Software Solutions Symposium 2017

March 20–23, 2017

Risks in the Software Supply Chain

Mark Sherman

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Copyright 2017 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

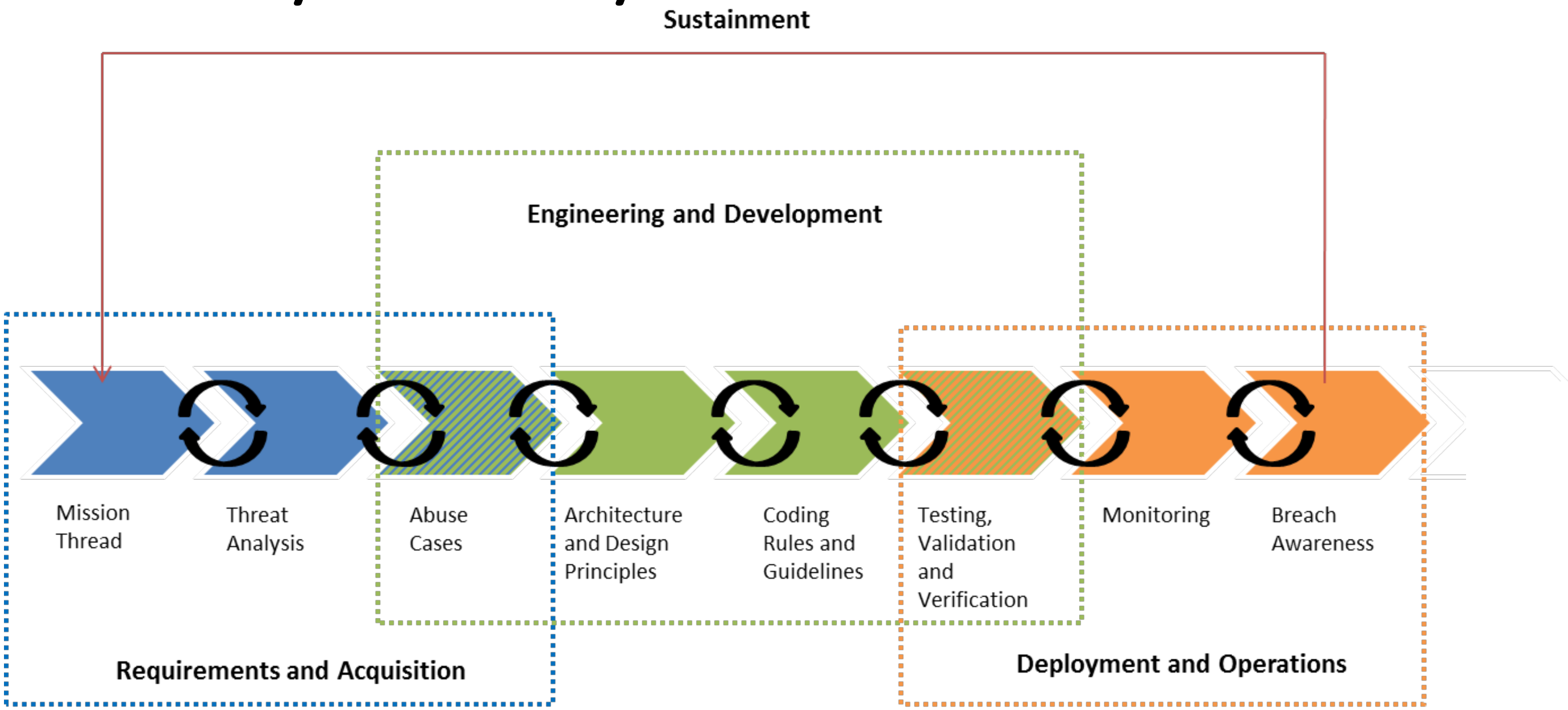
[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

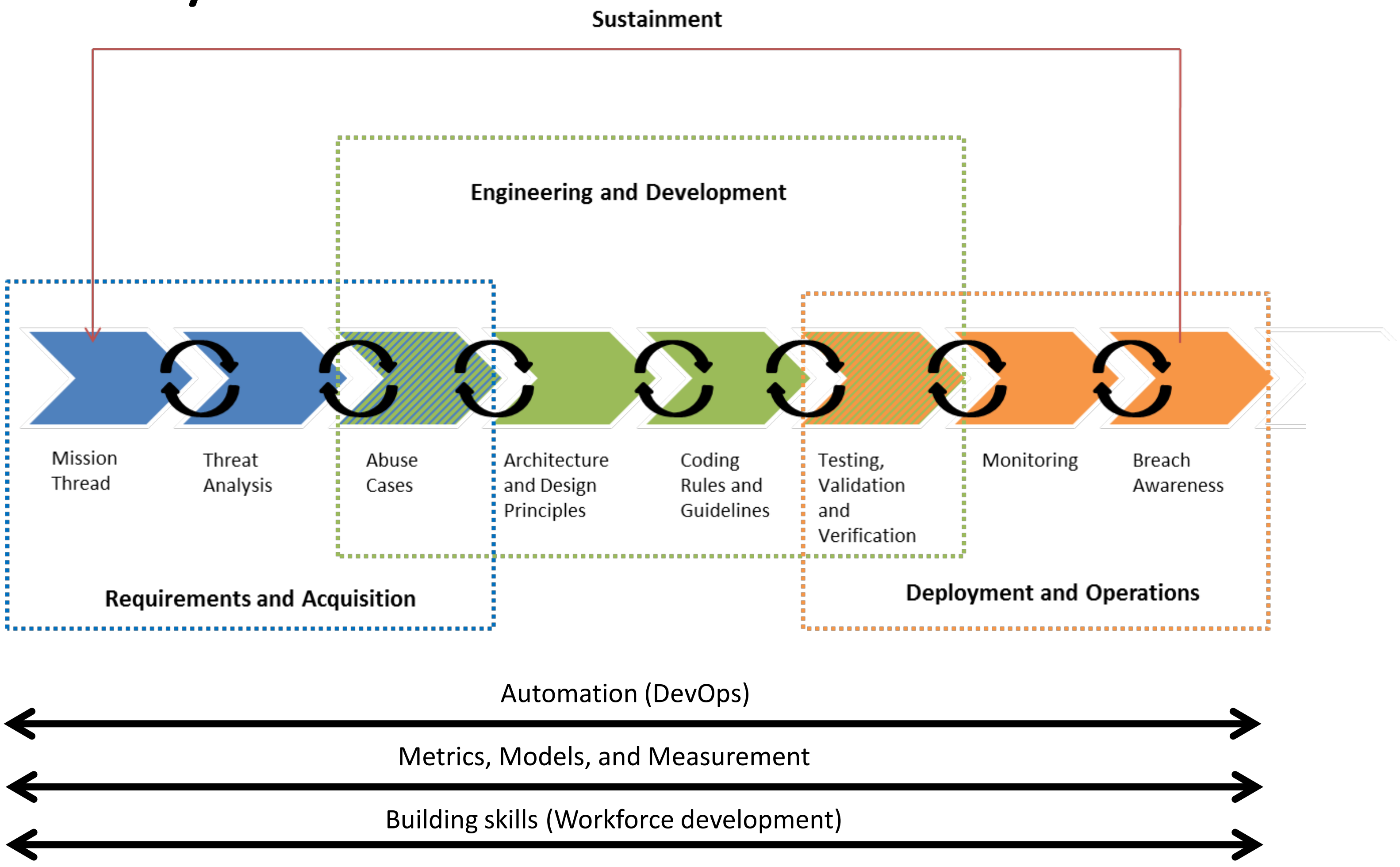
CERT® is a registered mark of Carnegie Mellon University.

DM-0004556

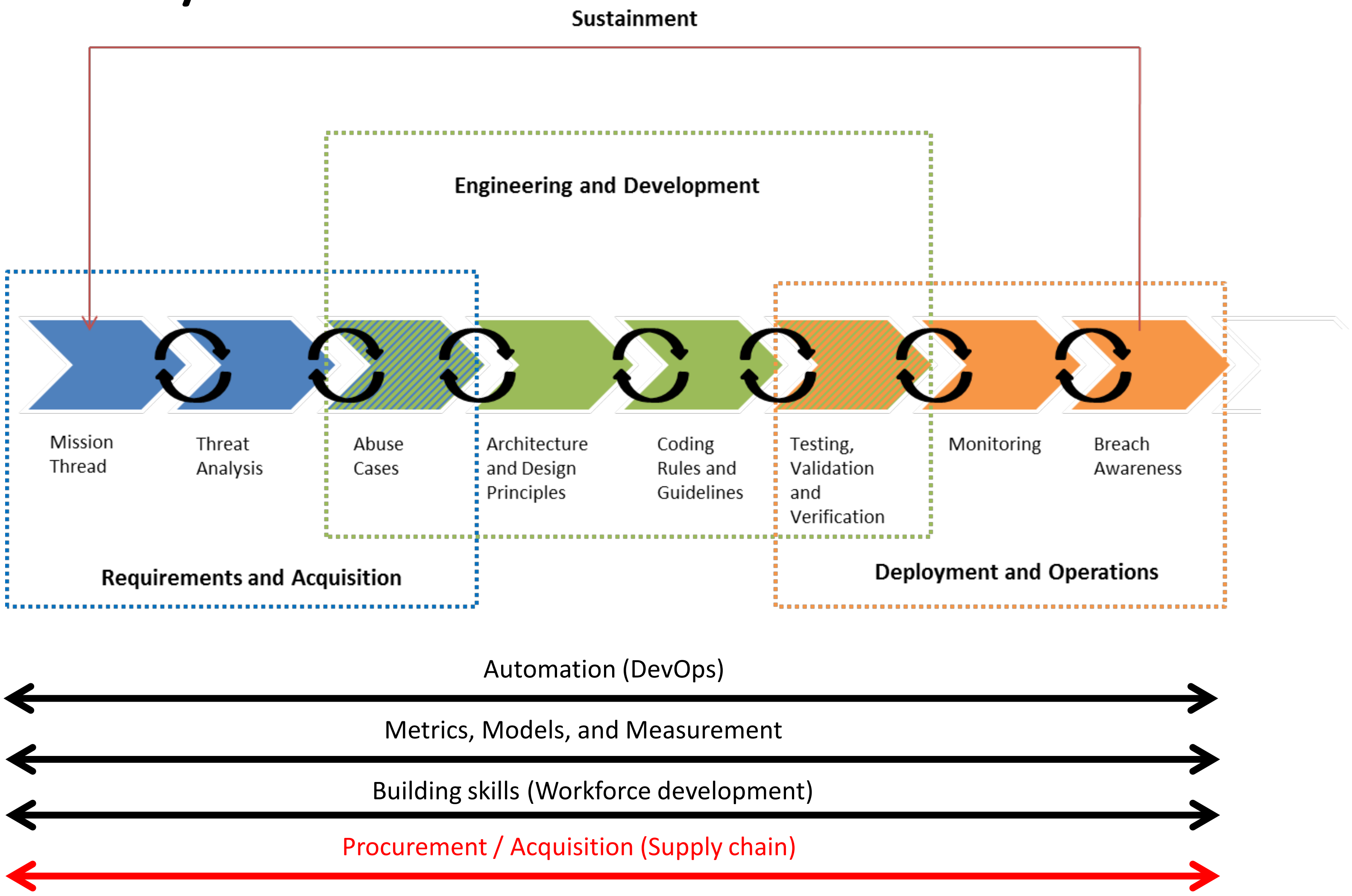
Cybersecurity is a lifecycle issue


Sustainment
Engineering and Development
Mission Thread
Threat Analysis
Abuse Cases
Architecture and Design Principles
Coding Rules and Guidelines
Testing, Validation and Verification
Monitoring
Breach Awareness
Requirements and Acquisition
Deployment and Operations

Cross lifecycle issues


Sustainment
Engineering and Development
Mission Thread
Threat Analysis
Abuse Cases
Architecture and Design Principles
Coding Rules and Guidelines
Testing, Validation and Verification
Monitoring
Breach Awareness
Requirements and Acquisition
Deployment and Operations
Automation (DevOps)
Metrics, Models, and Measurement
Building skills (Workforce development)

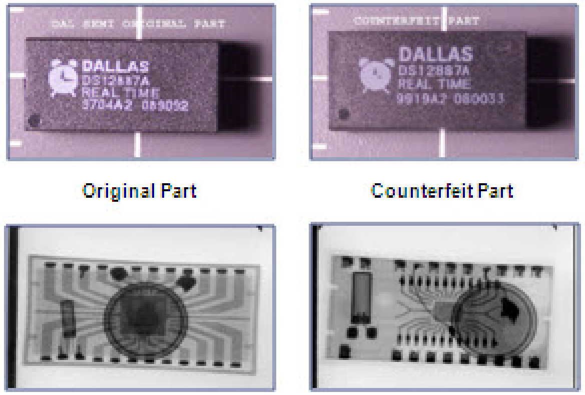
Cross lifecycle issues

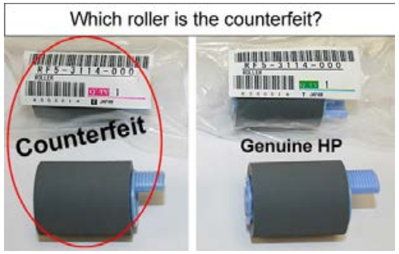

Sustainment
Engineering and Development
Mission Thread
Threat Analysis
Abuse Cases
Architecture and Design Principles
Coding Rules and Guidelines
Testing, Validation and Verification
Monitoring
Breach Awareness
Requirements and Acquisition
Deployment and Operations
Automation (DevOps)
Metrics, Models, and Measurement
Building skills (Workforce development)
Procurement / Acquisition (Supply chain)

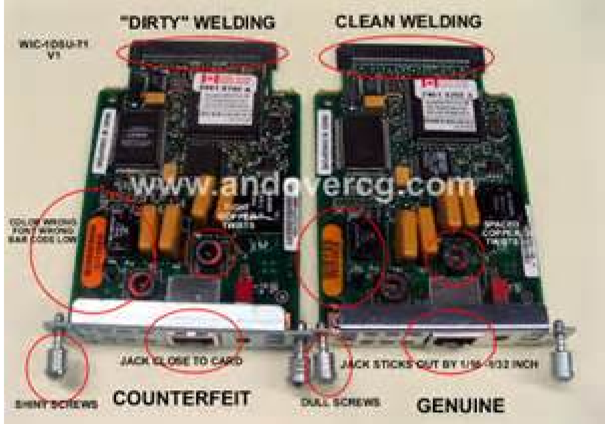
© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Conventional view of supply chain risk


DAL SEMI ORIGINAL PART
DALLAS DS12887A REAL TIME 9704A2 089092
COUNTERFEIT PART
DALLAS DS12887A REAL TIME 9919A2 080033
Original Part
Counterfeit Part





Sources: http://www.nytix.com/NewYorkCity/articles/handbags.html; http://www.laserwisetech.co.nz/secret.php; http://www.muscatdaily.com/Archive/Oman/Fake-car-parts-contribute-to-rise-in-road-accidents-Experts; http://www.andovercg.com/services/cisco-counterfeit-wic-1dsu-t1.shtml; http://unites-systems.com/l.php?id=191

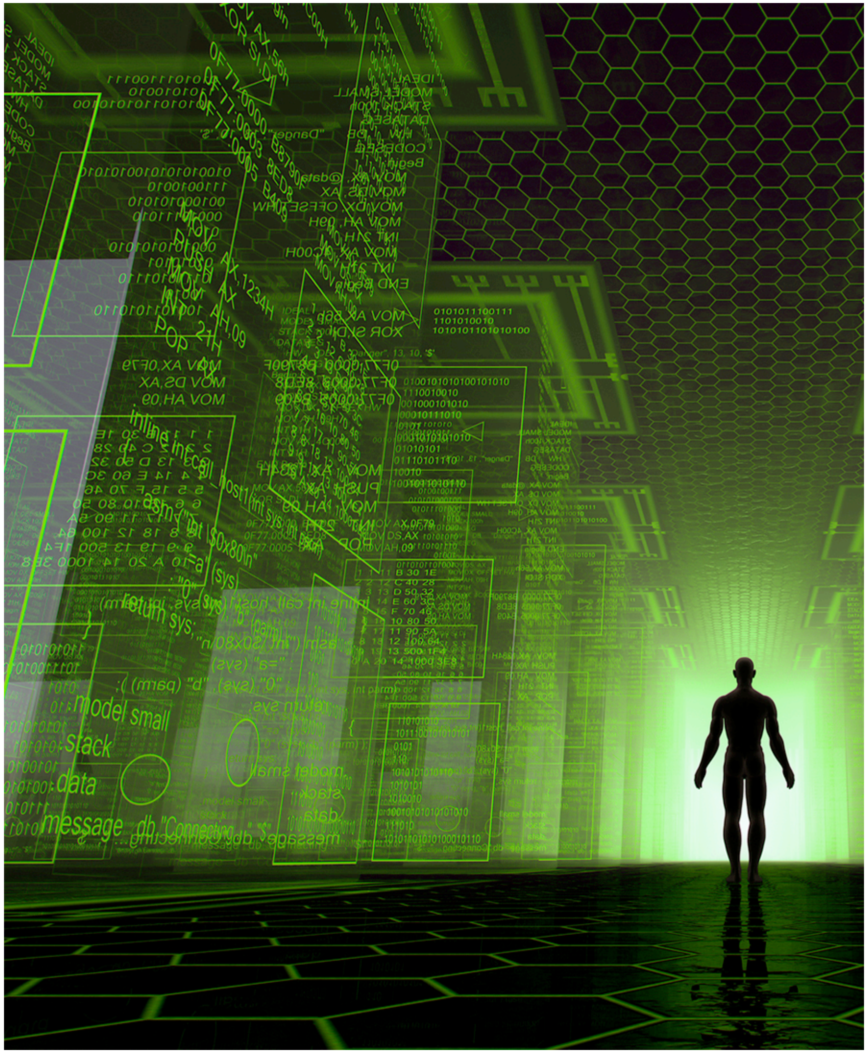
Supply chains also maintain product properties

Cold Chain

A cold chain is a temperature-controlled supply chain. An unbroken cold chain is an uninterrupted series of storage and distribution activities which maintain a given temperature range.

Source: Wikipedia, https://en.wikipedia.org/wiki/Cold_chain

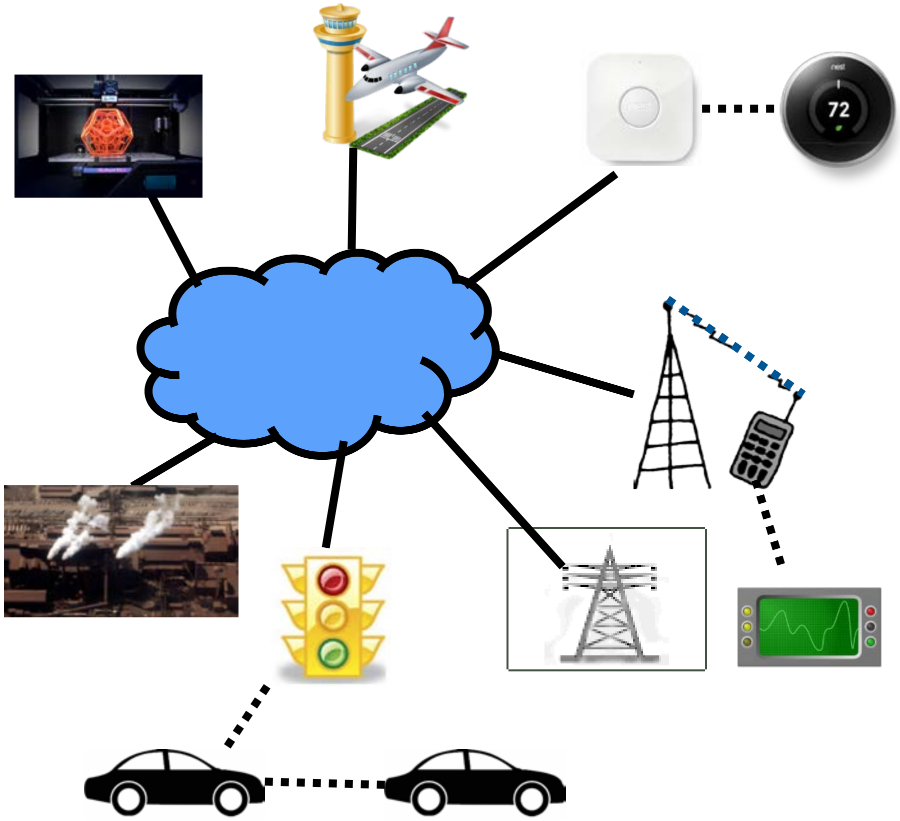
Software is the new hardware – IT



IT moving from specialized hardware to software, virtualized as

- Servers: virtual CPUs
- Storage: SANs
- Switches: Soft switches
- Networks: Software defined networks
- Communications: Software defined radios

Software is the new hardware – cyber physical



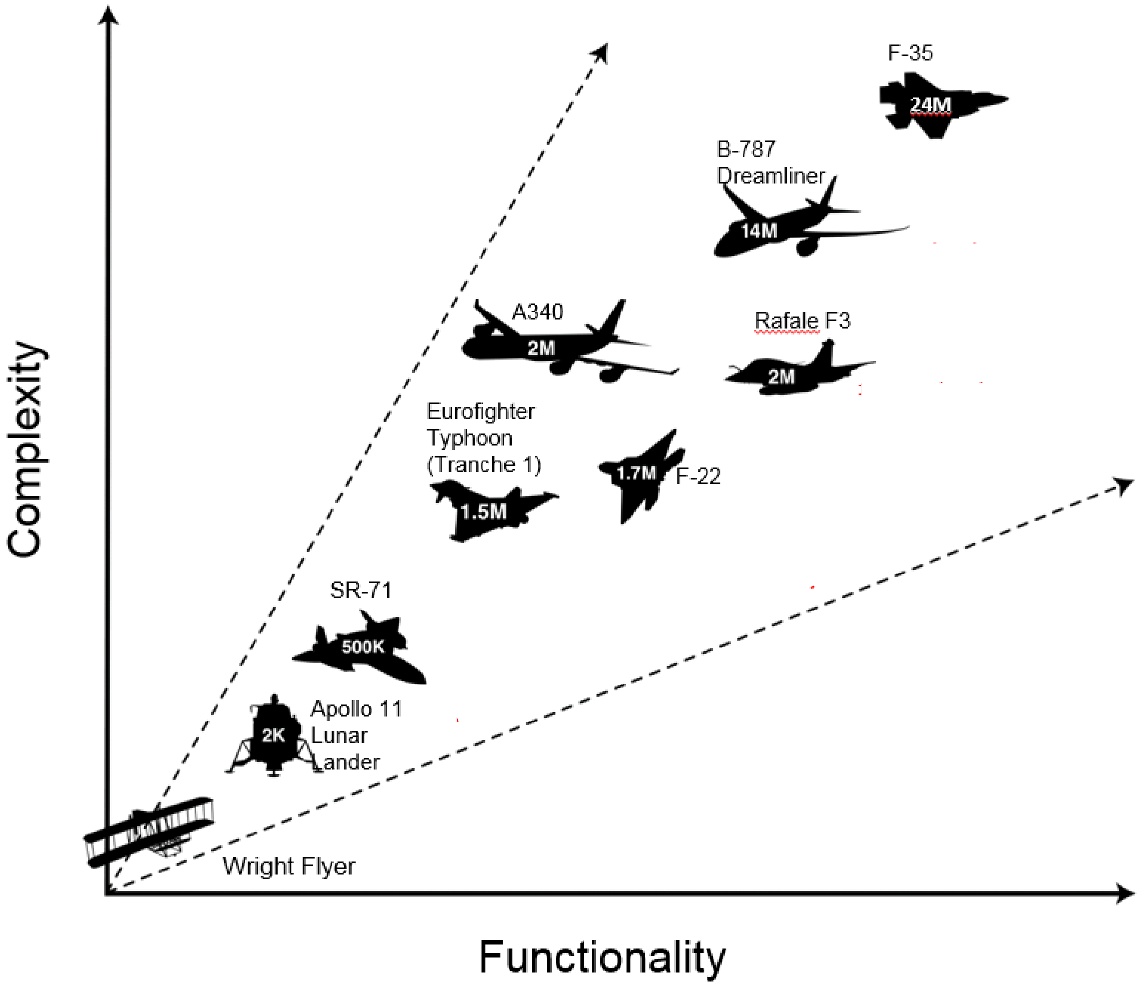
- Cellular
  - Main processor
  - Graphics processor
  - Base band processor (SDR)
  - Secure element (SIM)
- Automotive
  - Autonomous vehicles
  - Vehicle to infrastructure (V2I)
  - Vehicle to vehicle (V2V)
- Industrial and home automation
  - 3D printing (additive manufacturing)
  - Autonomous robots
  - Interconnected SCADA
- Aviation
  - Next Gen air traffic control
  - Fly by wire
- Smart grid
  - Smart electric meters
  - Smart metering infrastructure
- Embedded medical devices

……

© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

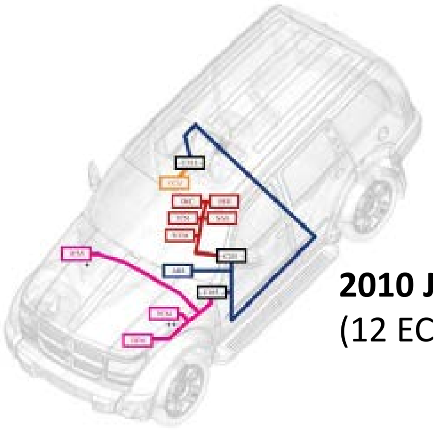
Mission function is increasingly delivered in software



“The [F-35] aircraft relies on more than 20 million lines of code to "fuze" information from the JSF's radar, infrared cameras, jamming gear, and even other planes and ground stations to help it hunt down and hide from opponents, as well as break through enemy lines to blow up targets on the ground. …. But if the computer doesn't work, the F-35's greatest advertised advantages over existing rivals and future threats would suddenly become moot.”
The Week, 2016

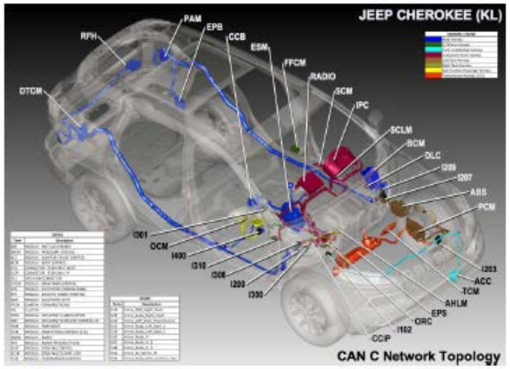
Source: Joseph Trevithick, http://theweek.com/articles/605165/f35-still-horribly-broken. Feb 26, 2016

Vehicle technology following the same path



2010 Jeep Cherokee
(12 ECUs)

2014 Jeep Cherokee
(32 ECUs)



CAN-C Network – 2014 Jeep Cherokee

Common assertion that modern high end vehicles have

- Over 100M lines of code
- Over 50 antennas
- Over 100 ECUs

Sources: Miller and Valasek, A Survey of Remote Automotive Attack Surfaces, http://illmatics.com/remote%20attack%20surfaces.pdf;
https://www.cst.com/webinar14-10-23~?utm_source=rfg&utm_medium=web&utm_content=mobile&utm_campaign=2014series
https://en.wikipedia.org/wiki/Electronic_control_unit

Software is the new hardware – everything

90 percent of [Samsung’s] products -- which includes everything from smartphones to refrigerator-- would be able to connect to the Web by 2017. In five years, every product in the company's entire catalog would be Internet connected.

B.K. Yoon, Samsung co-CEO

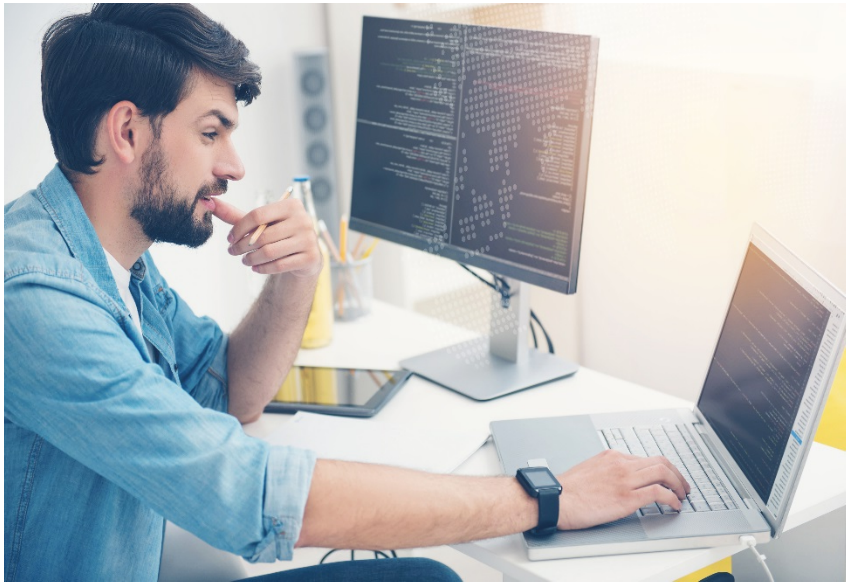
CNET

Jan 5, 2015

Software is eating the world.
Marc Andreessen, WSJ, Aug 20,2011

Source: http://www.cnet.com/news/samsung-co-ceo-in-5-years-all-our-products-will-be-internet-connected/
http://www.wsj.com/articles/SB10001424053111903480904576512250915629460

Evolution of software development – 1960s



Custom development – context:

- Software was limited
  - Size
  - Function
  - Audience
- Each organization employed developers
- Each organization created their own software

Supply chain: practically none

© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

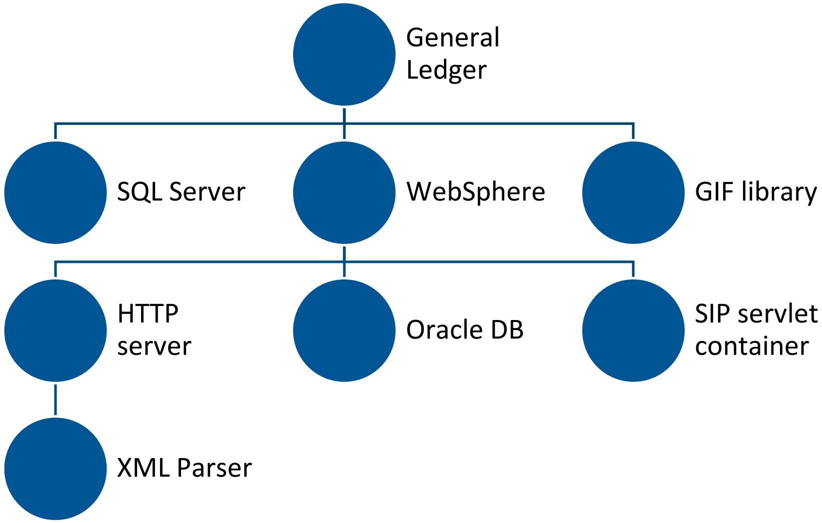
Evolution of software development – 1970s

Shared development – ISVs (COTS) – context:

- Function largely understood
  - Automating existing processes
- Grown beyond ability for using organization to develop economically
- Outside of core competitiveness by acquirers

Supply chain: software supplier

Evolution of software development – 1990s



Development is now assembly using collective development

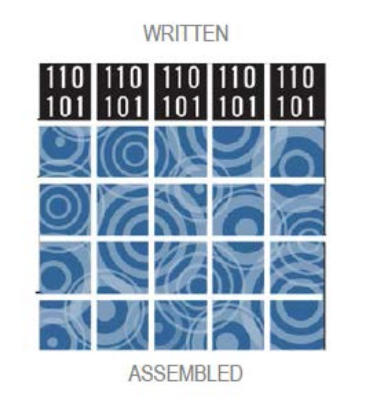
- Too large for single organization
- Too much specialization
- Too little value in individual components

Supply chain: long

Note: hypothetical application composition

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Evolution of software development – the rise of open source



- 90% of modern applications are assembled from 3rd party components
- Most applications are now assembled from hundreds of open source components, often reflecting as much as 90% of an application
- At least 75% of organizations rely on open source as the foundation of their applications

Distributed development – context:

- Amortize expense
- Outsource non-differential features
- Lower acquisition (CapEx) expense

Supply chain: opaque

Sources: Geer and Corman, “Almost Too Big To Fail,” ;login: (Usenix), Aug 2014; Sonatype, 2014 open source development and application security survey

© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Evolution of software development – the rise of open source

- 90% of
  assem
  - At le
    as th
- Most applications are now assembled from hundreds of open source components, often reflecting as much as 90% of an application

Distributed development – context:

ntial

Ex)

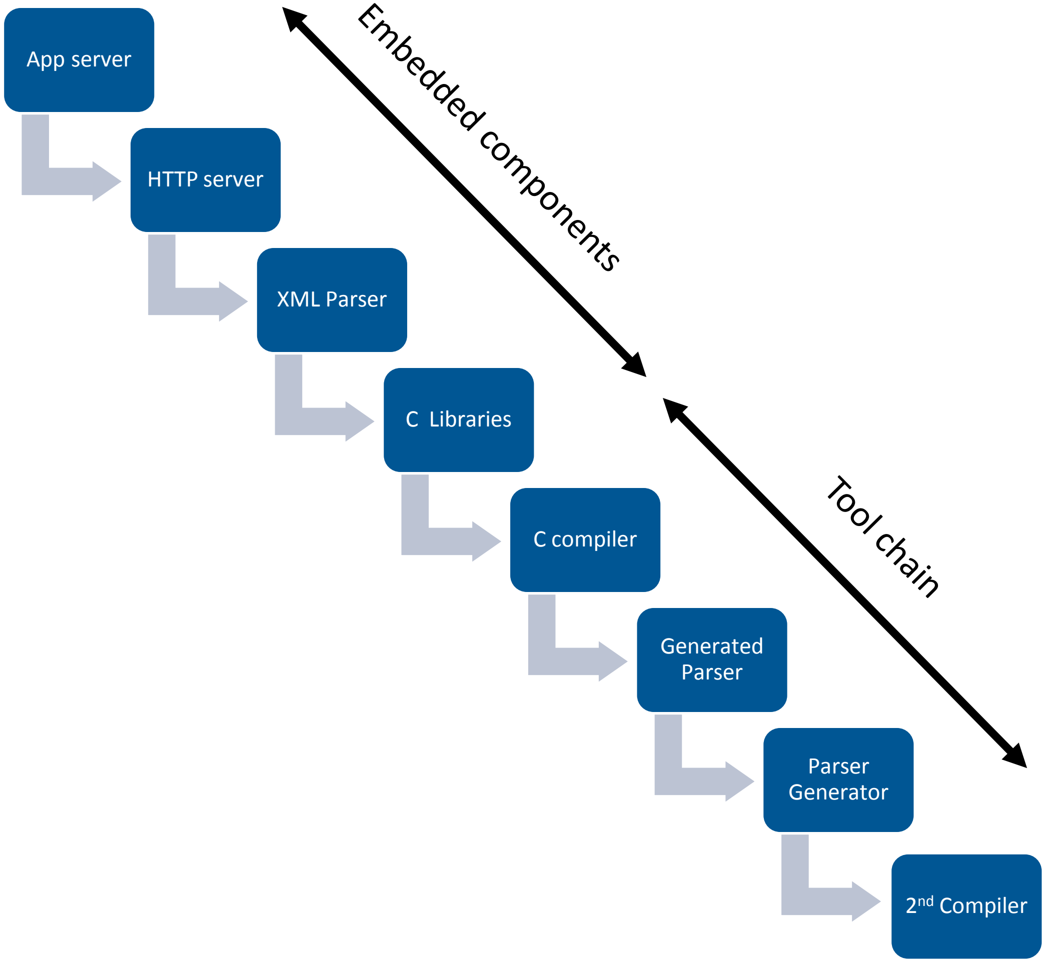
“Developers are gorging themselves on an ever expanding supply of open source components”

Sonatype, “2016 State of the Software Supply Chain”

Supply chain: opaque

Sources: Geer and Corman, “Almost Too Big To Fail,” ;login: (Usenix), Aug 2014; Sonatype, 2014 open source development and application security survey

Open source supply chain has a long path



App server

HTTP server

XML Parser

C Libraries

C compiler

Generated Parser

Parser Generator

2nd Compiler

Embedded components

Tool chain

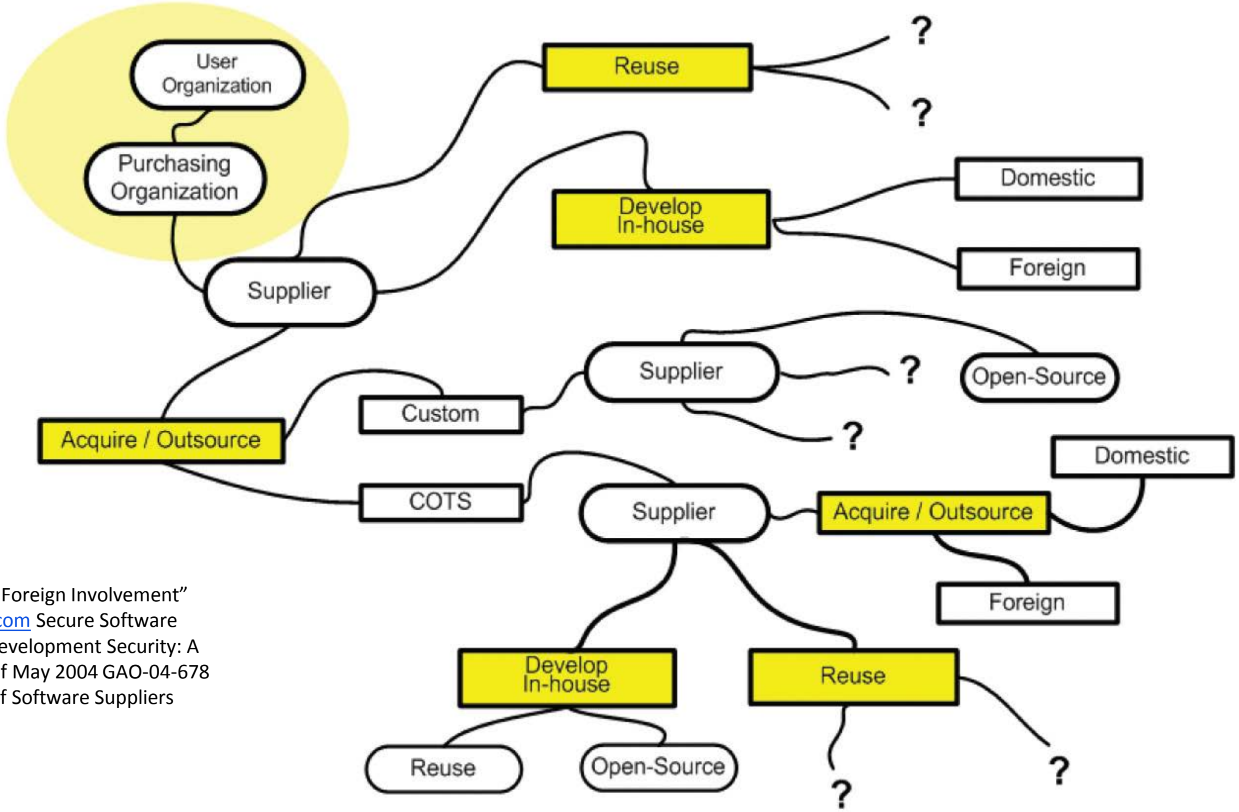
© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Today: Software supply chain for assembled software

Expanding the scope and complexity of acquisition and deployment

Visibility and direct controls are limited (only in shaded area)

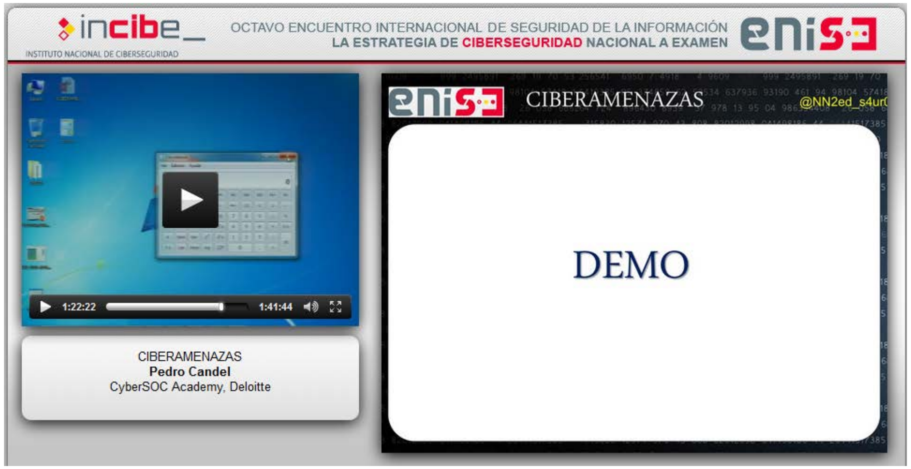


Source: “Scope of Supplier Expansion and Foreign Involvement” graphic in DACS www.softwaretechnews.com Secure Software Engineering, July 2005 article “Software Development Security: A Risk Management Perspective” synopsis of May 2004 GAO-04-678 report “Defense Acquisition: Knowledge of Software Suppliers Needed to Manage Risks”

© 2017 Carnegie Mellon University
[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Corruption along the supply chain is easy

Unexpected or unintended behaviors in components



OCTAVO ENCUENTRO INTERNACIONAL DE SEGURIDAD DE LA INFORMACIÓN
LA ESTRATEGIA DE CIBERSEGURIDAD NACIONAL A EXAMEN

CIBERAMENAZAS

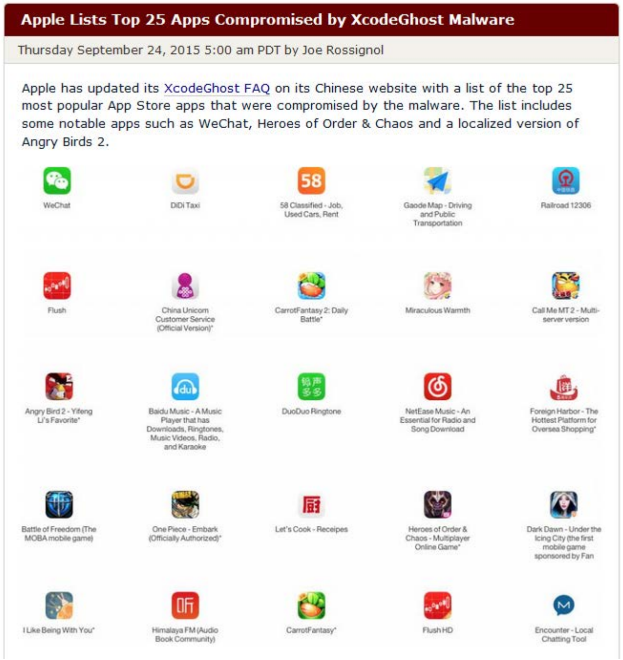
DEMO

CIBERAMENAZAS
Pedro Candel
CyberSOC Academy, Deloitte

Knowledgeable analysts can convert packaged binary into malware in minutes

Sources: Pedro Candel, Deloitte CyberSOC Academy , Deloitte
http://www.8enise.webcastlive.es/webcast.htm?video=08; http://www.microsoft.com/Products/Games/FSInsider/freeflight/PublishingImages/scene.jpg;
https://www.withfriendship.com/user/mithunss/easter-eggs-in-microsoft-products.php

Corruption in the tool chain already exists



Apple Lists Top 25 Apps Compromised by XcodeGhost Malware

Thursday September 24, 2015 5:00 am PDT by Joe Rossignol

Apple has updated its XcodeGhost FAQ on its Chinese website with a list of the top 25 most popular App Store apps that were compromised by the malware. The list includes some notable apps such as WeChat, Heroes of Order & Chaos and a localized version of Angry Birds 2.

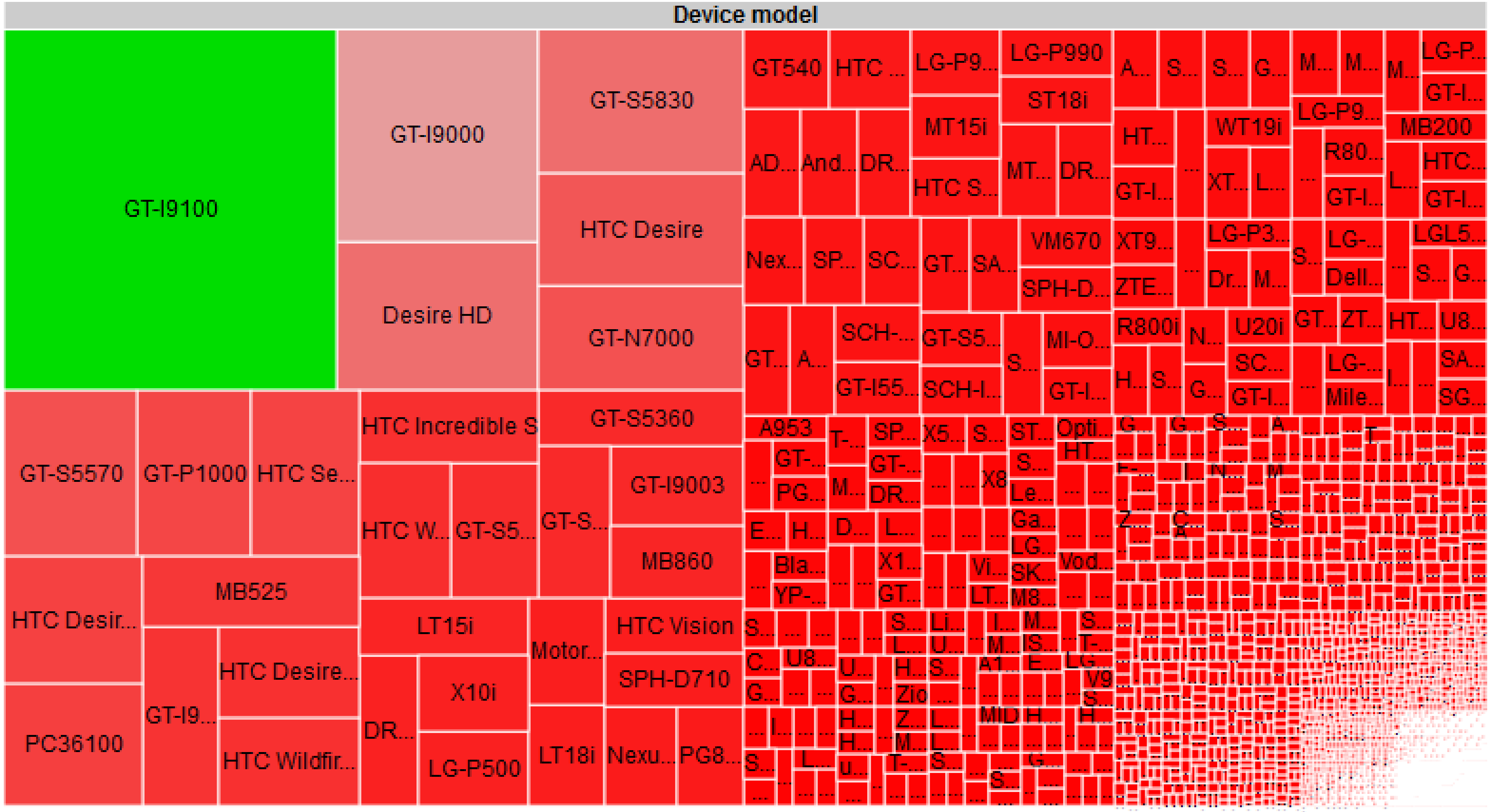
WeChat	DiDi Taxi	58 Classified - Job, Used Cars, Rent	Gaode Map - Driving and Public Transportation	Railroad 12306
Flush	China Unicom Customer Service (Official Version)*	CarrotFantasy 2: Daily Battle*	Miraculous Warmth	Call Me MT 2 - Multi-server version
Angry Bird 2 - Yifeng Li's Favorite*	Baidu Music - A Music Player that has Downloads, Ringtones, Music Videos, Radio, and Karaoke	DuoDuo Ringtone	NetEase Music - An Essential for Radio and Song Download	Foreign Harbor - The Hottest Platform for Oversea Shopping*
Battle of Freedom (The MOBA mobile game)	One Piece - Embark (Officially Authorized)*	Let's Cook - Receipes	Heroes of Order & Chaos - Multiplayer Online Game*	Dark Dawn - Under the Icing City (the first mobile game sponsored by Fan
I Like Being With You*	Himalaya FM (Audio Book Community)	CarrotFantasy*	Flush HD	Encounter - Local Chatting Tool

- XcodeGhost corrupted Apple's development environment

- Major programs affected
  - WeChat
  - Badu Music
  - Angry Birds 2
  - Heroes of Order & Chaos
  - iOBD2

Sources: http://www.macrumors.com/2015/09/24/xcodeghost-top-25-apps-apple-list/
http://www.itntoday.com/2015/09/the-85-ios-apps-affected-by-xcodeghost.html

Versions of Android illustrate open source fragmentation



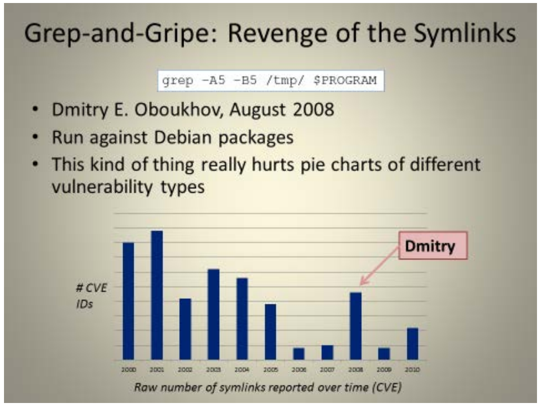
Source: http://opensignal.com/reports/fragmentation.php

.

Open source is not secure

Heartbleed and Shellshock were found by exploitation

Other open source software illustrates vulnerabilities from cursory inspection

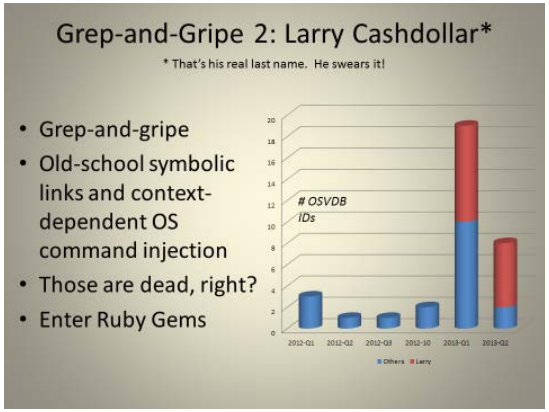


Grep-and-Gripe: Revenge of the Symlinks

```
grep -A5 -B5 /tmp/ $PROGRAM
```

- Dmitry E. Oboukhov, August 2008
- Run against Debian packages
- This kind of thing really hurts pie charts of different vulnerability types

Raw number of symlinks reported over time (CVE)



Grep-and-Gripe 2: Larry Cashdollar*

* That's his real last name. He swears it!

- Grep-and-gripe
- Old-school symbolic links and context-dependent OS command injection
- Those are dead, right?
- Enter Ruby Gems

Sources: Steve Christey (MITRE) & Brian Martin (OSF), Buying Into the Bias: Why Vulnerability Statistics Suck, https://media.blackhat.com/us-13/US-13-Martin-Buying-Into-The-Bias-Why-Vulnerability-Statistics-Suck-Slides.pdf; Sonatype, Sonatype Open Source Development and Application Security Survey; Sonatype, 2016 State of the Software Supply Chain; Aspect Software "The Unfortunate Reality of Insecure Libraries," March 2012

Open source is not secure

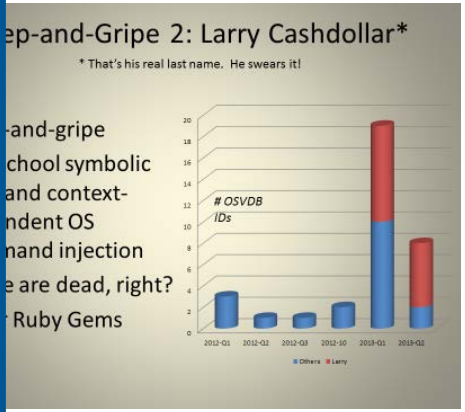
Heartbleed and Shellshock were fou by exploitation

Other open source software illustrates vulnerabilities from cu inspection

1.8 billion vulnerable open source components downloaded in 2015

26% of the most common open source components have high risk vulnerabilities

On average, applications have 22.5 open source vulnerabilities



ep-and-Gripe 2: Larry Cashdollar*

* That's his real last name. He swears it!

-and-gripe
chool symbolic
and context-
ndent OS
nand injection
e are dead, right?
Ruby Gems

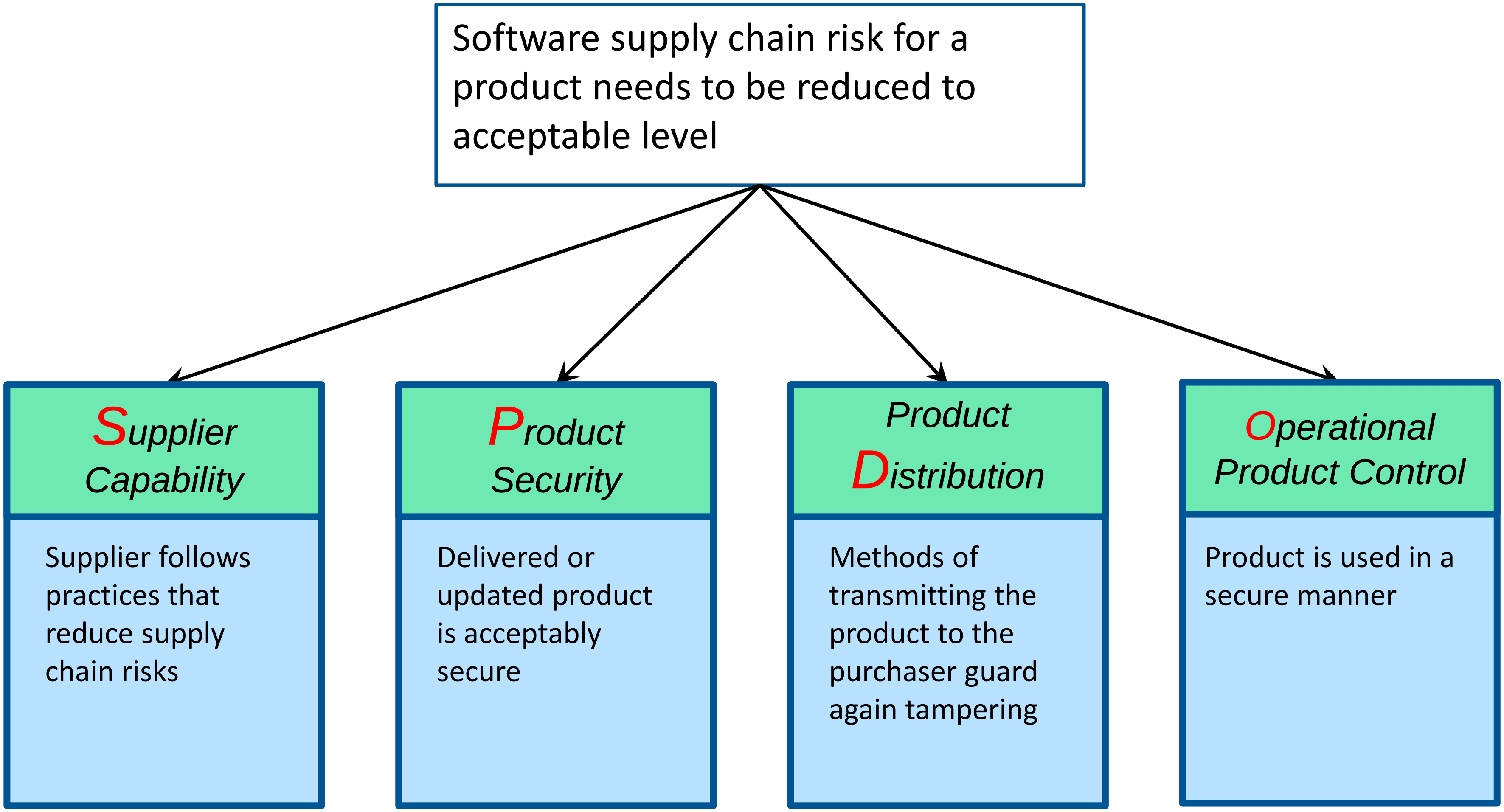
# OSVDB
IDs
0
2
4
6
8
10
12
14
16
18
20
2012-Q1
2012-Q2
2012-Q3
2012-10
2013-Q1
2013-Q2
Others
Larry

s Suck, https://media.blackhat.com/us-13/US-13-
ource Development and Application Security Survey;
ure Libraries," March 2012, Mike Pittenger, Black

© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Reducing software supply chain risk factors



Software supply chain risk for a product needs to be reduced to acceptable level

Supplier Capability

Supplier follows practices that reduce supply chain risks

Product Security

Delivered or updated product is acceptably secure

Product Distribution

Methods of transmitting the product to the purchaser guard again tampering

Operational Product Control

Product is used in a secure manner

Supplier security commitment evidence

Supplier employees are educated as to security engineering practices

- Documentation for each engineer of training and when trained/retrained
- Revision dates for training materials
- Lists of acceptable credentials for instructors
- Names of instructors and their credentials

Supplier follows suitable security design and development practices

- Documented design guidelines
- Has analyzed attack patterns appropriate to the design such as those that are included in Common Attack Pattern Enumeration and Classification (CAPEC)
- Protection against insider (developer) threat

© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Evaluate a product’s threat resistance

What product characteristics minimize opportunities to enter and change the product’s security characteristics?

- Attack surface evaluation: Exploitable features have been identified and eliminated where possible
  - Access controls
  - Input/output channels
  - Attack enabling applications – email, Web
- Design and coding weaknesses associated with exploitable features have been identified and mitigated (CWE)
- Independent validation and verification of threat resistance
- Dynamic, Static, Interactive Application Security Testing (DAST, SAST, IAST)
- Delivery in or compatibility with Runtime Application Self Protection (RASP) containers

Establishing good product distribution practices

Recognize that supply chain risks are accumulated

- Establish provenance procedures
  - Subcontractor/COTS-product supply chain risk is inherited by those that use that software, tool, system, etc.

Apply to the acquiring organizations and their suppliers

- Require good security practices by their suppliers
- Assess the security of delivered products
- Address the additional risks associated with using the product in their context

Minimize internal suppliers

- Single point of distribution to development community

Ideally open source is built with a compiler you trust

Maintain operational attack resistance

Who assumes responsibility for preserving product attack resistance with product deployment?

- Maintaining inventory of components
- Patching and version upgrades (component lifecycle management)
- Expanded distribution of usage
- Expanded integration

Usage changes the attack surface and potential attacks for the product

- Change in feature usage or risks
- Are supplier risk mitigations adequate for desired usage?
- Effects of vendor upgrades/patches and local configuration changes
- Effects of integration into operations (system of systems)

© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

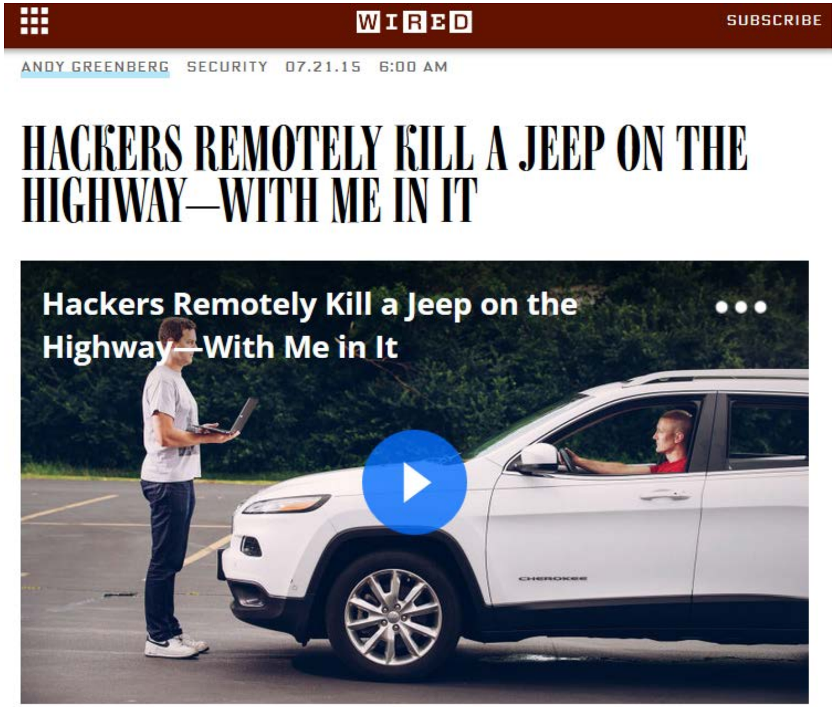
Steel furnaces have been successfully attacked

“Steelworks compromise causes massive damage to furnace.

One of the most concerning was a targeted APT attack on a German steelworks which ended **in the attackers gaining access to the business systems and through them to the production network** (including SCADA). The effect was that the attackers gained control of a steel furnace and this caused massive damages to the plant.”

Source: Sources: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Lageberichte/Lagebericht2014.pdf?__blob=publicationFile; http://www.resilienceoutcomes.com/state-ict-security/

Connecting automotive systems to internet opens system to attack



ANDY GREENBERG SECURITY 07.21.15 6:00 AM

HACKERS REMOTELY KILL A JEEP ON THE HIGHWAY—WITH ME IN IT

Hackers Remotely Kill a Jeep on the Highway—With Me in It

Extending systems opens vulnerabilities not anticipated

- Optimizations performed assuming one attack method
- Assumptions no longer hold with additional integrations

Studies suggest that new operational environments are a leading cause for introducing new vulnerabilities in existing systems.

Sources: http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/
Clark, Frei, Blaze, Smith, "Familiarity Breeds Contempt: The Honeymoon Effect and the Role of Legacy Code in Zero-Day Vulnerabilities," ACSAC '10 Dec. 6-10, 2010, p. 251-260."

What about open source?

Establish a supplier for open source components

Establish a process for tracking open source vulnerabilities

Restrict open source components that can be used

Establish an internal open source component distribution process

Maintain a registry of where open source components are used

Institute an update policy to remediate discovered and patched vulnerabilities

Source: http://opensource.org/

Business decisions are about risk

There are many risks to a business process or mission thread

- Within a system
- Collection of systems

Supply chain is one of many risk components

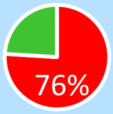
Evaluate software supply chain risk in the larger context of

- Supply chain risk
- System risk
- System of systems risk

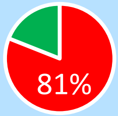
SERA: Security Engineering Risk Analysis

Where to start

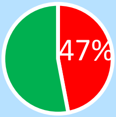
Anywhere



No meaningful controls over what components are applications



No coordination of security practices in various stages of the development life cycle



No acceptance tests for third-party code

Plenty of models to choose from

BSIMM: Building Security in Maturity Model

CMMI: Capability Maturity Model Integration for Acquisitions

PRM: SwA Forum Processes and Practices Group Process Reference Model

RMM: CERT Resilience Management Model

SAF: Software Assurance Framework

SAMM: OWASP Open Software Assurance Maturity Model

O-TTPS: Open Group Open Trusted Technology Provider™ Standard, Version 1.1

Sources: Sonatype, 2014 Sonatype Open Source Development and Application Security Survey;
Forrester Consulting, "State of Application Security," January 2011

© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Further reading

Alberts, Christopher, et al., “Introduction to the Security Engineering Risk Analysis (SERA) Fraemwork,” Software Engineering Institute, Nov 2014, http://resources.sei.cmu.edu/asset_files/TechnicalNote/2014_004_001_427329.pdf

Axelrod, C. Warren, “Mitigating Software Supply Chain Risk,” ISCA Journal Online, Vol 4., 2013, http://www.isaca.org/Journal/Past-Issues/2013/Volume-4/Pages/JOnline-Mitigating-Software-Supply-Chain-Risk.aspx

Axelrod, C. Warren, “Malware, Weakware and the Security of Software Supply Chains,” Cross-Talk, March/April 2014, p. 20, http://www.crosstalkonline.org/storage/issue-archives/2014/201403/201403-Axelrod.pdf

Ellison, Robert, et al, “Software Supply Chain Risk Management: From Products to Systems of Systems,” Software Engineering Institute, Dec 2010, https://resources.sei.cmu.edu/asset_files/technicalnote/2010_004_001_15194.pdf

Ellison, Robert, et al. “Evaluating and Mitigating Software Supply Chain Security Risks,” Software Engineering Institute, May 2010, http://resources.sei.cmu.edu/asset_files/technicalnote/2010_004_001_15176.pdf

Ellison, Robert and Woody, Carol, “Supply-Chain Risk Management: Incorporating Security into Software Development,” Proceedings of the 43rd Hawaii International Conference on System Sciences, 2010, http://resources.sei.cmu.edu/asset_files/WhitePaper/2013_019_001_297341.pdf

Jarzombek, Joe, “Collaboratively Advancing Strategies to Mitigate Software Supply Chain Risks,” July 30, 2009, http://csrc.nist.gov/groups/SMA/ispab/documents/minutes/2009-07/ispab_july09-jarzombek_swa-supply-chain.pdf

Software Assurance Forum, Processes and Practices Working Group, “Software Assurance Checklist for Software Supply Chain Risk Management,” https://buildsecurityin.us-cert.gov/sites/default/files/20101208-SwAChecklist.pdf

“Software Supply Chain Risk Management & Due-Diligence,” Software Assurance Pocket Guide Series: Acquisition & Outsourcing, Vol II, Version 1.2, June 16, 2009, https://buildsecurityin.us-cert.gov/sites/default/files/DueDiligenceMWV12_01AM090909.pdf

Third Party Software Security Working Group, “Appropriate Software Security Control Types for Third Party Service and Product Providers,” Financial Services Information Sharing and Analysis Center, 2013, http://docs.ismgcorp.com/files/external/WP_FSISAC_Third_Party_Software_Security_Working_Group.pdf

Young, Lisa, “SEI Cyber Minute: CERT Resilience Management Model (RMM), 2016, http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=485774

Contact Information

Mark Sherman

Technical Director
Cyber Security Foundations
Telephone: +1 412-268-9223
Email: mssherman@sei.cmu.edu

Web

www.sei.cmu.edu
www.sei.cmu.edu/contact.cfm

U.S. Mail

Software Engineering Institute
Customer Relations
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA

Customer Relations

Email: info@sei.cmu.edu

Telephone:	+1 412-268-5800
SEI Phone:	+1 412-268-5800
SEI Fax:	+1 412-268-6257

© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.