IoT Supply Chain Management:

Reducing Attack Vectors & Enabling Cybersecurity Assurance

Joe Jarzombek, CSSLP, PMP
Global Manager, Software Supply Chain Solutions
Synopsys Software Integrity Group

Previously: Director, Software and Supply Chain Assurance, U.S. Department of Homeland Security & Deputy Director, Information Assurance OCIO, U.S. Department of Defense

Joe.Jarzombek@synopsys.com

+1 703.627.4644

NIST IoT Colloquium, Oct 19, 2017

IoT Supply Chain: A context for addressing risks

- The expanding IoT landscape is subject to expanding list of threats, attacks, and corresponding outcomes. Most threats exploit weaknesses and vulnerabilities in IoT devices; introduced in development or in modifying or supporting the devices.
  - These ‘sloppy cyber hygiene’ supply chain practices put users at risk.
  - It is often more about the vulnerability of the users’ devices than the ingenuity of the attackers that contribute to exploitation outcomes.
- External dependencies on others to supply IoT products and services require a supply chain perspective, making it is more important to have means for:
  - Evaluating technical risks based on residual exploitable weaknesses, vulnerabilities and malware in IoT devices;
  - Understanding IoT device ‘patch ability’ and upgradability (and respective roles of users and manufacturers), and
  - Determining ‘fitness for use or purpose’ of devices based on the intended environment in which the devices will be used (based on safety, security, and privacy considerations).

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.

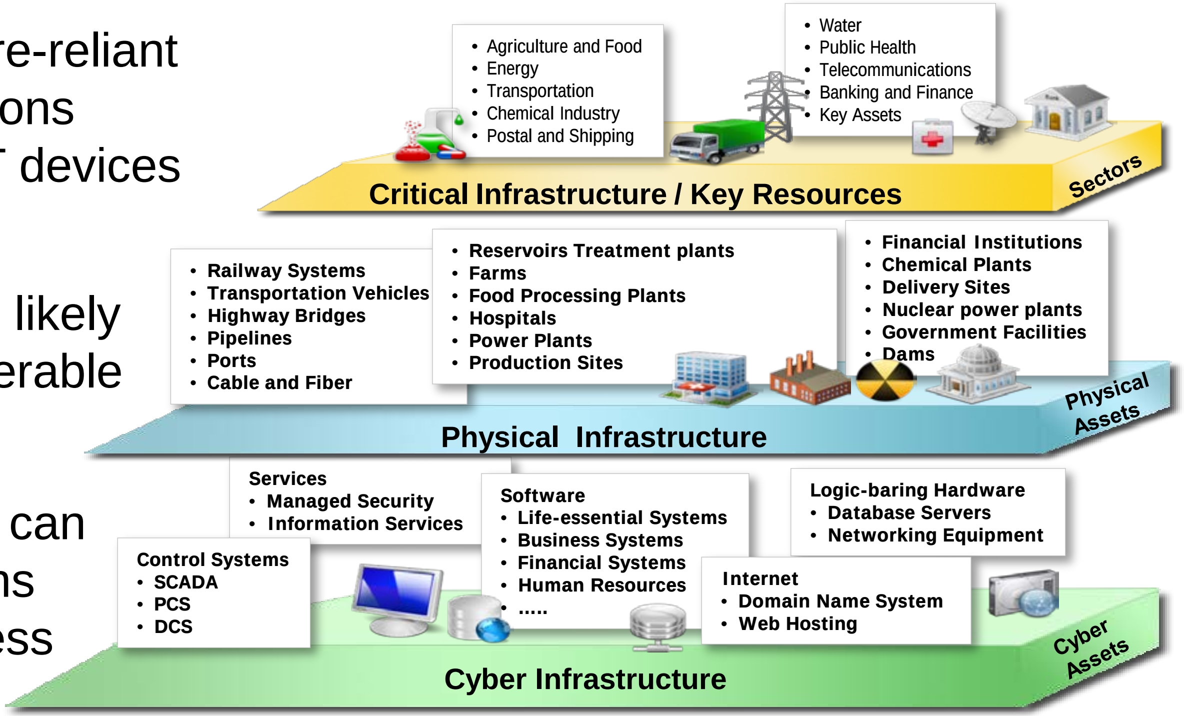
To safeguard one’s own strategic interests, all ecosystem constituents must reevaluate both their own development security assurance processes as well as those of their partners and suppliers.

Gaining confidence in ICT/IoT software-based technologies

- Dependencies on software-reliant Information Communications Technology (ICT) and IoT devices are greater then ever

- Possibility of disruption is likely because software is vulnerable and exploitable

- Loss of confidence alone can lead to stakeholder actions that disrupt critical business activities



Cyber Infrastructure is enabled and controlled by 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

Organizations expanding their IoT efforts need comprehensive security initiatives to address weaknesses resulting from both technological vulnerabilities and a lack of ‘cyber hygiene’ and caution among those who develop and use IoT devices.

Cyber risks and consequences in IoT solutions

Creating more attack vectors via networked devices

- 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)

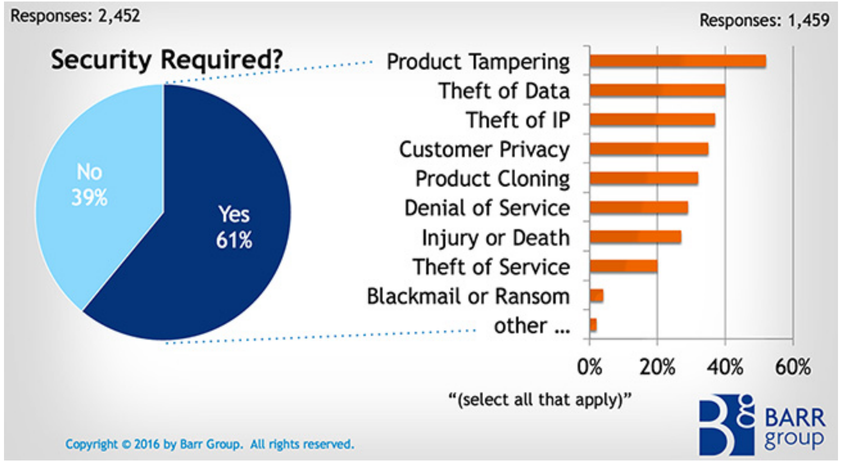
Lax security without accountability for the growing number of IoT embedded devices in appliances, industrial applications, vehicles, smart homes, smart cities, healthcare, medical devices, etc.

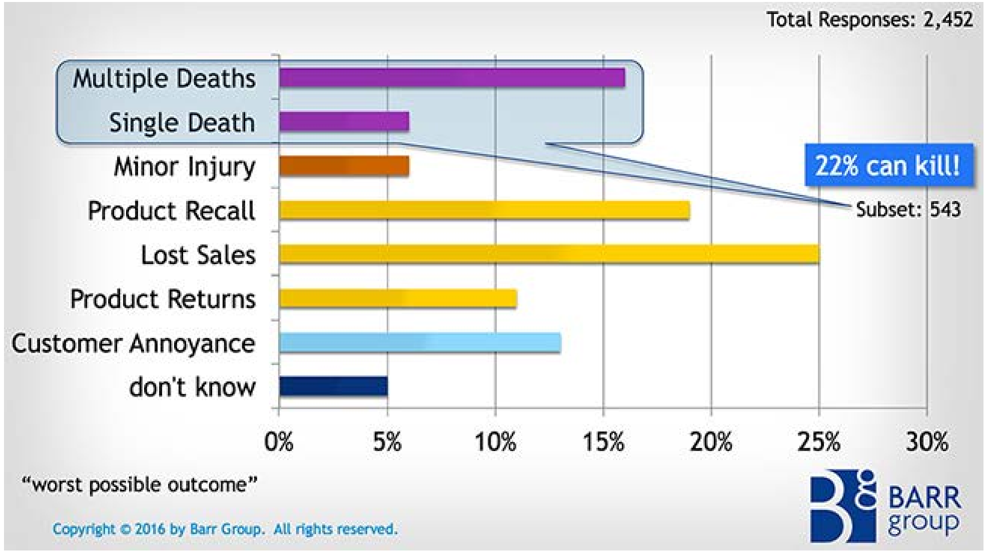
- Sloppy manufacturing ‘hygiene’ is compromising privacy, safety and security – incurring risks for faster time to market
- IoT risks provide more source vectors for financial exploitation
- IoT risks include virtual harm to physical harm
  - Cyber exploitation with physical consequences;
  - Increased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

Software Security Enumerations and Definitions

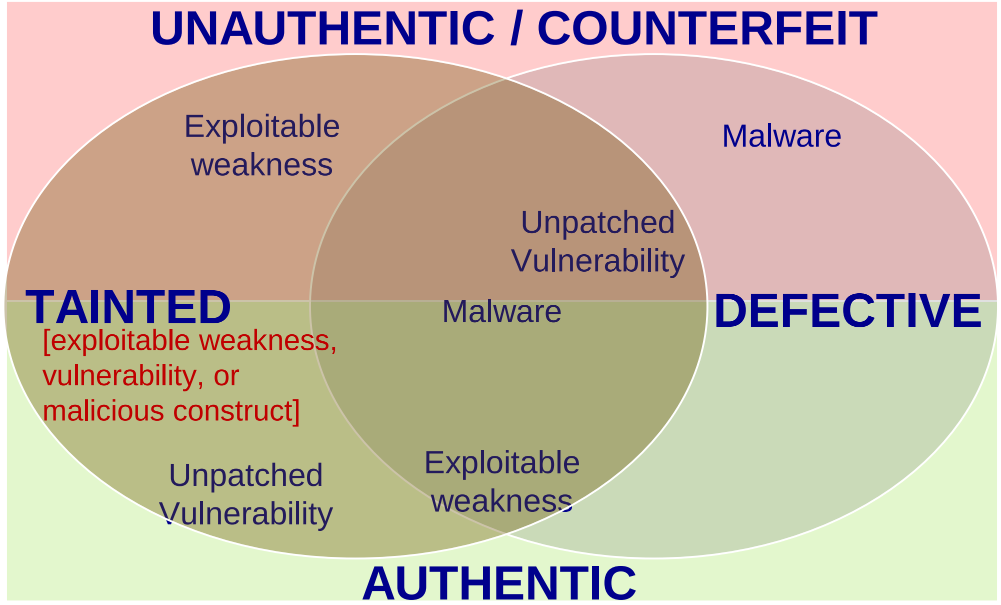
Enabling Standards-based Security Automation & Information Sharing

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



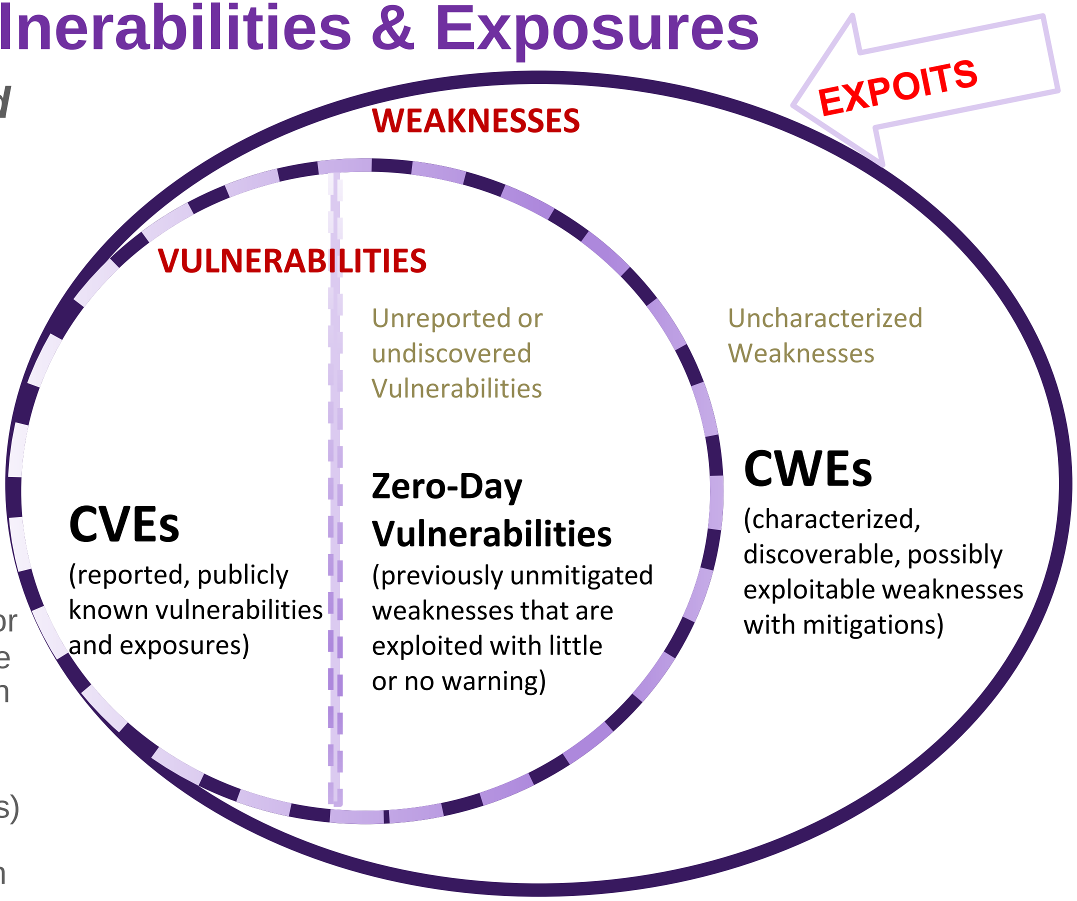
Components can become tainted intentionally or unintentionally throughout the supply chain, SDLC, and in Ops & sustainment

*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

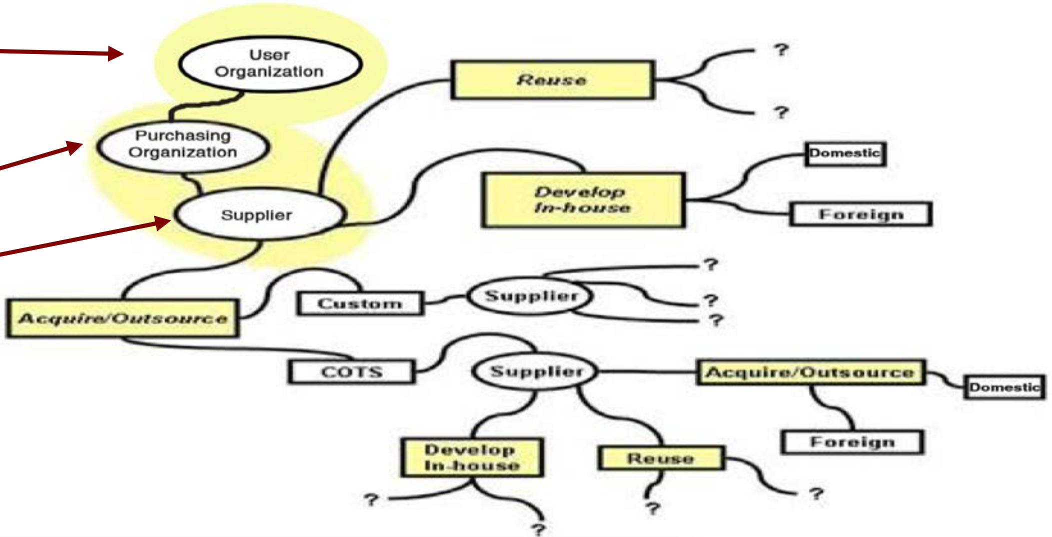
Software supply chain management

Enabling enterprise control of risks attributable to exploitable software

Software supply chain risk management

Mitigating third-party risks attributable to exploitable software

- Enterprise-level:
  - Regulatory compliance
  - Changing threat environment
  - Business case
- Program/project-level:
  - Cost
  - Schedule
  - Performance

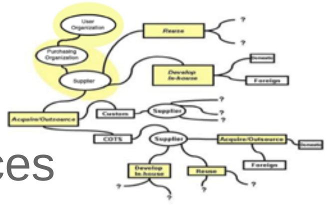


Who makes risk decisions?
Who determines 'fitness for use' criteria for technical acceptability?
Who "owns" residual risk from tainted products?

Note: "Tainted" products: corrupted with malware, or exploitable weaknesses and/or vulnerabilities

IoT supply chain risk management

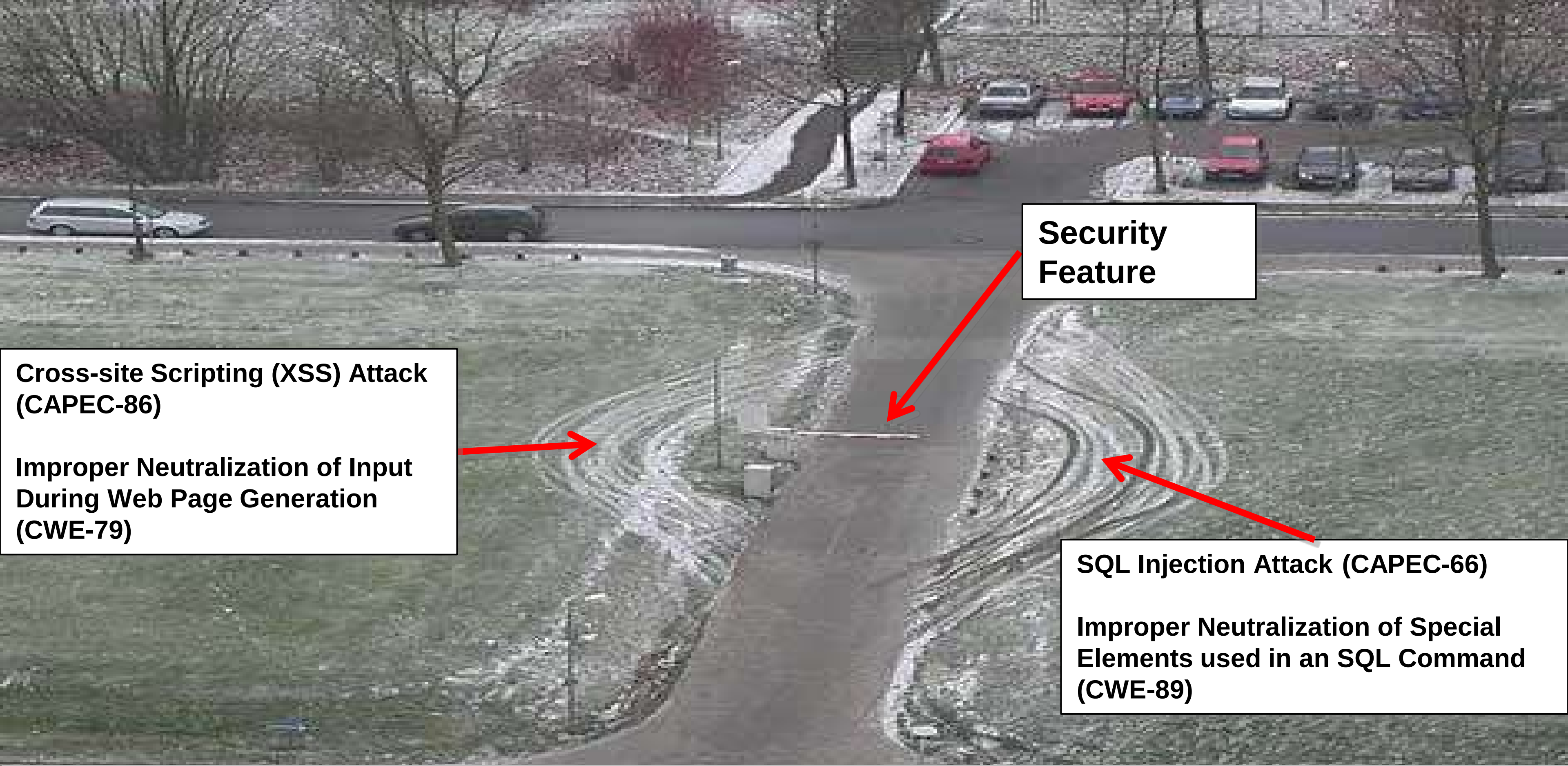
Mitigating 3rd-party risks attributable to exploitable software in IoT devices



Increased risk from supply chain due to:

Increasing dependence on globally sourced devices	• Varying levels of development/outsourcing controls • Lack of transparency in process chain of custody • Varying levels of acquisition 'due-diligence"
Residual risk	• Tainted products with malware, exploitable weaknesses (CWEs) and vulnerabilities (CVEs) • Defective and unauthentic/counterfeit products
Growing technological sophistication among adversaries	• Internet enables adversaries to probe, penetrate, & attack remotely • Supply chain attacks can exploit products and processes

Software in the supply chain is often the vector of attack


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

Software Testing

Enabling insight into risks attributable to exploitable software

IoT Supply Chain Risk Management:

Testing Software & Enabling Cybersecurity Assurance for Network-Connectable Devices

Software is buggy

How many exploitable weaknesses and vulnerabilities are in your systems and devices?

Input processing

Any software processing input can be attacked: network interfaces, device drivers, user interface, etc..

Hackers use binary analysis & fuzzing techniques to find vulnerabilities

These are used to exploit or launch attacks

These can also be discovered & mitigated by suppliers; should be used in test criteria for acceptance testing

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.

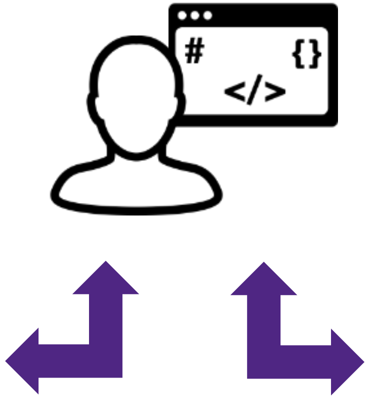
- Known Vulnerabilities (CVEs, ITU-T X.1520)
- If suppliers cannot mitigate known vulnerabilities prior to delivery and use, then what level of confidence can anyone have that patching and reconfiguring will be sufficient or timely to mitigate exploitation?

- 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 Today Is Assembled

Software Development

SW development process



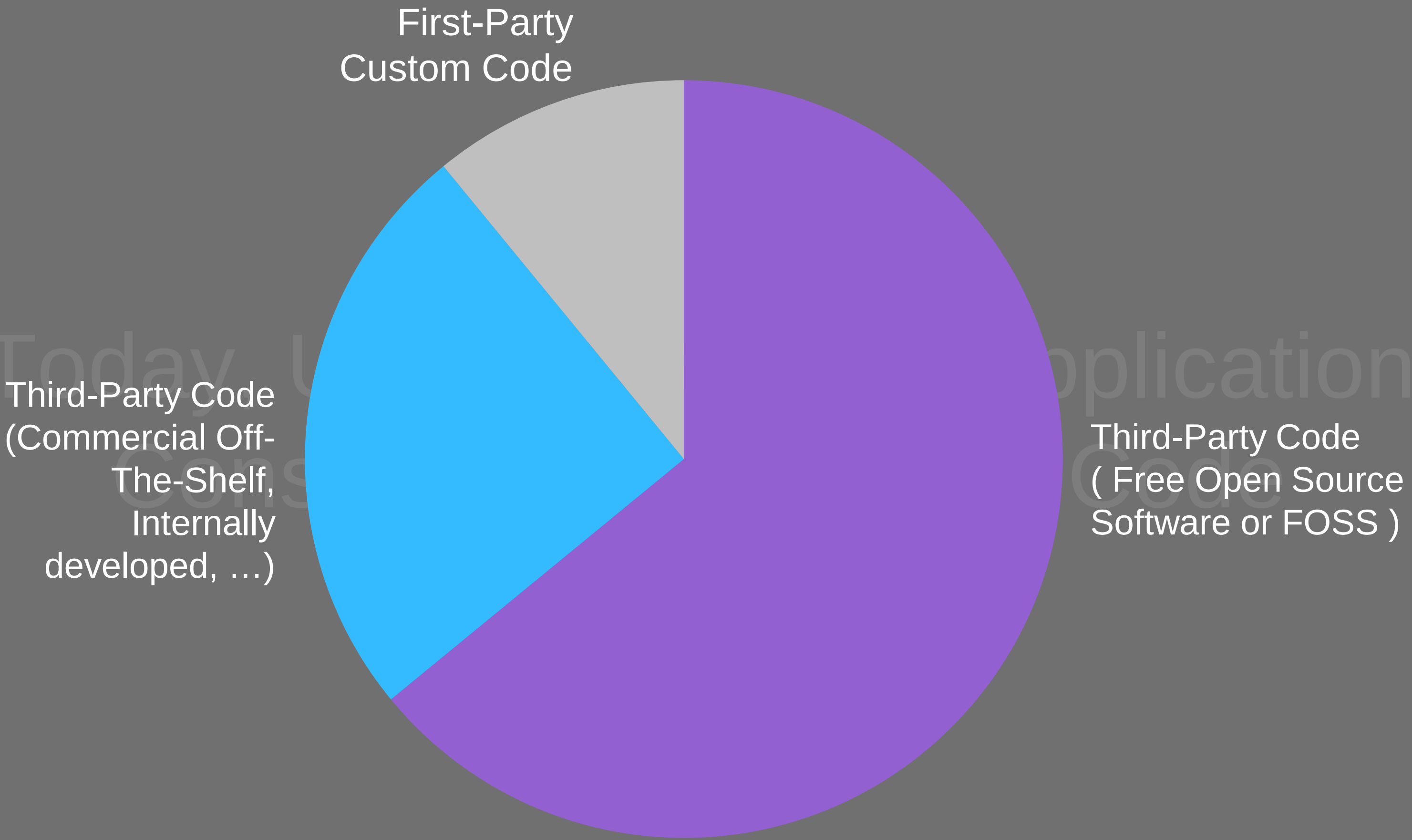
Part Original

Part Third Party

Supply Chain

SW components

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?

Types of Automated Tools/Testing

What They Find; How They Support 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** – Identifies license types and finds known vulnerabilities; 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.

Total Economic Impact of Software Testing Tools

Forrester Case Study – Useful Framework

September 2016

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

FORRESTER

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®

Complete support across the 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

Embedded

Cloud

Mobile

SYNOPSYS®

IoT supply chain risk management

Procurement requirements, independent testing and certification

IoT Software Supply Chain Risk Management:

Proactive Control with Procurement Language for Supply Chain Cyber Assurance

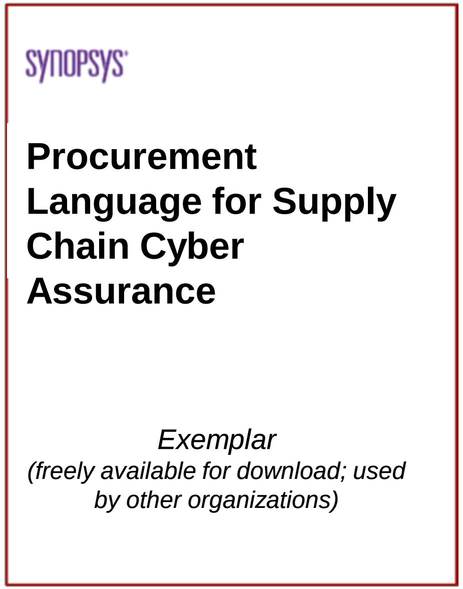
Product Development Specification and Policy

Security Program

System Protection and Access Control

Product Testing and Verification

Deployment and Maintenance



SYNOPSYS

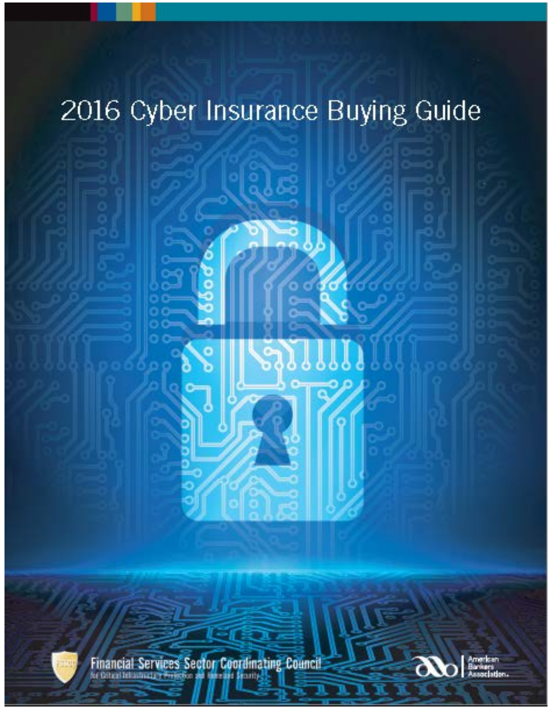
Procurement Language for Supply Chain Cyber Assurance

Exemplar
(freely available for download; used by other organizations)

https://www.synopsys.com/software-integrity/resources/white-papers/procurement-language-risk.html

SYNOPSYS

Supply Chain Cyber Assurance – Procurement Requirements


2016 Cyber Insurance Buying Guide
Financial Services Sector Coordinating Council
American Bankers Association.

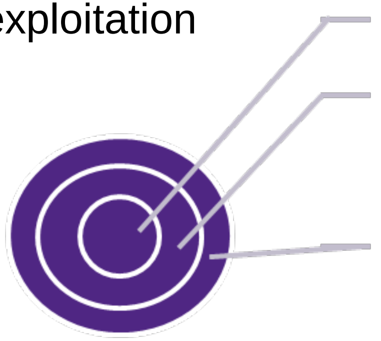
Source: Financial Services Sector Coordinating Council for Critical Infrastructure Protection and Homeland Security

- Product Development Specification and Policy
- Security Program
- System Protection and Access Control
- Product Testing and Verification
  - Communication Robustness Testing
  - Software Composition Analysis
  - Static Source Code Analysis
  - Dynamic Runtime Analysis
  - Known Malware Analysis
  - Bill of Materials
  - Validation of Security Measures
- Deployment and Maintenance

Software Supply Chain Risk Management:

Underwriters Labs Cybersecurity Assurance for Network-Connectable Devices

- UL Cybersecurity Assurance Program (**UL CAP**) provides independent testing and certification of network-connectable devices
- UL CAP uses Synopsys Software Integrity tool suite to comprehensively address software issues in systems and devices
- UL CAP is **Product Oriented & Industry Specific** with these goals:
  - Reduce software vulnerabilities
  - Reduce weaknesses, minimize exploitation
  - Address known malware



UL 2900-3: Organizational Processes

UL 2900-2-1, -2-2: Industry Specific Requirements (currently for ICS & healthcare systems & devices)

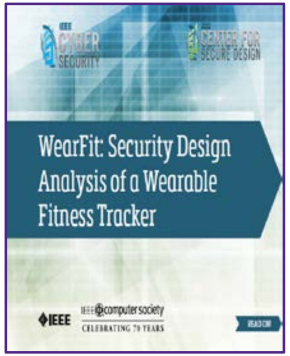
UL 2900-1: CAP General Requirements/

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.

Wearfit Security Design Analysis of a Wearable Fitness Tracker

- Learn how the [Top 10 Software Security Design Flaws](https://cybersecurity.ieee.org/center-for-secure-design) can be approached for wearable fitness tracking systems.
- Analysis based as much on real-world systems, providing a broad analysis of threats facing users of wearable fitness-tracking devices.



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

IoT Device Security: Upgradability and Patching

- DoC NTIA MultiStakeholder Process for IoT Security upgradability & patching
  - Are standards used for the design, build and support of IoT devices?
  - Are IoT devices capable of being patched and upgraded?
  - What are expected roles of users in patching devices?
  - What are the expected roles of manufactures of IoT devices associated with updates and patches?

- Understanding risk exposure attributable to IoT device

http://www.ntia.doc.gov/other-publication/2016/multistakeholder-process-iot-security

Evolving SDLC landscape impacts software integrity

Lack visibility into evolving application portfolio

Comprehensive view into risk

Accuracy and speed of quality defects and security vulnerability feedback

Focus

New tech stacks and attack surfaces

Embedded devices

Cloud (private, hybrid, public)

Languages, open source and frameworks

New development philosophies and approaches

Agile, DevOpsSec, CI/CD

Fit into toolchain eco-systems

Automation through toolchain integration

Changing testing demands

Align with workflow timeframes

Security as a core component of quality

Testing coverage and depth

Thank You

Joe Jarzombek – Global Manager, Software Supply Chain Solutions
Joe.Jarzombek@synopsys.com | 703.627.4644
Synopsys Software Integrity Group | www.synopsys.com/software

Join us in our online Software Integrity Community for software security and quality assurance
See State of Fuzzing 2017 to gain insight in software development where further testing remains
Synopsys named a Leader in AppSec Testing in Gartner’s 2017 Magic Quadrant