

Software Assurance Maturity Model

Tampa OWASP, May 20, 2009.



OWASP

The Open Web Application Security Project

Introduction

Who am I?

What are we doing here?

What am I talking about ?



Open SAMM – Executive Brief

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:

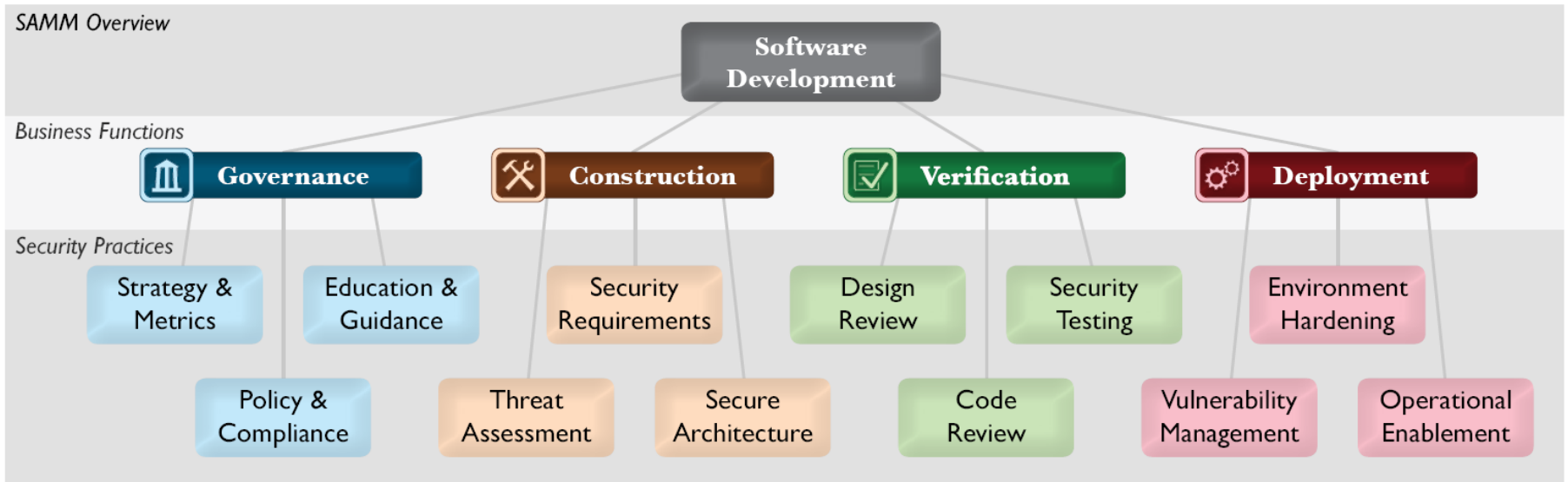
Evaluating an organization's existing software security practices

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

SAMM Overview





Governance

Strategy & Metrics



SM 1



SM 2



SM 3

OBJECTIVE

Establish 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

The Strategy & Metrics (SM) Practice is focused on establishing the framework within an organization for a software security assurance program.



Governance

Policy & Compliance



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

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.



Governance

Education & Guidance



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 life-cycle 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

The Education & Guidance (EG) Practice is focused on arming personnel involved in the software life-cycle with knowledge and resources to design, develop, and deploy secure software.



Construction

Threat Assessment



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 tie 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

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.



Construction

Security Requirements



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

The Security Requirements (SR) Practice is focused on proactively specifying the expected behavior of software with respect to security.



Construction

Secure Architecture



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

The Secure Architecture (SA) Practice is focused on proactive steps for an organization to design and build secure software by default.



Verification

Design Review



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

The Design Review (DR) Practice is focused on assessment of software design and architecture for security-related problems.



Verification

Code Review



OBJECTIVE

Opportunistically find basic code-level vulnerabilities and other high-risk security issues

Make code review during development more accurate and efficient through automation

Mandate comprehensive code 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

The Code Review (CR) Practice is focused on inspection of software at the source code level in order to find security vulnerabilities.



Verification

Security Testing



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

The Security Testing (ST) Practice is focused on inspection of software in the runtime environment in order to find security problems.



Deployment

Vulnerability Management



VM 1



VM 2



VM 3

OBJECTIVE

Understand high-level plan for responding to vulnerability 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 incident response process
- B. Adopt a security issue disclosure process

- A. Conduct root cause analysis for incidents
- B. Collect per-incident metrics

The Vulnerability Management (VM) Practice is focused on the processes within an organization with respect to handling vulnerability reports and operational incidents.



Deployment

Environment Hardening



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

The Environment Hardening (EH) Practice is focused on building assurance for the runtime environment that hosts the organization's software.



Deployment

Operational Enablement



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

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.

So how do we use this?

Conduct Assessments

- Lightweight
- Detailed

Create Score Cards

- Gap Analysis
- Show Improvement
- Current Progress

Building Your Program

- Set Metrics
- Roadmap

Assessment Worksheets

Example shows just Governance

Similar to other self audits for things such as PCI




Very basic, doesn't involve testing

Governance

Assessment worksheet




Strategy & Metrics

Yes/No

✦ Is there a software security assurance program already in place?		
✦ Do most of the business stakeholders understand your organization's risk profile?		
✦ Is most of your development staff aware of future plans for the assurance program?		 SM 1
✦ Are most of your applications and resources categorized by risk?		
✦ Are risk ratings used to tailor the required assurance activities?		
✦ Does most of the organization know about what's required based on risk ratings?		 SM 2
✦ Is per-project data for cost of assurance activities collected?		
✦ Does your organization regularly compare your security spend with other organizations?		 SM 3




Policy & Compliance

Yes/No

✦ Do most project stakeholders know their project's compliance status?		
✦ Are compliance requirements specifically considered by project teams?		 PC 1
✦ Does the organization utilize a set of policies and standards to control software development?		
✦ Are project teams able to request an audit for compliance with policies and standards?		 PC 2
✦ Are projects periodically audited to ensure a baseline of compliance with policies and standards?		
✦ Does the organization systematically use audits to collect and control compliance evidence?		 PC 3

Education & Guidance

Yes/No

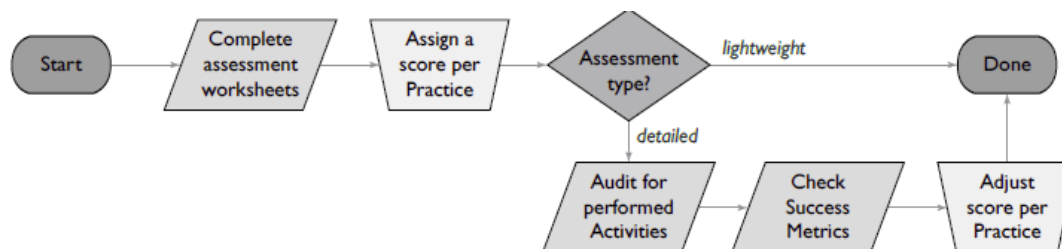
✦ Have most developers been given high-level security awareness training?		
✦ Does each project team have access to secure development best practices and guidance?		 EG 1
✦ Are most roles in the development process given role-specific training and guidance?		
✦ Are most stakeholders able to pull in security coaches for use on projects?		 EG 2
✦ Is security-related guidance centrally controlled and consistently distributed throughout the organization?		
✦ Are most people tested to ensure a baseline skill-set for secure development practices?		 EG 3

Lightweight Assessment

The assessment worksheets for each Practice are evaluated and scores are assigned based on answers.

This type of assessment is usually sufficient for an organization that is trying to map their existing assurance program into SAMM and just wants to get a quick picture of where they stand.

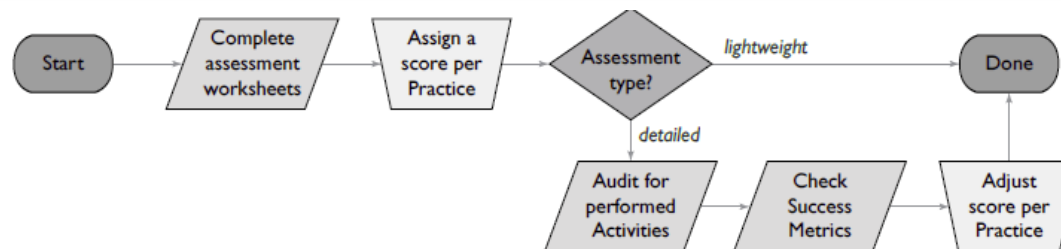
Complete Worksheet, assign score per practice.



Detailed Assessment

After completion of the assessment worksheets, additional audit work is performed to check the organization to ensure the Activities prescribed by each Practice are in place.

Additionally since each Practice also specifies Success Metrics, that data should be collected to ensure that the organization is performing as expected.



Score Cards



Used to show gaps in the current program

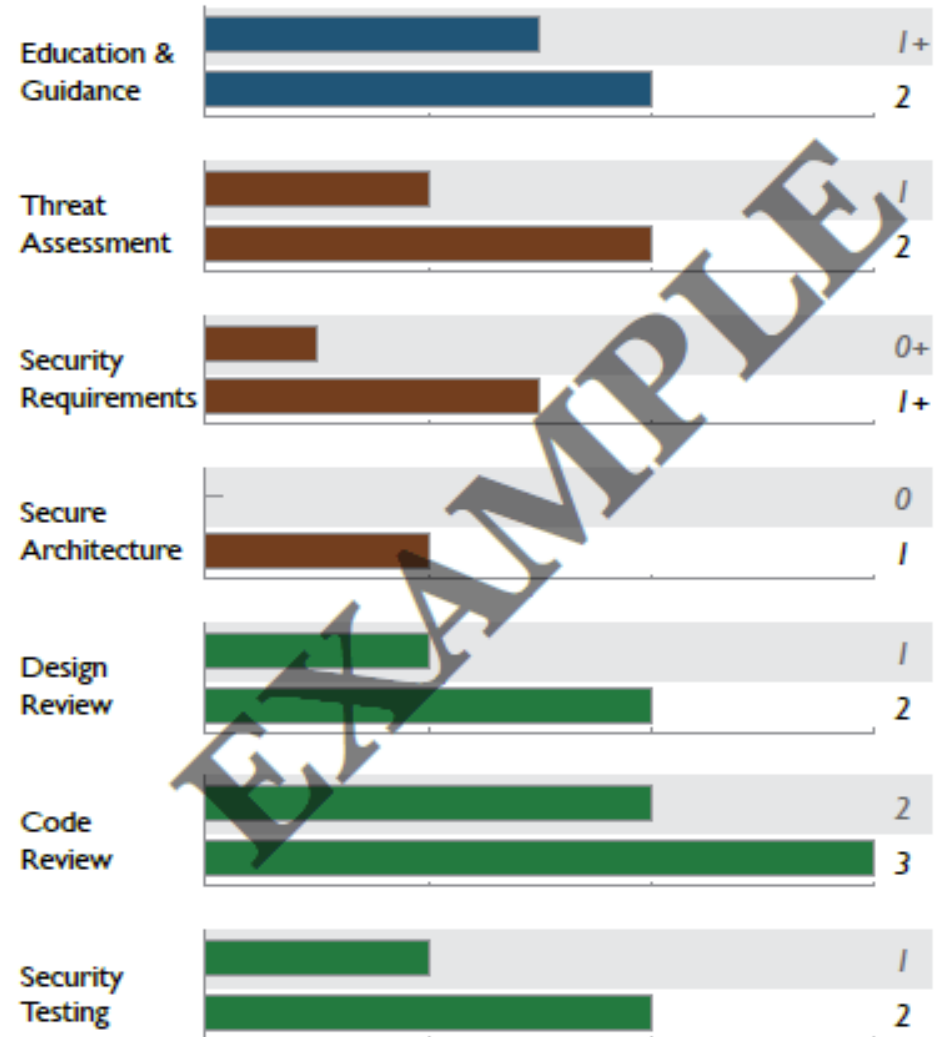
Show current vs. proposed progress

Interim scorecards can be used to show current status.

Example Score Card

Example to the right shows the level of each Program at the start of the phase, and the current level of the program.

A lot of information about change in Software Assurance in a simple graphic.



Build your Program Roadmap

A roadmap is a phased plan for achieving Objectives for each security Function

Use Scorecard to see what Practices need improvement

Create Phased roadmap to improve the practices that YOU feel need the most improvement

Take small steps, apply the activities and success metrics to make small improvements

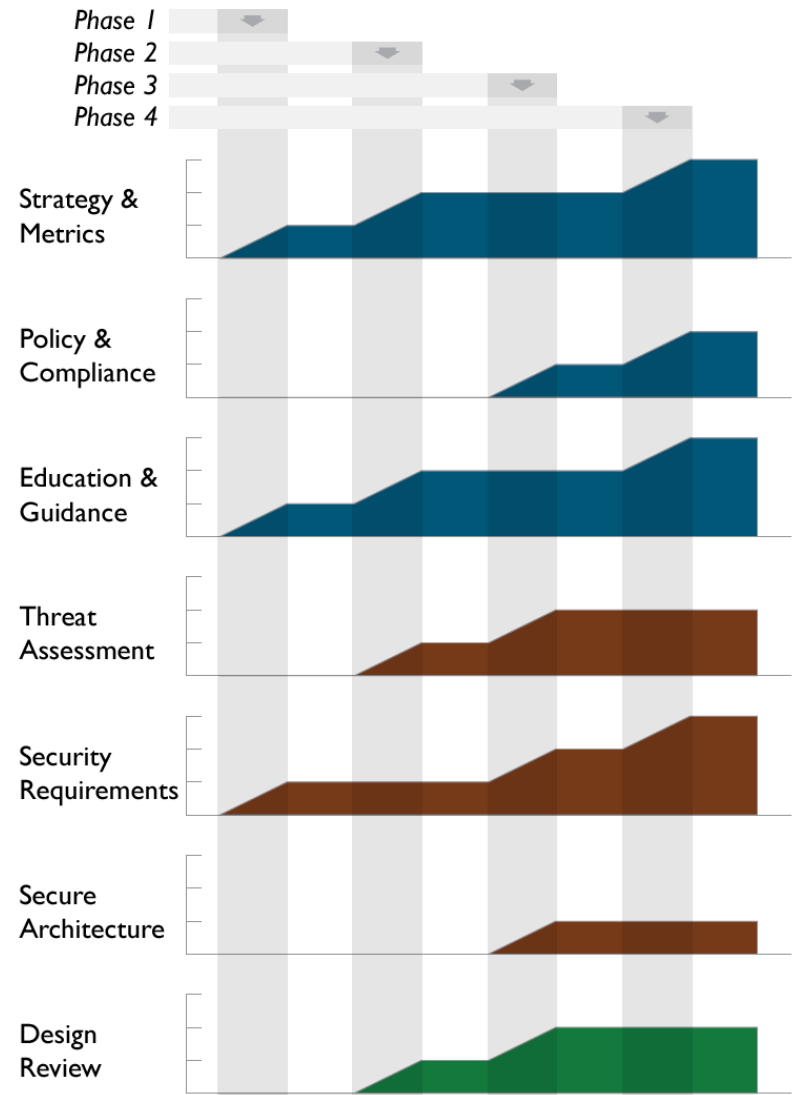
Redo scorecards often to check progress

Roadmap Example

In the sample on the right in Phase 1, several Functions are moved from 0 to 1

In Phase 2, some are further advanced from 1 to 2, some remain at 1, and others are moved from 0 to 1

SAMM includes case studies with specific details on implementation



Summary

Model to measure maturity of your software assurance program.

Based on 4 Business Functions

- Governance
- Construction
- Verification
- Deployment

Looks kind of like a SDLC doesn't it ?

Can be light weight or in depth, simple or complex.

Provides estimates of man hours, who should be involved, success metrics, activities.

More Info

<http://www.owasp.org/index.php/SAMM>

<http://www.opensamm.org/>

<https://lists.owasp.org/mailman/listinfo/samm>

<http://www.owasp.org/index.php/CLASP>

Questions ?

OMG!

HACKERS CAN TURN YOUR HOME COMPUTER INTO A BOMB

By RANDY JEFFRIES / Weekly World News

WASHINGTON — Right now, computer hackers have the ability to turn your home computer into a bomb and blow you to Kingdom Come — and they can do it anonymously from thousands of miles away!

Experts say the recent "break-ins" that paralyzed the Amazon.com, Buy.com and eBay websites are tame compared to what will happen in the near future.

Computer expert Arnold Yabenson, president of the Washington-based consumer group National CyberCrime Prevention Foundation (NCCPF), says that as far as computer crime is concerned, we've only seen the tip of the iceberg.

"The criminals who knocked out those three major online businesses are —"

... & blow your family to smithereens!

KABOOM! It might not look like it, but an innocent home computer like this one can be turned into a deadly weapon.

