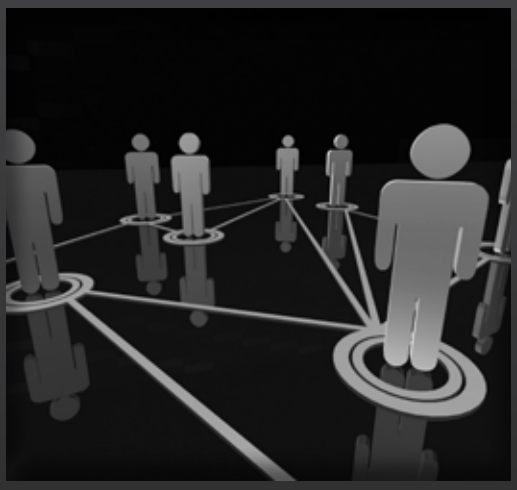


Software Assurance Maturity Model

A guide to building security into software development

Version 1.5

For the latest version and additional info, please see the project web site at

https://www.owasp.org/index.php/OWASP_SAMM_Project

Acknowledgements

This document was originally created through the OpenSAMM Project led by Pravir Chandra (chandra@owasp.org), an independent software security consultant. Creation of the first draft was made possible through funding from Fortify Software, Inc. Since the initial release of SAMM, this project has become part of the Open Web Application Security Project (OWASP). This document is currently maintained and updated through the OWASP SAMM Project led by Sebastien Deleersnyder, Bart De Win & Brian Glas. Thanks also go to many supporting organizations that are listed on back cover.

Contributors & Reviewers

This work would not be possible without the support of many individual reviewers and experts that offered contributions and critical feedback.

- Fabio Arciniegas
- Matt Bartoldus
- Jonathan Carter
- Darren Challey
- Brian Chess
- Justin Clarke
- Dan Cornell
- Michael Craigue
- Dinis Cruz
- Sebastien Deleersnyder
- Justin Derry
- Bart De Win
- John Dickson
- Alexios Fakos
- David Fern
- Brian Glas
- Kuai Hinojosa
- Jerry Hoff
- Carsten Huth
- Bruce Jenkins
- Daniel Kefer
- Yan Kravchenko
- James McGovern
- Matteo Meucci
- Jeff Payne
- Gunnar Peterson
- Jeff Piper
- Andy Steingruebl
- John Steven
- Chad Thunberg
- Colin Watson
- Jeff Williams
- Steven Wierckx

This is an OWASP Project

OWASP

The Open Web Application Security Project

OWASP is an international organization and the OWASP Foundation supports OWASP efforts around the world. OWASP is an open community dedicated to enabling organizations to conceive, develop, acquire, operate, and maintain applications that can be trusted. All of the OWASP tools, documents, forums, and chapters are free and open to anyone interested in improving application security. We advocate approaching application security as a people, process, and technology problem because the most effective approaches to application security include improvements in all of these areas. We can be found at https://www.owasp.org.

Executive Summary

The Software Assurance Maturity Model (SAMM) is an open framework to help organizations formulate and implement a strategy for software security that is tailored to the specific risks facing the organization. The resources provided by SAMM will aid in:

- *Evaluating an organization's existing software security practices.*
- *Building a balanced software security assurance program in well-defined iterations.*
- *Demonstrating concrete improvements to a security assurance program.*
- *Defining and measuring security-related activities throughout an organization.*

Version 1.1 of SAMM expanded and restructured its predecessor into four complementary resources: this document that describes the core SAMM model, the How-To Guide that explains how to apply the model, the Quick Start Guide to help accelerate learning and adoption, and the toolbox that provides simple automation for data collection, metrics, and graphs. Furthermore, a number of elements have been renamed to better represent their purpose.

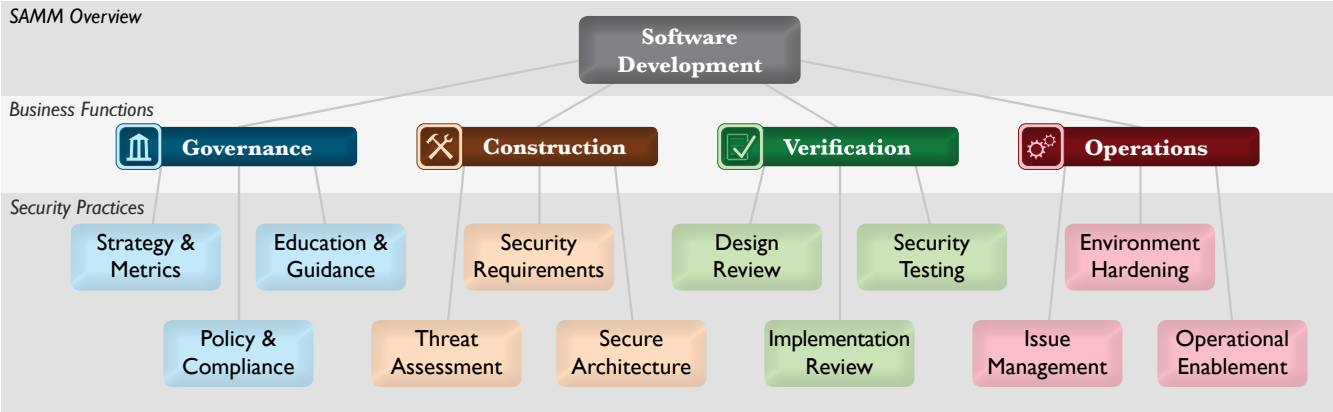
Version 1.5 of SAMM incorporates a refinement of the scoring model to provide more granularity to the scoring in an assessment. Now an organization will get credit for all the related work done in a practice rather than having the base number held at the highest completed maturity level. The updated scoring model has been designed to help SAMM assessors and organizations avoid the awkward discussion on whether to mark an answer yes or no when it is honestly something in between, and to show incremental improvements.

SAMM was defined with flexibility in mind such that it can be utilized by small, medium, and large organizations using any style of development. Additionally, this model can be applied organization-wide, for a single line-of-business, or even for an individual project. Beyond these traits, SAMM was built on the following principles:

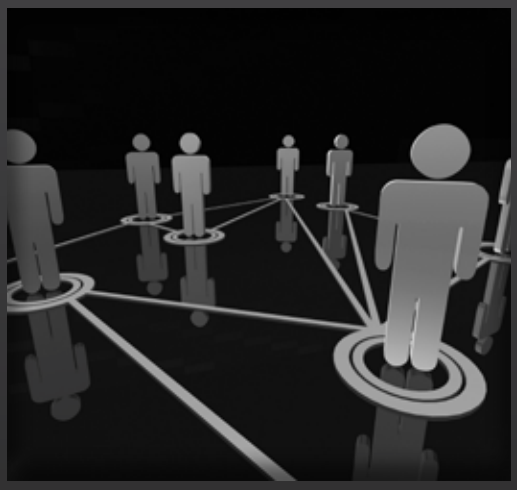
- *An organization's behavior changes slowly over time* - A successful software security program should be specified in small iterations that deliver tangible assurance gains while incrementally working toward long-term goals.
- *There is no single recipe that works for all organizations* - A software security framework must be flexible and allow organizations to tailor their choices based on their risk tolerance and the way in which they build and use software.
- *Guidance related to security activities must be prescriptive* - All the steps in building and assessing an assurance program should be simple, well-defined, and measurable. This model also provides roadmap templates for common types of organizations.

The foundation of the model is built upon the core business functions of software development with security practices tied to each (see diagram below). The building blocks of the model are the three maturity levels defined for each of the twelve security practices. These define a wide variety of activities in which an organization could engage to reduce security risks and increase software assurance. Additional details are included to measure successful activity performance, understand the associated assurance benefits, estimate personnel and other costs.

As an open project, SAMM content shall always remain vendor-neutral and freely available for all to use.



Contents



Understanding the Model

A view of the big picture

SAMM is built upon a collection of security practices that are tied back into the core business functions involved in software development. This section introduces those business functions and the corresponding security practices for each. After covering the high-level framework, the maturity levels for each security practice are also discussed briefly in order to paint a picture of how each can be iteratively improved over time.

Business Functions

***At the highest level, SAMM defines four critical* business functions.** Each business function is a category of activities related to the nuts-and-bolts of software development, or stated another way, any organization involved with software development must fulfill each of these business functions to some degree.

***For each business function, SAMM defines three* security practices.** Each security practice is an area of security-related activities that build assurance for the related business function. There are twelve security practices that are the independent silos for improvement that map to the four business functions of software development.

***For each security practice, SAMM defines three* maturity levels *as* objectives**. Each level within a security practice is characterized by a successively more sophisticated objective defined by specific activities, and more stringent success metrics than the previous level. Additionally, each security practice can be improved independently, though related activities can lead to optimizations.

Governance

Governance is centered on the processes and activities related to how an organization manages overall software development activities. More specifically, this includes concerns that impact cross-functional groups involved in development, as well as business processes that are established at the organization level.

Strategy & Metrics involves the overall strategic direction of the software assurance program and instrumentation of processes and activities to collect metrics about an organization's security posture.

Policy & Compliance involves setting up a security, compliance, and audit control framework throughout an organization to achieve increased assurance in software under contruction and in operation.

Education & Guidance involves increasing security knowledge amongst personnel in software development through training and guidance on security topics relevant to individual job functions.

...more on page 10

Construction

Construction concerns the processes and activities related to how an organization defines goals and creates software within development projects. In general, this will include product management, requirements gathering, high-level architecture specification, detailed design, and implementation.

Threat Assessment involves accurately identifying and characterizing potential attacks upon an organization's software in order to better understand the risks and facilitate risk management.

Security Requirements involves promoting the inclusion of security-related requirements during the software development process in order to specify correct functional-ity from inception.

Secure Architecture involves bolstering the design process with activities to promote secure-by-default designs and control over technologies and frameworks upon which software is built.

...more on page 12

Verification

Verification is focused on the processes and activities related to how an organization checks, and tests artifacts produced throughout software development. This typically includes quality assurance work such as testing, but it can also include other review and evaluation activities.

Design Review involves inspection of the artifacts created from the design process to ensure provision of adequate security mechanisms, and adherence to an organization's expectations for security.

Implementation Review involves assessment of an organization's source code to aid vulnerability discovery and related mitigation activities as well as establish a baseline for secure coding expectations.

Security Testing involves testing the organization's software in its runtime environment, in order to both discover vulnerabilities, and establish a minimum standard for software releases.

...more on page 14

Operations

Operations entails the processes and activities related to how an organization manages software releases that has been created. This can involve shipping products to end users, deploying products to internal or external hosts, and normal operations of software in the runtime environment.

Issue Management involves establishing consistent processes for managing internal and external vulnerability reports to limit exposure and gather data to enhance the security assurance program.

Environment Hardening involves implementing controls for the operating environment surrounding an organization's software to bolster the security posture of applications that have been deployed.

Operational Enablement involves identifying and capturing security-relevant information needed by an operator to properly configure, deploy, and run an organization's software.

...more on page 16

Maturity Levels

Each of the twelve security practices has three defined maturity levels and an implicit starting point at zero. The details for each level differs between the practices, but they generally represent:

0 Implicit starting point representing the activities in the practice being unfulfilled

1 Initial understanding and adhoc provision of security practice

2 Increase efficiency and/or effectiveness of the security practice

3 Comprehensive mastery of the security practice at scale

Notation

Throughout this document, the following terms will be reserved words that refer to the SAMM components defined in this section:

- ✦Business Function
- ✦Security Practice
- ✦Maturity Level or Objective

Assurance programs might not always consist of activities that neatly fall on a boundary between maturity levels, e.g. an organization that assesses to a Level 1 for a given practice might also have additional activities in place but not such that Level 2 is completed. Prior to v1.5, the organization's score should be annotated with a "**+**" symbol to indicate there's additional assurances in place beyond those indicated by the Level obtained. For example, an organization that is performing all Level 1 activities for operational enablement as well as one Level 2 or 3 activity would be assigned a "**1+**" score. Likewise, an organization performing all activities for a security practice, including some beyond the scope of SAMM, would be given a "**3+**" score.

The scoring model has changed in v1.5 to provide more granularity to the scoring in an assessment. Now an organization will get credit for different levels of work they have done within a practice rather than having the base number held at the highest completed maturity level. The scoring is now fractional to two decimal places for each practice and a single decimal for an answer. Questions have also been changed from Yes/No to four options that represent different levels of coverage or maturity. This change will assist practitioners completing SAMM assessments with the inevitable debate whether to mark an answer yes or no when it is honestly something in between.

The primary reason for the scoring change was to ensure organizations would receive full credit for their work in software security and to make it easier to show improvements in scoring when activities and programs grow and mature. The hope is this change will bring us closer to understanding what works in different scenarios for different organizations to benefit all.

The toolbox spreadsheet has been updated to reflect more context aware answers for each of the questions in the assessment. The formulas in the toolbox will also average the answers to calculate the score for each practice, a roll up average for each business function, and an overall score. The toolbox also has updated scorecard graphics that help represent the current score and can help show improvements to the program as the answers to the questions change. The worksheets later in this document are also updated to align with the new scoring model.

No = 0 **Few/Some = .2** **At Least Half = .5** **Many/Most = 1**

Governance

Description of Security Practices

Strategy & Metrics

The Strategy & Metrics (SM) practice is focused on establishing the framework within an organization for a software security assurance program. This is the most fundamental step in defining security goals in a way that's both measurable and aligned with the organization's real business risk.

By starting with lightweight risk profiles, an organization grows into more advanced risk classification schemes for application and data assets over time. With additional insight on relative risk measures, an organization can tune its project-level security goals and develop granular roadmaps to make the security program more efficient. At the more advanced levels within this practice, an organization draws upon many data sources, both internal and external, to collect metrics and qualitative feedback on the security program. This allows fine tuning of cost outlay versus the realized benefit at the program level.

Policy & Compliance

The Policy & Compliance (PC) practice is focused on understanding and meeting external legal and regulatory requirements while also driving internal security standards to ensure compliance in a way that's aligned with the business purpose of the organization.

A driving theme for improvement within this practice is focus on project-level audits that gather information about the organization's behavior in order to check that expectations are being met. By introducing routine audits that start out lightweight and grow in depth over time, organizational change is achieved iteratively.

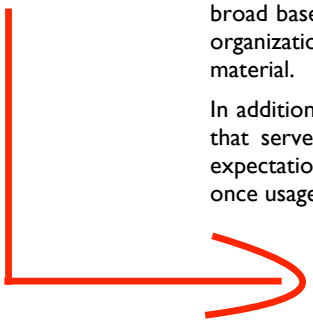
In a sophisticated form, provision of this practice entails organization-wide understanding of both internal standards and external compliance drivers while also maintaining low-latency checkpoints with project teams to ensure no project is operating outside expectations without visibility.

Education & Guidance

The Education & Guidance (EG) practice is focused on arming personnel involved in the software lifecycle with knowledge and resources to design, develop, and deploy secure software. With improved access to information, project teams will be better able to proactively identify and mitigate specific security risks that apply to their organization.

One major theme for improvement across the objectives is providing training for employees, either through instructor-led sessions or computer-based modules. As an organization progresses, a broad base of training is built by starting with developers and moving to other roles throughout the organization, culminating with the addition of role-based certification to ensure comprehension of the material.

In addition to training, this practice also requires pulling security-relevant information into guidelines that serve as reference information to staff. This builds a foundation for establishing a baseline expectation for security practices in your organization, and later allows for incremental improvement once usage of the guidelines has been adopted.



Governance

Activities overview

Strategy & Metrics

...more on page 24

	SM 1	SM 2	SM 3
Objective	**Establish a unified strategic roadmap for software security within the organization.**	**Measure relative value of data and software assets and choose risk tolerance.**	**Align security expenditure with relevant business indicators and asset value.**
Activities	A. Estimate overall business risk profile B. Build and maintain assurance program roadmap	A. Classify data and applications based on business risk B. Establish and measure per-classification security goals	A. Conduct periodic industry-wide cost comparisons B. Collect metrics for historic security spend

Policy & Compliance

...more on page 28

	PC 1	PC 2	PC 3
Objective	**Understand relevant governance and compliance drivers to the organization.**	**Establish security and compliance baseline and understand per-project risks.**	**Require compliance and measure projects against organization-wide policies and standards.**
Activities	A. Identify and monitor external compliance drivers B. Build and maintain compliance guidelines	A. Build policies and standards for security and compliance B. Establish project audit practice	A. Create compliance gates for projects B. Adopt solution for audit data collection

Education & Guidance

...more on page 32

	EG 1	EG 2	EG 3
Objective	**Offer development staff access to resources around the topics of secure programming and deployment.**	**Educate all personnel in the software lifecycle with role-specific guidance on secure development.**	**Mandate comprehensive security training and certify personnel for baseline knowledge.**
Activities	A. Conduct technical security awareness training B. Build and maintain technical guidelines	A. Conduct role-specific application security training B. Utilize security coaches to enhance project teams	A. Create formal application security support portal B. Establish role-based examination/certification

Culture

Construction

Description of Security Practices

Threat Assessment

The Threat Assessment (TA) practice is centered on identification and understanding the project-level risks based on the functionality of the software being developed and characteristics of the runtime environment. From details about threats and likely attacks against each project, the organization as a whole operates more effectively through better decisions about prioritization of initiatives for security. Additionally, decisions for risk acceptance are more informed, therefore better aligned to the business.

By starting with simple threat models and building to more detailed methods of threat analysis and weighting, an organization improves over time. Ultimately, a sophisticated organization would maintain this information in a way that is tightly coupled to the compensating factors and pass-through risks from external entities. This provides greater breadth of understanding for potential downstream impacts from security issues while keeping a close watch on the organization's current performance against known threats.

Security Requirements

The Security Requirements (SR) practice is focused on proactively specifying the expected behavior of software with respect to security. Through addition of analysis activities at the project level, security requirements are initially gathered based on the high-level business purpose of the software. As an orga-nization advances, more advanced techniques are used such as access control specifications to discover new security requirements that may not have been initially obvious to development.

In a sophisticated form, provision of this practice also entails pushing the security requirements of the organization into its relationships with suppliers and then auditing projects to ensure all are adhering to expectations with regard to specification of security requirements.

Secure Architecture

The Secure Architecture (SA) practice is focused on proactive steps for an organization to design and build secure software by default. By enhancing the software design process with reusable services and components, the overall security risk from software development can be dramatically reduced.

Beginning from simple recommendations about software frameworks and explicit consideration of secure design principles, an organization evolves toward consistently using design patterns for security functionality. Also, activities encourage project teams to increased utilization of centralized security services and infrastructure.

As an organization evolves over time, sophisticated provision of this practice entails organizations building reference platforms to cover the generic types of software they build. These serve as frameworks upon which developers can build custom software with lower risk of vulnerabilities.

Design

Construction

Activities overview

Threat Assessment

...more on page 36

	TA 1	TA 2	TA 3
Objective	**Identify and understand high-level threats to the organization and individual projects.**	**Increase accuracy of threat assessment and improve granularity of per-project understanding.**	**Concretely align compensating controls to each threat against internal and third-party software.**
Activities	A. Build and maintain application-specific threat models B. Develop attacker profile from software architecture	A. Build and maintain abuse-case models per project B. Adopt a weighting system for measurement of threats	A. Explicitly evaluate risk from third-party components B. Elaborate threat models with compensating controls

Security Requirements

...more on page 40

	SR 1	SR 2	SR 3
Objective	**Consider security explicitly during the software requirements process.**	**Increase granularity of security requirements derived from business logic and known risks.**	**Mandate security requirements process for all software projects and third-party dependencies.**
Activities	A. Derive security requirements from business functionality B. Evaluate security and compliance guidance for requirements	A. Build an access control matrix for resources and capabilities B. Specify security requirements based on known risks	A. Build security requirements into supplier agreements B. Expand audit program for security requirements

Secure Architecture

...more on page 44

	SA 1	SA 2	SA 3
Objective	**Insert consideration of proactive security guidance into the software design process.**	**Direct the software design process toward known-secure services and secure-by-default designs.**	**Formally control the software design process and validate utilization of secure components.**
Activities	A. Maintain list of recommended software frameworks B. Explicitly apply security principles to design	A. Identify and promote security services and infrastructure B. Identify security design patterns from architecture	A. Establish formal reference architectures and platforms B. Validate usage of frameworks, patterns, and platforms

Verification

Description of Security Practices

Design Review

The Design Review (DR) practice is focused on assessment of software design and architecture for security-related problems. This allows an organization to detect architecture-level issues early in software development and thereby avoid potentially large costs from refactoring later due to security concerns.

Beginning with lightweight activities to build understanding of the security-relevant details about an architecture, an organization evolves toward more formal inspection methods that verify completeness in provision of security mechanisms. At the organization level, design review services are built and offered to stakeholders.

In a sophisticated form, provision of this practice involves detailed, data-level inspection of designs, and enforcement of baseline expectations for conducting design assessments and reviewing findings before releases are accepted.

Implementation Review

The Implementation Review (IR) practice is focused on inspection of software at the source code and configuration level in order to find security vulnerabilities. Code-level vulnerabilities are generally simple to understand conceptually, but even informed developers can easily make mistakes that leave software open to potential compromise.

To begin, an organization uses lightweight checklists and for efficiency, only inspects the most critical software modules. However, as an organization evolves it uses automation technology to dramatically improve coverage and efficacy of implementation review activities.

Sophisticated provision of this practice involves deeper integration of implementation review into the development process to enable project teams to find problems earlier. This also enables organizations to better audit and set expectations for implementation review findings before releases can be made.

Security Testing

The Security Testing (ST) practice is focused on inspection of software in the runtime environment in order to find security problems. These testing activities bolster the assurance case for software by checking it in the same context in which it is expected to run, thus making visible operational misconfigurations or errors in business logic that are difficult to otherwise find.

Starting with penetration testing and high-level test cases based on the functionality of software, an organization evolves toward usage of security testing automation to cover the wide variety of test cases that might demonstrate a vulnerability in the system.

In an advanced form, provision of this practice involves customization of testing automation to build a battery of security tests covering application-specific concerns in detail. With additional visibility at the organization level, security testing enables organizations to set minimum expectations for security testing results before a project release is accepted.

Verification

Activities overview

Design Review

...more on page 48

	DR 1	DR 2	DR 3
Objective	**Support ad-hoc reviews of software design to ensure baseline mitigations for known risks.**	**Offer assessment services to review software design against comprehensive best practices for security.**	**Require assessments and validate artifacts to develop detailed understanding of protection mechanisms.**
Activities	A. Identify software attack surface B. Analyze design against known security requirements	A. Inspect for complete provision of security mechanisms B. Deploy design review service for project teams	A. Develop data-flow diagrams for sensitive resources B. Establish release gates for design review

Implementation Review

...more on page 52

	IR 1	IR 2	IR 3
Objective	**Opportunistically find basic code-level vulnerabilities and other high-risk security issues.**	**Make implementation review during development more accurate and efficient through automation.**	**Mandate comprehensive implementation review process to discover language-level and application-specific risks.**
Activities	A. Create review checklists from known security requirements B. Perform point-review of high-risk code	A. Utilize automated code analysis tools B. Integrate code analysis into development process	A. Customize code analysis for application-specific concerns B. Establish release gates for code review

Security Testing

...more on page 56

	ST 1	ST 2	ST 3
Objective	**Establish process to perform basic security tests based on implementation and software requirements.**	**Make security testing during development more complete and efficient through automation.**	**Require application-specific security testing to ensure baseline security before deployment.**
Activities	A. Derive test cases from known security requirements B. Conduct penetration testing on software releases	A. Utilize automated security testing tools B. Integrate security testing into development process	A. Employ application-specific security testing automation B. Establish release gates for security testing

Testing

Operations

Description of Security Practices

Issue Management

The Issue Management (IM) practice is focused on the processes within an organization with respect to handling issue reports and operational incidents. By having these processes in place, an organization's projects will have consistent expectations and increased efficiency for handling these events, rather than chaotic and uninformed responses.

Starting from lightweight assignment of roles in the event of an incident, an organization grows into a more formal incident response process that ensures visibility and tracking on issues that occur. Communications are also improved to improve overall understanding of the processes.

In an advanced form, issue management involves thorough dissecting of incidents and issue reports to collect detailed metrics and other root-cause information to feedback into the organization's downstream behavior.

Environment Hardening

The Environment Hardening (EH) practice is focused on building assurance for the runtime environment that hosts the organization's software. Since secure operation of an application can be deteriorated by problems in external components, hardening this underlying infrastructure directly improves the overall security posture of the software.

By starting with simple tracking and distributing of information about the operating environment to keep development teams better informed, an organization evolves to scalable methods for managing deployment of security patches and instrumenting the operating environment with early-warning detectors for potential security issues before damage is done.

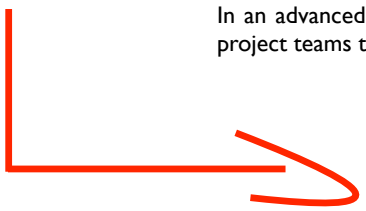
As an organization advances, the operating environment is further reviewed and hardened by deployment of protection tools to add layers of defenses and safety nets to limit damage in case any vulnerabilities are exploited.

Operational Enablement

The Operational Enablement (OE) practice is focused on gathering security critical information from the project teams building software and communicating it to the users and operators of the software. Without this information, even the most securely designed software carries undue risks since important security characteristics and choices will not be known at a deployment site.

Starting from lightweight documentation to capture the most important details for users and operators, an organization evolves toward building complete operational security guides that are delivered with each release.

In an advanced form, operational enablement also entails organization-level checks against individual project teams to ensure that information is being captured and shared according to expectations.



Operations

Activities overview

Issue Management

...more on page 60

	IM 1	IM 2	IM 3
Objective	**Understand high-level plan for responding to issue reports or incidents.**	**Elaborate expectations for response process to improve consistency and communications.**	**Improve analysis and data gathering within response process for feedback into proactive planning.**
Activities	A. Identify point of contact for security issues B. Create informal security response team(s)	A. Establish consistent issue reponse process B. Adopt a security issue disclosure process	A. Conduct root cause analysis for for issues B. Collect per-issue metrics

Environment Hardening

...more on page 64

	EH 1	EH 2	EH 3
Objective	**Understand baseline operational environment for applications and software components.**	**Improve confidence in application operations by hardening the operating environment.**	**Validate application health and status of operational environment against known best practices.**
Activities	A. Maintain operational environment specification B. Identify and install critical security upgrades and patches	A. Establish routine patch management process B. Monitor baseline environment configuration status	A. Identify and deploy relevant operations protection tools B. Expand audit program for environment configuration

Operational Enablement

...more on page 68

	OE 1	OE 2	OE 3
Objective	**Enable communications between development teams and operators for critical security-relevant data.**	**Improve expectations for continuous secure operations through provision of detailed procedures.**	**Mandate communication of security information and validate artifacts for completeness.**
Activities	A. Capture critical security information for deployment B. Document procedures for typical application alerts	A. Create per-release change management procedures B. Maintain formal operational security guides	A. Expand audit program for operational information B. Perform code signing for application components

Monitoring

Governance

Assessment worksheet

Strategy & Metrics

	Score	0.0	0.2	0.5	1.0	
✦	Is there a software security assurance program in place?	No	<1 yr	>1 yr	Mature	
✦	Are development staff aware of future plans for the assurance program?	No	Some	Half	Most	
✦	Do the business stakeholders understand your organization's risk profile?	No	Some	Half	Most	SM 1
✦	Are many of your applications and resources categorized by risk?	No	Some	Half	Most	
✦	Are risk ratings used to tailor the required assurance activities?	No	Some	Half	Most	
✦	Does the organization know about what's required based on risk ratings?	No	Some	Half	Most	SM 2
✦	Is per-project data for the cost of assurance activities collected?	No	Some	Half	Most	
✦	Does your organization regularly compare your security spend with that of other organizations?	No	Once	Every 2-3 yrs	Annually	SM 3

Policy & Compliance

	Score	0.0	0.2	0.5	1.0	
✦	Do project stakeholders know their project's compliance status?	No	Some	Half	Most	
✦	Are compliance requirements specifically considered by project teams?	No	Not Apply	Ad-hoc	Yes	
✦	Does the organization utilize a set of policies and standards to control software development?	No	Per Team	Org Wide	Integrated Process	PC 1
✦	Are project teams able to request an audit for compliance with policies and standards?	No	Some	Half	Most	
✦	Are projects periodically audited to ensure a baseline of compliance with policies and standards?	No	Some	Half	Most	PC 2
✦	Does the organization systematically use audits to collect and control compliance evidence?	No	Bus Area	Org Wide	Org Wide & Required	PC 3

Education & Guidance

	Score	0.0	0.2	0.5	1.0	
✦	Have developers been given high-level security awareness training?	No	Once	2-3 yrs	Annually	
✦	Does each project team understand where to find secure development best-practices and guidance?	No	Some	Half	Most	
✦	Are those involved in the development process given role-specific security training and guidance?	No	Some	Half	Most	EG 1
✦	Are stakeholders able to pull in security coaches for use on projects?	No	Some	Half	Most	
✦	Is security-related guidance centrally controlled and consistently distributed throughout the organization?	No	Per Team	Org Wide	Integrated Process	EG 2
✦	Are developers tested to ensure a baseline skill-set for secure development practices?	No	Once	Every 2-3 yrs	Annually	EG 3

CULTURE

Construction

Assessment worksheet

Threat Assessment

	Score	0.0	0.2	0.5	1.0	
✦ Do projects in your organization consider and document likely threats?		No	Some	Half	Most	
✦ Does your organization understand and document the types of attackers it faces?		No	Some	Half	Most	TA 1
✦ Do project teams regularly analyze functional requirements for likely abuses?		No	Some	Half	Most	
✦ Do project teams use a method of rating threats for relative comparison?		No	Some	Half	Most	
✦ Are stakeholders aware of relevant threats and ratings?		No	Some	Half	Most	TA 2
✦ Do project teams specifically consider risk from external software?		No	Some	Half	Most	
✦ Are the majority of the protection mechanisms and controls captured and mapped back to threats?		No	Some	Half	Most	TA 3

Security Requirements

	Score	0.0	0.2	0.5	1.0	
✦ Do project teams specify security requirements during development?		No	Some	Half	Most	
✦ Do project teams pull requirements from best practices and compliance guidance?		No	Per Team	Org Wide	Integrated Process	SR 1
✦ Do stakeholders review access control matrices for relevant projects?		No	Some	Half	Most	
✦ Do project teams specify requirements based on feedback from other security activities?		No	Some	Half	Most	SR 2
✦ Do stakeholders review vendor agreements for security requirements?		No	Some	Half	Most	
✦ Are audits performed against the security requirements specified by project teams?		No	Once	Every 2-3 yrs	Annually	SR 3

Secure Architecture

	Score	0.0	0.2	0.5	1.0	
✦ Are project teams provided with a list of recommended third-party components?		No	Per Team	Org Wide	Integrated Process	
✦ Are project teams aware of secure design principles and do they apply them consistently?		No	Some	Half	Most	SA 1
✦ Do you advertise shared security services with guidance for project teams?		No	Bus Area	Org Wide	Org Wide & Required	
✦ Are project teams provided with prescriptive design patterns based on their application architecture?		No	Per Team	Org Wide	Integrated Process	SA 2
✦ Do project teams build software from centrally-controlled platforms and frameworks?		No	Some	Half	Most	
✦ Are project teams audited for the use of secure architecture components?		No	Once	Every 2-3 yrs	Annually	SA 3

Design

Verification

Assessment worksheet

Design Review

	Score	0.0	0.2	0.5	1.0	
✦ Do project teams document the attack perimeter of software designs?		No	Some	Half	Most	
✦ Do project teams check software designs against known security risks?		No	Some	Half	Most	DR 1
✦ Do project teams specifically analyze design elements for security mechanisms?		No	Some	Half	Most	
✦ Are project stakeholders aware of how to obtain a formal secure design review?		No	Some	Half	Most	DR 2
✦ Does the secure design review process incorporate detailed data-level analysis?		No	Some	Half	Most	
✦ Does a minimum security baseline exist for secure design review results?		No	Per Team	Org Wide	Integrated Process	DR 3

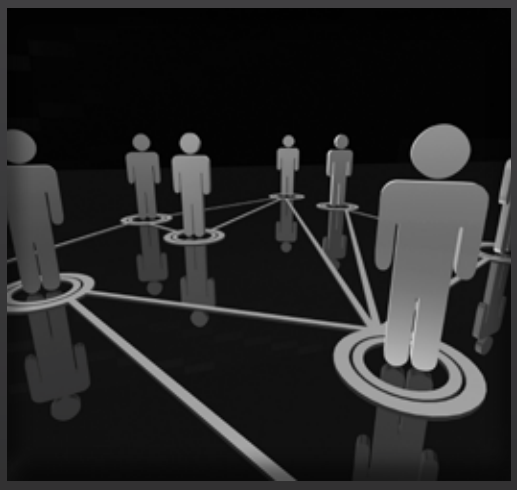
Implementation Review

	Score	0.0	0.2	0.5	1.0	
✦ Do project teams have review checklists based on common security related problems?		No	Bus Area	Org Wide	Org Wide & Required	
✦ Do project teams review selected high-risk code?		No	Some	Half	Most	IR 1
✦ Can project teams access automated code analysis tools to find security problems?		No	Per Team	Org Wide	Integrated Process	
✦ Do stakeholders consistently review results from code reviews?		No	Some	Half	Most	IR 2
✦ Do project teams utilize automation to check code against application-specific coding standards?		No	Bus Area	Org Wide	Org Wide & Required	
✦ Does a minimum security baseline exist for code review results?		No	Per Team	Org Wide	Integrated Process	IR 3

Security Testing

	Score	0.0	0.2	0.5	1.0	
✦ Do projects specify security testing based on defined security requirements?		No	Some	Half	Most	
✦ Is penetration testing performed on high risk projects prior to release?		No	Some	Half	Most	
✦ Are stakeholders aware of the security test status prior to release?		No	Some	Half	Most	ST 1
✦ Do projects use automation to evaluate security test cases?		No	Some	Half	Most	
✦ Do projects follow a consistent process to evaluate and report on security tests to stakeholders?		No	Some	Half	Most	ST 2
✦ Are security test cases comprehensively generated for application-specific logic?		No	Some	Half	Most	
✦ Does a minimum security baseline exist for security testing?		No	Per Team	Org Wide	Integrated Process	ST 3

Testing



The Security Practices

An explanation of the details