism
- Unauthorized Gain of Privileges /Assumption of Identity
- Execution of Unauthorized Code or Command
- Unauthorized Alteration of Execution Logic
- Unauthorized Modification of data, files, directories or memory
- Information leakage or unauthorized reading of data, files, directories or memory
- Hiding of Activities
- Degradation of Quality
- Unexpected State or other Technical Risk

Complete Support Across Your SDLC

TRAINING	REQUIREMENTS & DESIGN	IMPLEMENTATION	VERIFICATION	RELEASE
Core Security Training	Architecture Risk Analysis	SAST (IDE)	SAST (Managed)	DAST (Managed)
Secure Coding Training	Security Code Design Analysis	SAST (Build)	Fuzz Testing	Pen Testing
eLearning	Threat Modeling	SCA (Source)	SCA (Binary)	Network Pen Testing
		IAST	Mobile Testing	

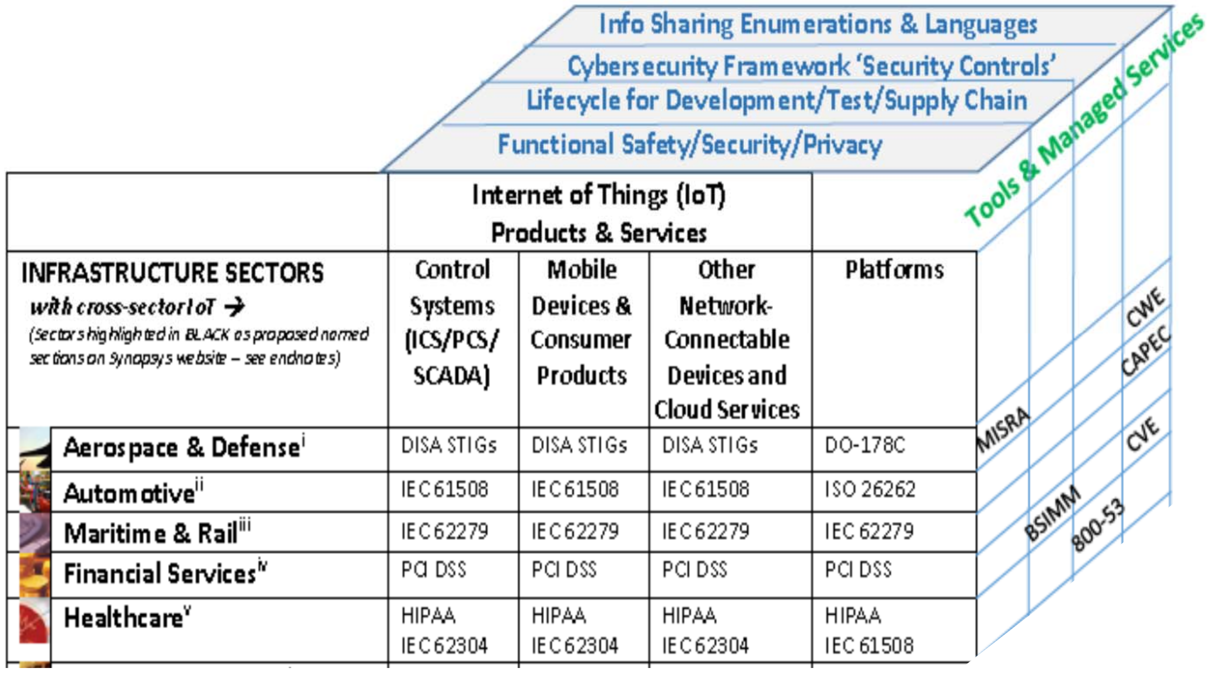
ANY DEVELOPMENT APPROACH

Agile CI/CD DevOps

ANY DEPLOYMENT ENVIRONMENT

Embedde d Cloud Mobile

Linking Software Security with Regulatory Compliance


Info Sharing Enumerations & Languages
Cybersecurity Framework 'Security Controls'
Lifecycle for Development/Test/Supply Chain
Functional Safety/Security/Privacy
Tools & Managed Services
MISRA
CWE
CAPEC
CVE
BSIMM
800-53

	Internet of Things (IoT) Products & Services			
INFRASTRUCTURE SECTORS *with cross-sector IoT →* *(Sectors highlighted in BLACK as proposed named sections on Synopsys website – see endnotes)*	Control Systems (ICS/PCS/ SCADA)	Mobile Devices & Consumer Products	Other Network-Connectable Devices and Cloud Services	Platforms
Aerospace & Defense[i]	DISA STIGs	DISA STIGs	DISA STIGs	DO-178C
Automotive[ii]	IEC61508	IEC61508	IEC61508	ISO 26262
Maritime & Rail[iii]	IEC62279	IEC62279	IEC62279	IEC 62279
Financial Services[iv]	PCI DSS	PCI DSS	PCI DSS	PCI DSS
Healthcare[v]	HIPAA IEC62304	HIPAA IEC62304	HIPAA IEC62304	HIPAA IEC 61508

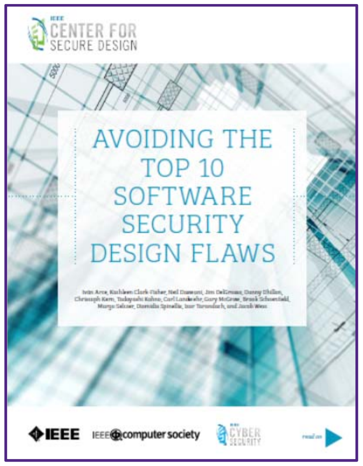
RTCA Special Committee 216 — Aeronautical Systems Security harmonizing RTCA and EUROCAE standards

- RTCA SC-216 Aeronautical Systems Security is concerned with aircraft security at both the systems and aircraft levels.
- Most standards from RTCA are harmonized with EUROCAE and are essentially equivalent; for example, RTCA DO-326A and EUROCAE ED-202A are equivalent for "Airworthiness Security Process Specification".
  - Two outliers were DO-256A and ED-203A both addressing "Airworthiness Security Methods and Considerations." Our committee is working on a common language between these two so they end up as a single document.
  - Then, there will be a complete set of harmonized standards covering software development through aircraft level airworthiness that satisfy both the FAA and EASA, as well as ANAC in Brazil.
- Other relevant standards for aircraft safety are published by SAE, and RTCA work fits those frameworks.

Avoiding the Top 10 Software Security Design Flaws

- Most software built and released with defects — implementation bugs and design flaws
- This shifts some of the focus in security from finding bugs to identifying design flaws in the hope that software architects can learn from others' mistakes.

Free Resource


IEEE CENTER FOR SECURE DESIGN

AVOIDING THE TOP 10 SOFTWARE SECURITY DESIGN FLAWS

Iván Arce, Kathleen Clark-Fisher, Neil Daswani, Jim DelGrosso, Danny Dhillon, Christoph Kern, Tadayoshi Kohno, Carl Landwehr, Gary McGraw, Brook Schoenfield, Margo Seltzer, Diomidis Spinellis, Izar Tarandach, and Jacob West

IEEE IEEE computer society CYBER SECURITY

https://cybersecurity.ieee.org/center-for-secure-design

Thank You

Follow-up Discussion…

Joe Jarzombek – Global Manager, Software Supply Chain Solutions
Joe.Jarzombek@synopsys.com | 703.627.4644
Synopsys Software Integrity Group | www.synopsys.com/software
Software Risk Assessment at http://software.synopsys.com/ssca