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

Principles for Software Assurance Assessment

A Framework for Examining the Secure Development Processes of Commercial Technology Providers

PRIMARY AUTHORS:

Shaun Gilmore, Senior Security Program Manager, Trustworthy Computing, Microsoft Corporation

Reeny Sondhi, Senior Director, Product Security Engineering, EMC Corporation

Stacy Simpson, Director, SAFECode

Table of Contents

Foreword

Customers increasingly ask questions about the software assurance practices of their suppliers and how they can be confident in the security of the software these suppliers produce. The answers to these questions help customers select and purchase more secure technology products; they also further enable them to better assess their technology suppliers and manage their broader information technology risk. This has led to a number of recent efforts in industry and government to define a path forward for assessing the security of commercial technology products.

Many of these efforts have provided useful points of reference for customers seeking to understand and assess software security. However, these initiatives take many different forms – from emerging global standards, to ad hoc program efforts by industry and government groups, to published perspectives from security vendors and other interested parties. While all of these initiatives bring value to the overall software security landscape, and to the specific organizations that they may serve, their varying approaches and claims can create challenges for those seeking to select the most effective assessment approach for their organizations.

Not only are there negative implications of poorly formed security evaluations for technology providers, there are also less recognized, but significant, drawbacks for customers that base their procurement decision-making around an incomplete or misleading assessment.

This paper provides a framework for examining the secure development process of commercial technology providers. It is designed to help readers select the most appropriate assessment method for their needs, and provides guidance to help them develop a process-based assessment for use in cases when an appropriate international standard does not apply.

Software assurance encompasses the development and implementation of methods and processes for ensuring that software functions as intended, while mitigating the risks of vulnerabilities and malicious code that could bring harm to the end user.

Methodology

As with other SAFECode work, this paper is grounded in an analysis of what SAFECode member organizations actually see and do in their day-to-day software assurance efforts. We reviewed with our members the types of security documentation they provide to customers, the questions and documentation most often requested by customers, their experiences with current standards and evaluation methods, and their assessments of the impact of customer security reviews on their internal secure development processes.

To broaden our perspective, we also reached out to a number of representatives of prominent enterprises with experience in managing the security of acquired software. Through informal conversations and more formal outreach, we gathered feedback on their biggest challenges in dealing with software suppliers, what is most helpful to their risk assessment process, and their requirements for effective security assessment. The assessment framework put forth in this paper was developed based on this year-long effort. In this way, we hope to present a framework that is effective, practical, and spans a diverse and global set of customer and supplier needs

Problem Statement

All customers have software assurance concerns, and all customers want confidence that commercial software is secure and reliable. When acquiring software, customers have a primary concern that they may be introducing new vulnerabilities into their IT environments. Software vulnerabilities can compromise customer data, disrupt business services, and jeopardize trust. Therefore, customers require that software be developed in a way that minimizes the number of vulnerabilities, and customers expect suppliers to have appropriate update mechanisms for use when vulnerabilities emerge. This is only achieved when software is created and sustained using best practices for a secure software development lifecycle.

To gain the necessary confidence in acquired software, customers need a method for assessing the security of the software, including the impact the software may have on the organization's risk posture. A process-based assessment of a supplier's software assurance practices can deliver this confidence, empowering customers to better manage risk.

The highest degree of confidence is achieved when a supplier conforms to international standards for software assurance and secure development. Conformance to international standards indicates a more formal commitment by the supplier to software assurance, and to a well-structured and governed approach to integrating security into the software development lifecycle. However, historically we have lacked a broadly accepted industry standard for software assurance and secure development that performs a role similar to that of ISO 9000 for quality. Common Criteria [ISO 15408] is a widely used mechanism for evaluating security capabilities in commercial IT products. However, it primarily focuses on product security features, not on the security development process.

Recently, there have been positive developments on the standardization front, and new standards that focus on secure development processes are emerging. IEC/ISA-62443[1] is a standard for industrial automation and control systems that has become more broadly adopted. It addresses Security Development Lifecycle Assurance (SDLA) processes for products and software applications integrated in an industrial automation and control system. Also, the FDA has accepted IEC/ISA-62443 as a consensus standard for medical devices. In addition, ISO/IEC 27034 is an international standard for specifying secure development lifecycle processes. Part 1[2] of the standard is published, and provides an overview of a mature security development process. Additional sections, including a validation framework, are currently under development.

Even with this progress, however, we recognize that these standards are not widely adopted. While mature, IEC/ISA 62443 has a specific, narrow use case on applications in industrial environments. ISO/IEC 27034 is more broadly applicable across different types of applications, but its validation framework is still being developed.

The lack of a broadly accepted industry standard has deprived the marketplace of a consensus approach to assessing the software development process of a supplier. In an attempt to fill this gap, numerous ad hoc assessment methods have been created, which, though well-intentioned, have not been effective in helping customers manage, and suppliers communicate, risk. While there is widespread agreement on the importance of a process-based approach to software assurance, this principle is not always reflected in

SAFECode believes software assurance assessment efforts should focus on supporting the development and acceptance of international standards. Examples of relevant standards include:

ISO/IEC 27034-1:2011 Information technology – Security techniques – Application security is an international standard for secure development processes. Additional parts, covering other related areas, are under development.

IEC/ISA-62443 is a standard for industrial automation and control systems. In particular, part 4-1 of this standard addresses product development.

1 Applies to automation solutions and systems used in industrial applications including building, transportation and medical. Includes functional requirements for products and systems as well as requirements for the processes of the operator (based on ISO/IEC 27001) and the integrator and the product development lifecycle of products. See http://isa99.isa.org/ISA99%20Wiki/Home.aspx .

2 http://www.iso.org/iso/catalogue_detail.htm?csnumber=44378

currently proposed efforts to assess software security. Further, procurement decision-makers do not always have the knowledge required to properly assess a software development process. These factors make it difficult to accurately quantify and compare risk factors during the procurement process, contributing to marketplace confusion and the erosion of trust between supplier and customer. The following table highlights a few key negative impacts for the supplier and customer due to the lack of a common assessment methodology.

Customer Concerns	Supplier Concerns
• No single, consistent way to achieve clear, testable, repeatable ways to build and maintain a fact-based trust between suppliers and customers • General lack of awareness within many enterprises of what to look for when evaluating software • Inadequate insight into what security due diligence has been performed on the components included in software • Need to understand whether a company has a secure development process and whether that process was applied to the specific product being purchased	• No scalable way to provide multiple customers with the information they require to make purchase decisions • Clearing multiple, often diverse, customer hurdles is costly and diverts resources from critical engineering tasks – a problem more acute for small and mid-sized vendors • No current agreement on what information customers should be requesting; some requests do not align well with real-world secure development practices

Framework Overview

Despite the lack of a broadly adopted software assurance standard, we have a large body of work to leverage within SAFECode, including work accomplished and ongoing, to help assess the software development process of a supplier, while continuing support of international standardization. The framework put forth in this paper aims to help bridge the current standardization gap while at the same time promoting a process-centric approach that aligns to emerging international standards.

The core principle behind the SAFECode framework is that a software assurance assessment should primarily focus on the secure software development process and its application to the product being assessed, while taking into consideration the context of a product's intended operating environment. There is no single practice, tool, or checklist that acts as a silver bullet and guarantees better software assurance. Rather, the efficacy and efficiency of software security practices and tools varies based on how they are applied and whether they are implemented as part of a holistic software development process within each unique organization.

"The core principle behind the SAFECode framework is that a software assurance assessment should primarily focus on the secure software development process and its application to the product being assessed, while taking into consideration the context of a product's intended operating environment."

With that principle understood, we recognize that the maturity of secure development practices varies among technology suppliers. This has created challenges for assessing the processes of suppliers who are either unable or unwilling to provide enough information for an informative process-based review.

The result is that despite the key benefits of a process-centric review, there is no one-size-fits-all approach to assessing the security of commercial software. Rather, the approach that customers take in evaluating the security of purchased software must reflect both their internal risk management requirements and the maturity of the supplier in question. Therefore, SAFECode has developed a tiered approach to software security assessment.

In some cases, customer risk management requirements for software assurance assessment may require evidence to support a supplier's claims. Some may require more insight not only into the software assurance process itself, but also into how it was applied to the product under consideration. An effective software assurance assessment framework must address these customer requirements to achieve broad adoption.

SAFECode's Software Assurance Assessment Framework was developed to address all of the above requirements and is grounded in the following principles.

"The approach that customers take in evaluating the security of purchased software must reflect both their internal risk management requirements and the maturity of the supplier in question."

Guiding Principles for Software Security Assessment

1. Software assurance is not achieved by a single practice, tool, or checklist; rather it is the result of a comprehensive secure software engineering process.
2. The diversity of approaches used by organizations acquiring software and the unequal adoption of software assurance practices by IT development organizations has made it clear that we need a tiered approach for assessing the security of acquired software based on the maturity of the technology provider developing the software.
3. The current problems faced by many customers and suppliers require an immediate solution in the short term, and comprehensive, widely accepted international standards in the medium/long term.
4. Customers may require evidence to support a supplier's claims.
5. Customers need insight into the assurance process at both the corporate and product levels to support their risk management needs.

The SAFECode Supplier Software Assurance Assessment Framework

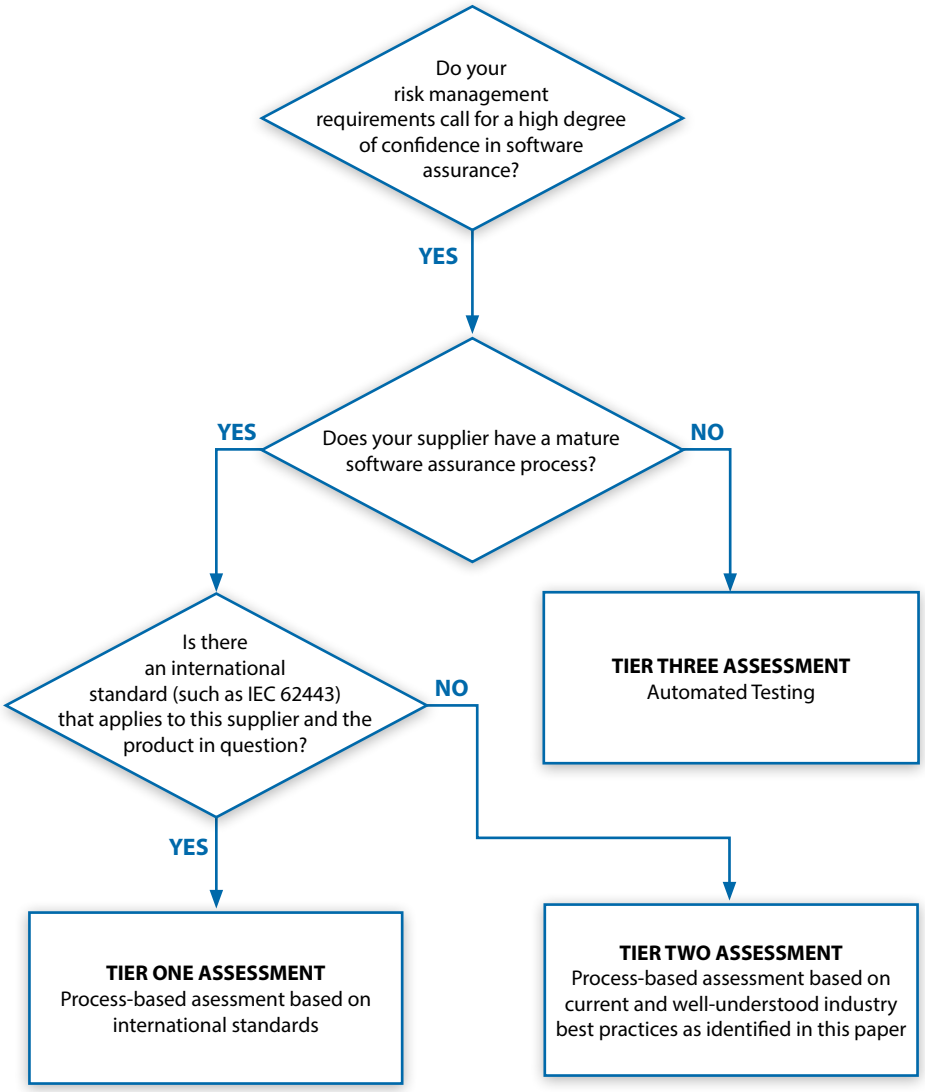


Figure 1: Overview of SAFECode Assessment Framework

What Are Your Risk Management Requirements?

The SAFECode Supplier Software Assurance Assessment Framework prescribed in this document begins with the internal risk management requirements of the customer. The customer needs to make a determination, based on an internal risk assessment[3], of what degree of confidence in the assurance of an application is necessary or tolerable for a given application context.

Key Indicators of Supplier Maturity

- Provide a way to report a security vulnerability
- Have a central group tasked with software security responsibilities
- Offer public documentation of software assurance process

3 ISO/IEC 27005:2011 Information technology – Security techniques – Information security risk management provides guidance for completing an internal risk assessment. More: http://www.iso.org/iso/catalogue_detail?csnumber=56742

If it is determined that a high-degree of confidence is necessary, then ideally a customer will select a supplier with a demonstrable, mature secure development lifecycle, and perform a Tier One or Tier Two assessment. However, in cases where a customer must work with a supplier that is either unable or unwilling to provide sufficient detail on its secure development process, then the customer can pursue a Tier Three assessment.

The Tier Three Assessment

If the supplier lacks a mature process for software security, or is unable or unwilling to provide information on that process, then using testing tools or other product testing techniques to detect security flaws in product code can serve as the basis of an assessment approach *(Tier Three Assessment in Figure 1)*. This approach can be effective as a way of detecting simple security flaws, and can provide a customer with a limited degree of confidence in a product's security. For example, binary code analysis tools can be particularly well-suited to this approach because they are able to analyze compiled code for certain vulnerabilities without executing the application.

These tools can be an effective and scalable approach, when viewed as one part of a supplier's overall secure development process and a supplement to other secure development activities such as threat modeling and manual code reviews/testing. As we'll describe below, when working with a security-mature supplier, a customer will derive more value from understanding how a supplier uses testing within its overall development process than from performing or reviewing the test results themselves.

Despite this value, there are limitations to using automated testing tools for assessment purposes due to their inability to understand the functional design of the software. Some of these limitations can be managed, but customers generally will have to accept a greater level of risk when using this approach.

While a detailed overview of best practices for automated testing and its role in security assessment is outside the scope of this paper, readers can find some more information on this approach in a white paper from the FS-ISAC Third Party Software Security Working Group: Appropriate Software Security Control Types for Third Party Service and Product Providers.

The paper is freely available at: http://docs.ismgcorp.com/files/external/WP_FSISAC_Third_Party_Software_Security_Working_Group.pdf

The Tier One and Tier Two Assessments

If a customer determines that its risk management requirements call for a high degree of confidence in assurance for a given software application and a supplier has (1) a mature secure software development process, and (2) can provide insight into its software security methodology, then a process-based assessment approach is recommended. In the case where an international standard is available and applicable to a supplier, that standard should drive the scope of the assessment *(Tier One Assessment in Figure 1)*.

If an industry standard does not apply, the assessment should be based on current and well understood industry best practices *(Tier Two Assessment in Figure 1)*. This paper provides guidance and examples on how to structure a Tier Two process-based evaluation.

A mature software security process will have three key elements, each of which should be reviewed as part of the supplier assessment. These include:

1. Secure development and integration practices
2. Product security governance
3. Vulnerability response process

Secure Development and Integration Practices

In 2008, SAFECode first published *Fundamental Practices for Secure Software Development*[4]. The paper was updated in 2011. Informed by existing models, including OWASP, CVE, CWE and the Microsoft SDL, its objective was to *"aid others within the software industry in adopting and using these software assurance best practices effectively."* These secure software development practices are core to mature SDLCs, and SAFECode believes they provide customers and their suppliers with a common set of fact-based criteria for assessing secure development.

Using these practices as a foundation, customers can use the following questions to determine how the software was designed, built, and implemented, and how externally sourced components were examined prior to their inclusion in the software:

- Does the supplier define product-specific security requirements as part of its development lifecycle?
- Does the supplier conduct architectural risk analysis or threat modeling as part of its product lifecycle, and define appropriate mitigations?
- Does the supplier perform automated static code reviews to identify security defects introduced during coding?
- Does the supplier perform automated dynamic security testing to identify common security vulnerabilities?
- Does the supplier triage security defects identified from the above activities and remediate them as part of its lifecycle?
- Does the supplier have a supply chain risk management process to manage the security and integrity of sourced components?[5, 6]

Product Security Governance

Suppliers with a mature secure software assurance process typically are able to demonstrate a robust governance structure that provides oversight to the complete assurance process. Additionally, they are able to assure that the software assurance process and the application of the process are well understood within their organizations. The key characteristics to review of a good governance structure include:

- Does the supplier require security training for its product development team and a method to ensure that the requirements of its SDL are broadly understood?[7, 8]
- Do the appropriate levels of management in the organization review and sign off on the security posture of the product?
- Does the supplier conduct proper roadmap planning to identify future steps for remediating any unmitigated findings?

Governance should support a proper review or assessment of any risks identified as part of the application of the process and the activities performed during the product development lifecycle. A supplier cannot be expected to remediate every security defect identified during

Examples of SAFECode Secure Development Practices
Threat Modeling
Use Least Privilege
Implement Sandboxing
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 Tracking
Determine Attack Surface
Use Appropriate Testing Tools
Perform Fuzz/Robustness Testing
Perform Penetration Testing
Use a Current Compiler Toolset
Use Static Analysis Tools

4 Fundamental Practices for Secure Software Development, 2nd Edition: www.safecode.org/wp-content/uploads/2014/09/SAFECode_Dev_Practices0211.pdf

5 ISO/IEC 27036-1:2014 Information technology -- Security techniques -- Information security for supplier relationships – Part 1: Overview and concepts provides additional guidance on managing supplier relationships. More: http://www.iso.org/iso/catalogue_detail.htm?csnumber=59648

6 Additional Guidance -- Open Group: http://www.opengroup.org/standards/trusted-technology-standards

7 SAFECode Guidance: Security Engineering Training: A Framework for Corporate Training Programs on the Principles of Secure Software Development. More: http://www.safecode.org/publications/SAFECode_Training0409.pdf

8 SAFECode Free Training Modules: https://training.safecode.org/

the lifecycle, but having a risk-based approach to dealing with security defects demonstrates the supplier's maturity in adhering to the process.

Vulnerability Response Process

Customers must also assess the supplier's assurance regarding sustainment. The supplier should be expected to share the process it follows for vulnerability remediation once software has been released to customers. The key questions to ask suppliers around vulnerability response include:

- Does the supplier have a way for researchers or customers to report a security vulnerability to it?[9]
- Does the supplier issue security advisories or alerts as a way to notify customers of remediation of security vulnerabilities?
- Does the supplier use a CVE ID to list the vulnerability in the National Vulnerability Database?

Examples of Tier Two Process Assessments

To expand upon the above assessment elements and key questions and to further illustrate a Tier Two process assessment, we've included snapshots of two example process assessments as appendices to this document: a sample assessment questionnaire provided by Boeing (Appendix A), and "Appropriate Software Security Control Types for Third Party Service and Product Providers," published by the FS-ISAC (Appendix B).

Assessment Methodology

Using the framework outlined above as the basis for the scope of the assessment, customers should then select a methodology for applying it to a specific supplier based on requirements derived from its internal risk management process. There are three approaches to consider while examining a secure development process based on risk management requirements:

1. **Transparency of process documentation:** Documenting their secure software development lifecycle or methodology is becoming common practice for suppliers; many software suppliers openly document their process on security-focused web pages, by way of white papers or public blog posts. This public documentation includes a detailed view into the practices the supplier considers to be important to its product development methodology, which directly relates to the process outlined above.
2. **Sharing under NDA:** If public documentation is not available or not adequate in detail, the supplier may be able to share the details under NDA with a customer to demonstrate trust in the supplier-customer relationship and provide insight into the process and governance applied on the dimensions outlined above.
3. **Third-Party Validation:** In the case where third-party validation is a requirement, the assessment should be based on the elements of the process and governance provided above.

Ideally, any assessment in this field should be done in accordance with emerging international standards such as ISO/IEC 27034 and IEC/ISA-62443 to provide widely acceptable evidence of the soundness of a supplier's process and of the supplier's implementation of the process, including risk-based governance of secure

9 ISO/IEC 29147:2014 Information technology -- Security techniques -- Vulnerability disclosure provides guidance around details of the methods a vendor should use to address issues related to vulnerability disclosure. More: http://www.iso.org/iso/catalogue_detail.htm?csnumber=45170

development activities. Developer conformance to these standards indicates a formal commitment to secure development and a well-structured management approach to integrating security into the software development lifecycle. For some specific use cases, such as industrial automation and control systems, third-party validation against IEC-62443 exists (based on the draft of 4-1) and is expected to be offered in 2015.[10]

While a validation framework for ISO 27034 and other standards applicable to more general purpose applications is still maturing, the framework in this paper in the form of the Tier 2 assessment provides an interim approach grounded in governance and process assessment.

Product Adherence to the Process

Last is a method to capture whether the supplier has applied the aforementioned process and governance to a specific product being procured by the customer. This can be covered by asking the supplier to provide self-attestation for product-level compliance with the process as well as adherence to the governance established by the supplier on the process.

Summary and Future Directions

To gain the necessary confidence in acquired software, customers need a method for assessing the security of the software and the impact the software may have on their organization's risk posture. The approach that customers take in evaluating the security of purchased software must reflect both their internal risk management requirements and the maturity of the supplier in question.

The framework described within this paper provides a tiered approach for a customer to gain confidence in the assurance of acquired software. It focuses primarily on assessing a supplier's secure development lifecycle process and provides guidance and examples on evaluating the key elements of such a process: 1. secure development and integration practices; 2. product security governance; and 3. vulnerability response process. This process-based assessment approach is also the underlying theme of emerging international standards in this space.

In accordance with our mission, our intent at SAFECode is to continue to promote collaboration between customers and suppliers to drive towards a common, internationally acceptable practical framework that provides transparency and trust in assessing software security. We are encouraged to see that progress is being made with the ongoing development of standards such as the ISO/IEC 27034 series and IEC/ISA-62443 series. Yet, we recognize the urgency behind this challenge and offer this framework of less formal assessment approaches as an intermediate step toward that future.

SAFECode provides a number of free resources that offer guidance on an effective software security process. These include:

- Fundamental Practices for Secure Software Development, 2nd Edition
- Practices for Secure Development of Cloud Applications
- Guidance for Agile Practitioners
- Overview of Software Integrity Controls

These are available to download at no cost by visiting: www.safecode.org/publications/

10 http://www.tuev-sued.de/uploads/images/1444299446699022170094/tuv-sud-iec-62443-certification-lowres.pdf

Appendix A: Sample Questionnaire for a Process-Based Assessment: Boeing

Secure Development Elements

- **a.** Does the supplier implement a secure development process that includes activities for requirements definition, design, implementation, and test phases?
- **b.** Does the supplier apply ISO/IEC 27034 as an internal standard for secure development?
- **c.** Does the supplier's software development process include a specification of application security controls (formal security requirements)?
- **d.** Does the supplier advertise its application security controls (formal security requirements)?
- **e.** Are the supplier's standard security architectures based on Threat Models?
- **f.** Does the supplier create Threat Models to identify significant attack vectors for each published application?
- **g.** Does the supplier update Threat Models at each minor version release?
- **h.** Does the supplier's secure development process use automated security testing tools and is the use of these tools included in the specification of application security controls?
- **i.** Does the supplier use automated standards-based assessment tools in its test and assessment methodologies?
- **j.** Do the supplier's test and assessment methodologies (to include appropriate build-in testing tools) of its secure development process produce a high quality, repeatable result?
- **k.** Does the supplier include secure coding standards in the software security policy?
- **l.** Do the supplier's automated standards-based assessment tools utilize public vulnerability and security flaw repositories (e.g., CWE, CVE, CAPEC, etc.)?
- **m.** Does the supplier routinely calibrate the test and assessment methodology against the latest threat landscape (e.g., through security response, root cause analysis, third-party review/assessment, etc.)?

Secure Supply Chain Elements

- **a.** Does the supplier have a process that manages risk from its supply chain?
- **b.** Does the supplier have appropriate configuration management controls of all software components used in the product, including third-party and sourced software libraries or components?
- **c.** Does the supplier identify all binary executables (i.e., compiled or byte code; source code is not required) of the software, including all libraries or components?
- **d.** Does the supplier disclose the development origin of all software components (i.e., compiled or byte code; source code is not required) used in the product, including third-party and sourced software libraries or components?
- **e.** Does the supplier provide secure deployment guidelines for its security relevant products?

Secure Sustainment Elements

- **a.** Does the supplier routinely disclose vulnerabilities and prepare customers for patch deployment?
- **b.** Does the supplier provide clear vulnerability reporting methods, to include reporting to commonly used repositories (e.g., CVE), and provide frequent feedback on submitted vulnerabilities?
- **c.** Does the supplier prepare remediation roadmaps for significant security issues?
- **d.** Does the supplier have a history and reputation for actively patching reported vulnerabilities?
- **e.** Does the supplier engage with independent researchers to encourage vulnerability discovery?

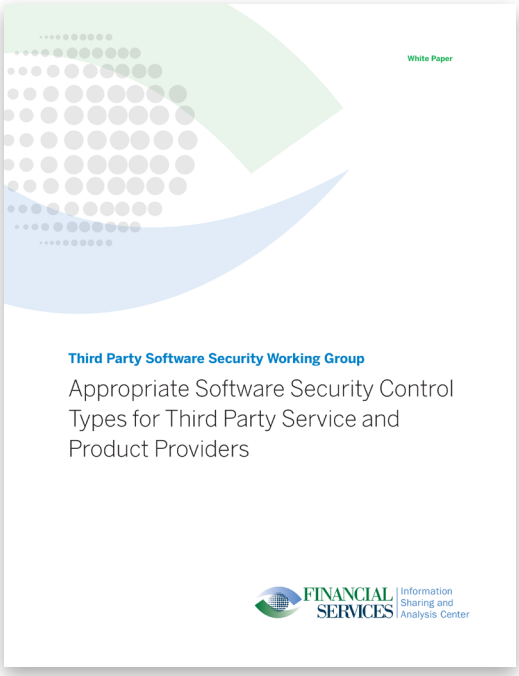
Secure Governance Elements

- **a.** Does the supplier include its software security policy in its Organization-level policy?
- **b.** Are the supplier's software developers required to adhere to the software security policy?
- **c.** Do the supplier's Organization-level and software security policies include requirements that produce a high-quality, repeatable result?
- **d.** Does the supplier's Organization-level policy require its secure development process for all applications and patch releases?
- **e.** Do the supplier's software applications have quality metrics (e.g., security control verification) tied to its software security policy or secure coding standards?
- **f.** Do the supplier's quality metrics produce a high-quality, repeatable result?
- **g.** Does the supplier's organization chart include roles that align to the Organization-level and software security policies, procedures, and reporting standards?
- **h.** Are the supplier's major software version releases validated for compliance to the Organization-level policy?
- **i.** Does the supplier conduct an internal verification activity on all pre-release software to ensure that all software releases are free from significant security defects, unless appropriate justification is recorded by application owner?
- **j.** Do the supplier's verification activity results drive changes in the implementation of secure architecture and coding standards?
- **k.** Do the supplier's verification activity results drive changes in the implementation of the secure development process?
- **l.** Are the supplier's development teams audited for compliance to security policy routinely (e.g., per release, yearly)?
- **m.** Does the supplier provide its developers with ongoing internal training opportunities?
- **n.** Does the supplier provide its developers with ongoing external training opportunities?

Appendix B: Sample Questionnaire for a Process-Based Assessment: FS-ISAC

The white paper from the Financial Services Information Sharing and Analysis Center (FS-ISAC) Third Party Software Security Working Group, *"Appropriate Software Security Control Types for Third Party Service and Product Providers,"* contains a questionnaire for a process-based assessment (starting on page 22 of the paper).

The FS-ISAC paper is freely available at:
http://docs.ismgcorp.com/files/external/WP_FSISAC_Third_Party_Software_Security_Working_Group.pdf



Key Contributors

The authors would like to thank the following individuals for their guidance and mentorship in the development of this paper and in the formation of the ideas it puts forth:

Eric Baize, Senior Director, Product Security and Trusted Engineering for EMC Corporation

Steve Lipner, Partner Director of Software Security, Microsoft Corporation

Acknowledgements

Vishal Asthana, Security Compass
Nadya Bartol, Utilities Telecom Council
Edward Bonver, Symantec Corporation
David Doughty, Intel Corporation
Gerold Huebner, SAP AG
David Lenoe, Adobe
Anders Magnusson, CA Technologies
John Martin, The Boeing Company
Frances Paulisch, Siemens AG
Glenn Pittaway, Microsoft Corporation
Jim Routh, Aetna
Anne Nielsen, Veracode
Wendy Poland, Adobe
Chris Wysopal, Veracode

About SAFECode

The Software Assurance Forum for Excellence in Code (SAFECode) is a non-profit organization exclusively dedicated to increasing trust in information and communications technology products and services through the advancement of effective software assurance methods. 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. For more information, visit www.safecode.org.

Software Assurance Forum for Excellence in Code
401 Edgewater Place, Suite 600
Wakefield, MA 01880
(p) +1 781-876-8833
(f) +1 781-224-1239
info@safecode.org

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