



Bringing Operational Knowledge to Secure Development

Steve Lipner
SAFECode and Microsoft Corporation

September 2011



The Software Assurance Forum for Excellence in Code (SAFECode) is a global, industry-led effort to identify and promote best practices for developing and delivering more secure and reliable software, hardware and services





- Secure development practices
- Vulnerability response and root cause
- Vulnerability response, root cause and security process
- Summary



Fundamental Practices for Secure Software Development - Second Edition

- **New** in 2nd Edition:
 - **Verification** methods and tools were developed for each listed practice to help managers confirm whether a practice was applied.
 - **Common Weakness Enumeration (CWE)** references were added to each practice to provide a more detailed illustration of the security issues these practices aim to resolve.





| Section | Practice |
|----------------------------|--|
| Secure Design Principles | Threat Modeling |
| | Use Least Privilege |
| | Implement Sandboxing |
| Secure Coding Practices | Minimize Use of Unsafe String and Buffer Functions |
| | Validate Input and Output to Mitigate Common Vulnerabilities |
| | Use Robust Integer Operations for Dynamic Memory Allocations and Array Offsets |
| | Use Anti-Cross Site Scripting (XSS) Libraries |
| | Use Canonical Data Formats |
| | Avoid String Concatenation for Dynamic SQL Statements |
| | Eliminate Weak Cryptography |
| | Use Logging and Tracing |
| Testing Recommendations | Determine Attack Surface |
| | Use Appropriate Testing Tools |
| | Perform Fuzz / Robustness Testing |
| | Perform Penetration Testing |
| Technology Recommendations | Use a Current Compiler Toolset |
| | Use Static Analysis Tools |



- All development organizations must practice vulnerability response
 - No perfect software
 - Active communities of vulnerability finders
- Naïve organization fixes vulnerabilities as reported
 - Again and again and again
- Mature organization integrates root cause analysis into response
 - Fixes “internal finds” with external reports
 - Avoids repeated reports of same vulnerability
- Root cause analysis increases “time to fix” - hence need to cooperate with vulnerability finders



- Vulnerability response and root cause analysis advise secure development practices
 - What practices are important?
 - What practices are ineffective?
 - What practices are missing?
- SAFECode Secure Development Practices associated with CWE vulnerability classes
 - CWE can be effective for characterizing vulnerabilities, development tools, and mitigations
 - Identifying correct CWE of vulnerabilities is important!
- If a CWE is prevalent “in the wild” and no tools, processes, or mitigations address it - a clear area for process improvement



- Operational experience and root cause analysis are key to effective development security
 - Security response
 - Secure development process
- CWE is a useful tool for process improvement
- "Top N" lists are a good check on process, but a sound process covers broad classes of vulnerabilities
- Process improvement is important! No perfect secure development process (or secure products) in the real world



www.safecode.org
Twitter: @safecodeforum
Blog: <http://blog.safecode.org>

Steve Lipner
Senior Director of Security Engineering Strategy,
Trustworthy Computing Security, Microsoft Corporation
SAFECode Board Chair
(425) 705-5082
slipner@microsoft.com

10001
01111
10001
11110
10001

SAFECode
Software Assurance Forum for Excellence in Code
Driving Security and Integrity



STOP



SAFECode Vision

Trusted and reliable information
and communications systems
powered by high-quality, secure
software development practices



SAFECode Mission

As a center of excellence for vendor software assurance practices, SAFECode unites subject matter experts with unparalleled experience in managing complex global processes for software sourcing, development and delivery to:

- Encourage broad industry adoption of proven software security, integrity and authenticity practices
- Drive clarity into vendor software assurance practices to empower customers and other key stakeholders to better manage risk
- Foster a trusted exchange of insights that advance software assurance practices



SAFECode Outreach

SAFECode and its work is well known by government stakeholders in both the EU and US.

- SAFECode facilitates direct member interaction with influential officials in Brussels and Washington.
- Our work is frequently cited by homeland security officials, policymakers and international standards organizations as fundamental to their thinking on software security issues.

SAFECode is also an active contributor to the technical community and our representatives and members are frequent speakers and participants in key industry events and initiatives throughout the US and Europe.



Outreach Initiatives



Data Protection

Home » Data Protection » Application Security

NEWS

Code Security: SAFECode report highlights best practices

The report sheds light on what companies like Adobe, Juniper, EMC and Microsoft are doing to bake security into their code. Given Adobe's troubles, the process remains a challenge.

» Comments

By Bill Brenner, Senior Editor

June 14, 2010 — CSO —

A new report from the Software Assurance Forum for Excellence in Code (SAFECode) sheds light on how vendors are trying to work more secure coding into the product development process.

The vendors contributing to the report are SAFECode members who have enjoyed the process of developing secure development guides.

Like +7 0



SAFECode updates secure development guide

Angela Moscaritolo February 08, 2011

PRINT EMAIL REPRINT PERMISSIONS TEXT: A | A | A

Tweet 0 Like

The Software Assurance Forum for Excellence in Code (SAFECode), a nonprofit seeking to advance software assurance, released on Tuesday an updated guidance document outlining the most effective secure development practices in use today. The free report builds upon the first edition by including verification methods and tools that can be used to confirm whether development teams have followed the guidelines. The report is intended to help organizations improve their software security programs and encourage the use of secure development methods. - AM

RELATED ARTICLES

- SAFECode releases software integrity guidance
- Paul Kurtz, executive director, SAFECode: partner, Good



Fresh advice on building safer software

SAFECode updates best practices for secure development

By William Jackson • Feb 08, 2011

An industry group promoting reliability in commercial software has released a set of guidelines for secure software development, with a focus on the use of recommended practices.

The second edition of "Fundamental Practices for Secure Software Development" reflects changes in the industry and was published to evolve the guidelines.

The software assurance forum for excellence in code (SAFECode) has published a new report on secure development practices for those who develop software for the government.



The Kaspersky Lab Security News Service

New Study Sees Need for Better Software Integrity Controls

by Inis Fisher

Twitter Facebook LinkedIn StumbleUpon Share +7 0

3 Comments

Software security has become one of the more widely discussed and debated topics in the security industry in the past few years, as many software vendors and enterprises both large and small have begun to focus more attention on improving the processes they have in place for producing software. But far less attention has been shone on the security of the software supply chain, an increasingly thorny problem in today's software development environment. As more and more software makers and enterprises have turned to third parties, both domestic and foreign, to produce software, new considerations have arisen. A new study from InformationWeek Government Threat Post, "Increase Risk in the Software Supply Chain," looked at the threat to software development that has become well-understood.



E-mail this page | Print this page | BOOKMARK

SAFECode Issues Best Practices For Writing Secure Code

Nonprofit members Adobe, EMC, Juniper, Microsoft, Nokia, SAP, and Symantec share secure development methods

Feb 08, 2011 | 12:44 PM | 0 Comments

By Kelly Jackson Higgins
Dark Reading

The nonprofit Software Assurance Forum for Excellence in Code, a.k.a. SAFECode, today published a best practices guide for the software development community based on techniques and processes used by its high-profile membership.



From the Editors: Opening up about security

By SD Times Editorial Board

August 15, 2010 — (Page 1 of 2)

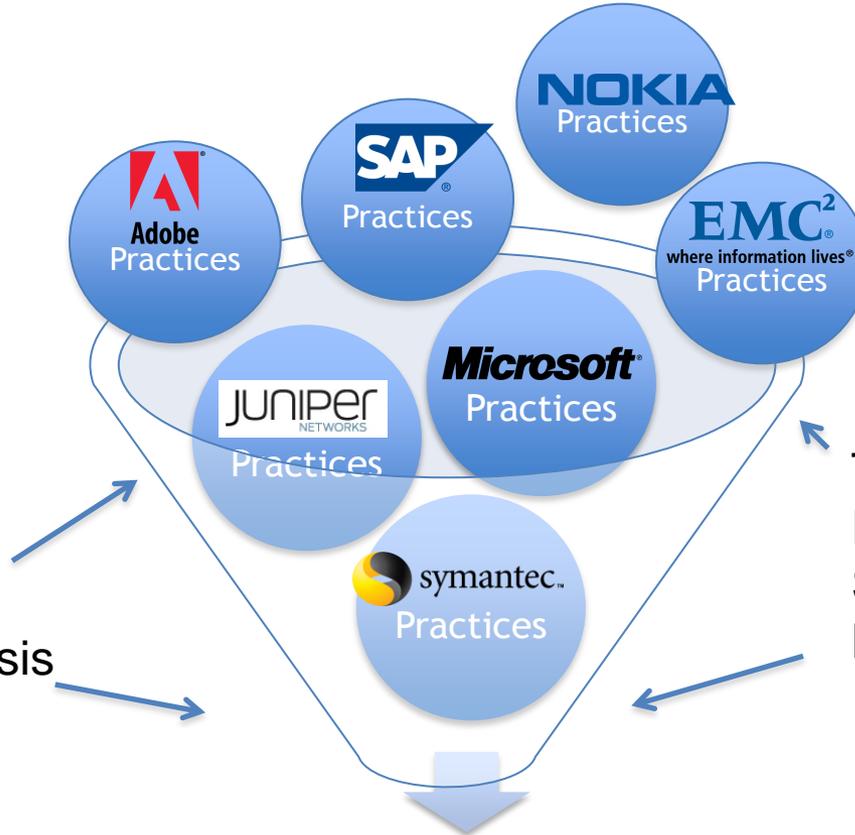
We are pleased that members of the Software Assurance Forum for Excellence in Code were active participants at the Black Hat Technical Security Conference and have begun to work together to devise new security best practices.

SAFECode members, including giants Adobe and Microsoft, took time to listen to the security community during a brainstorming session about how to produce more secure software over the next decade. Those ideas will be incorporated into white papers and best practices documents.



Working Group Process

Common Practice Identification, Analysis and Discussion



Trusted Sharing Environment Strengthened by NDA

SAFECode-Recommended Practices



Software Assurance: Confidence that software, hardware and services are free from intentional and unintentional vulnerabilities and that the software functions as intended.

In practice, software vendors take action in three key, overlapping areas to achieve software assurance—security, authenticity and integrity.





Security: Security threats to the software are anticipated and addressed during the software's design, development and testing through secure engineering practices. This requires a focus on code quality and functional requirements to reduce unintentional vulnerabilities in the code.





Integrity: Security threats to the software are addressed in the processes used to source software components, create software components, and deliver software to customers. These processes contain controls to enhance confidence that the software was not modified without the consent of the supplier.



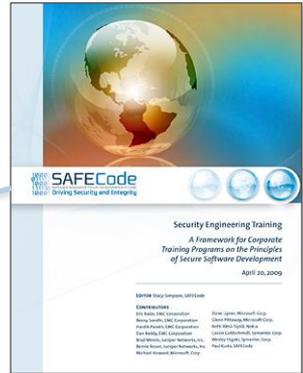


Authenticity: The software is not counterfeit and the software supplier provides customers ways to differentiate genuine from counterfeit software.

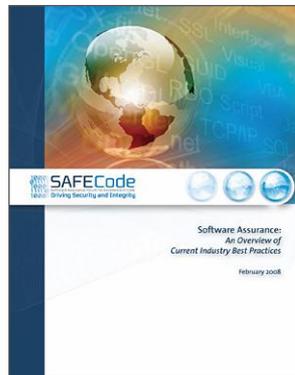




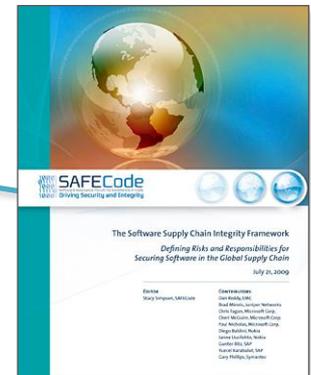
Secure Software Development



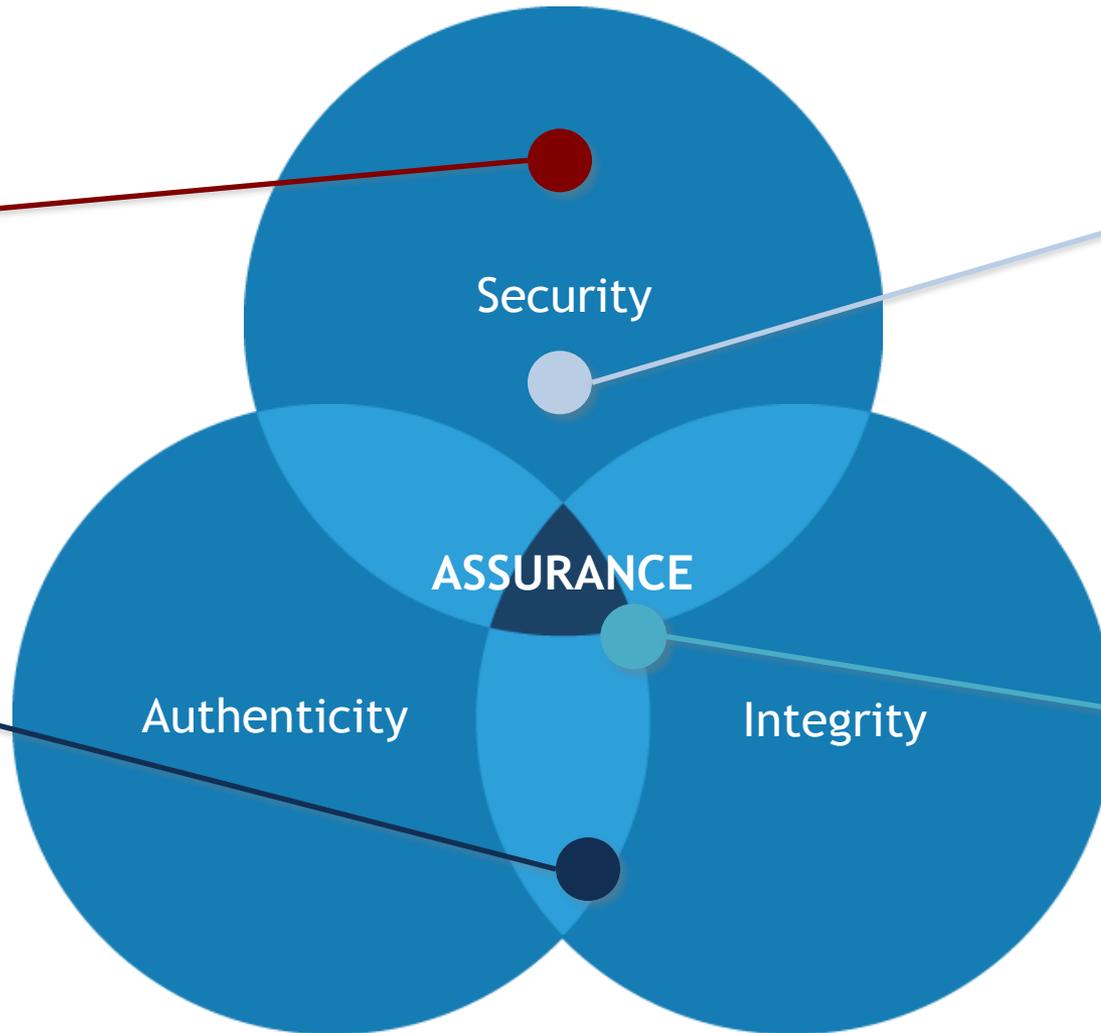
Training



Software Integrity Controls



Software Integrity Framework





Fundamental Practices for Secure Software Development - Second Edition

- **Focus:** Provide a foundational set of secure development practices based on an analysis of the real-world actions of SAFECode members
- **Key Objectives:** Help others initiate or improve their own software security programs and encourage the industry-wide adoption of fundamental secure development methods.





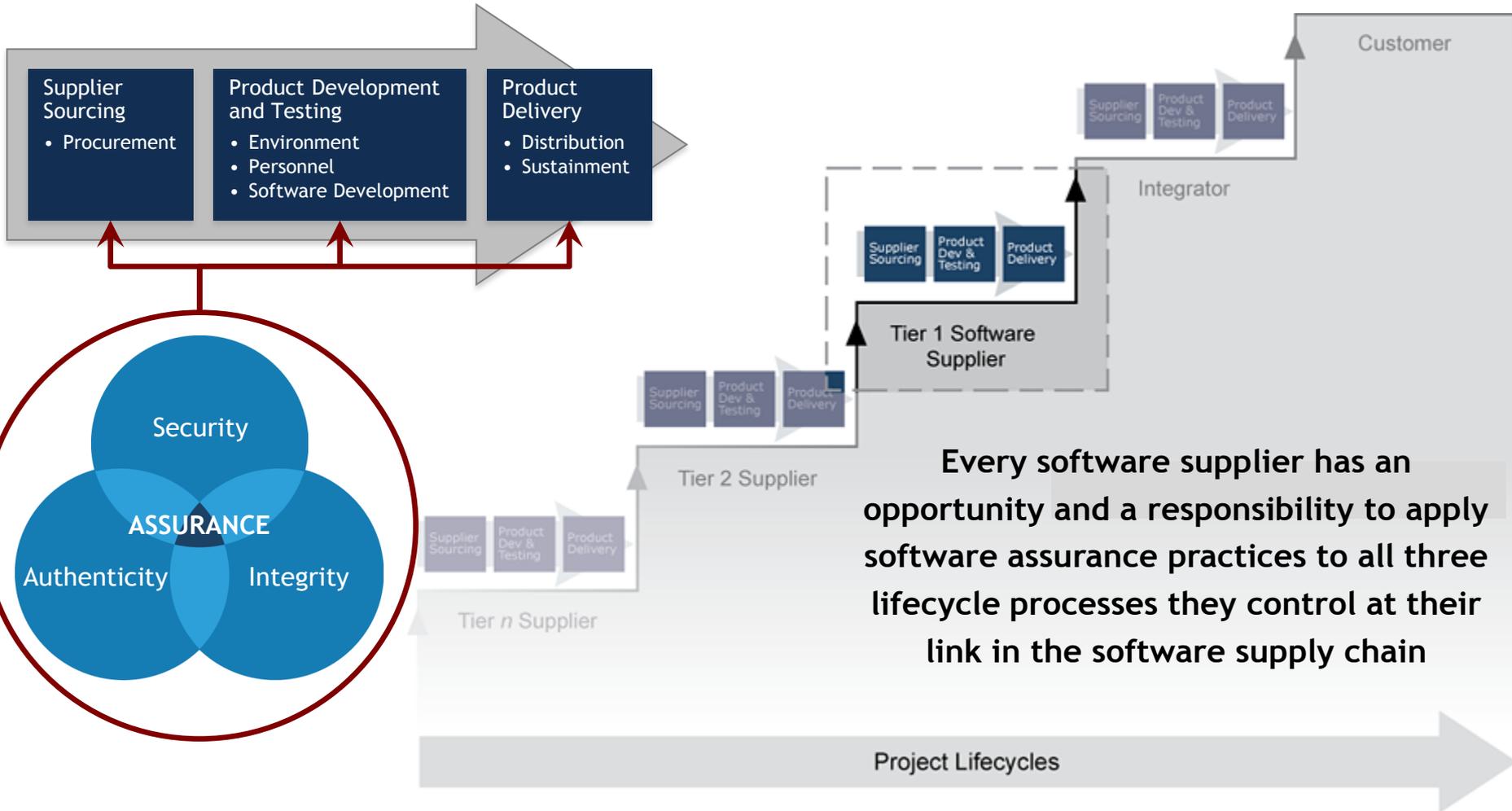
The Software Supply Chain Integrity Framework: Defining Risks and Responsibilities for Securing Software in the Global Supply Chain

- **Focus:** Provide the first industry-driven framework for analyzing and describing the efforts of software suppliers to mitigate the potential that software could be intentionally compromised during its sourcing, development or distribution.
- **Key Objectives:** Create a foundation for evaluating and describing software supply chain risks to enable the identification and analysis of mitigating controls and practices.





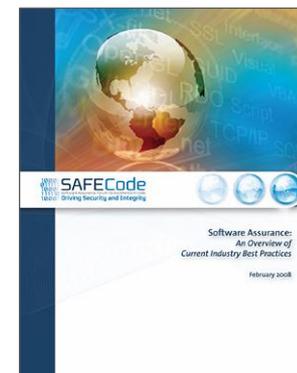
Software Supply Chain Integrity Framework





Software Integrity Controls: An Assurance-Based Approach to Minimizing Risks in the Software Supply Chain

- **Focus:** Provide actionable recommendations for minimizing the risk of vulnerabilities being inserted into a software product during its sourcing, development and distribution.
- **Key Objectives:** Help others initiate or improve their software supply chain security programs and encourage broad industry adoption of software integrity controls.





Software Supply Chain Integrity Controls

| Processes | Controls | | |
|---------------------------------|---|---|--|
| Software Sourcing | Vendor Contractual Integrity Controls | <ul style="list-style-type: none"> •Defined expectations •Ownership and responsibilities | <ul style="list-style-type: none"> •Vulnerability response •Security training |
| | Vendor Technical Integrity Controls for Suppliers | <ul style="list-style-type: none"> •Secure transfer •Sharing of system and network resources •Malware scanning | <ul style="list-style-type: none"> •Secure storage •Code exchange |
| Software Development & Testing | Technical Controls | <ul style="list-style-type: none"> •People security •Physical security •Network security | <ul style="list-style-type: none"> •Code repository security •Build environment security |
| | Security Testing Controls | <ul style="list-style-type: none"> •Peer review | <ul style="list-style-type: none"> •Testing for secure code |
| Software Delivery & Sustainment | Publishing & Dissemination Controls | <ul style="list-style-type: none"> •Malware scanning •Code signing | <ul style="list-style-type: none"> •Delivery •Transfer |
| | Authenticity Controls | <ul style="list-style-type: none"> •Cryptographic hashed or digitally signed components •Notification technology | <ul style="list-style-type: none"> •Authentic verification during program execution |
| | Product Deployment and Sustainment Controls | <ul style="list-style-type: none"> •Patching •Secure configurations | <ul style="list-style-type: none"> •Custom code extension |