

OpenSAMM Best Practices, Lessons from the Trenches

Seba Deleersnyder
seba@owasp.org

Bart De Win
bart.dewin@owasp.org

OpenSAMM project co-leaders

Bart / Seba ?



Sebastien Deleersnyder

15+ years developer / information security experience

Belgian OWASP chapter founder
OWASP volunteer

Co-organizer www.BruCON.org

Application security specialist Toreon



Bart De Win, Ph.D.

15+ years experience in secure software development

Belgian OWASP chapter co-leader

Author of >60 publications

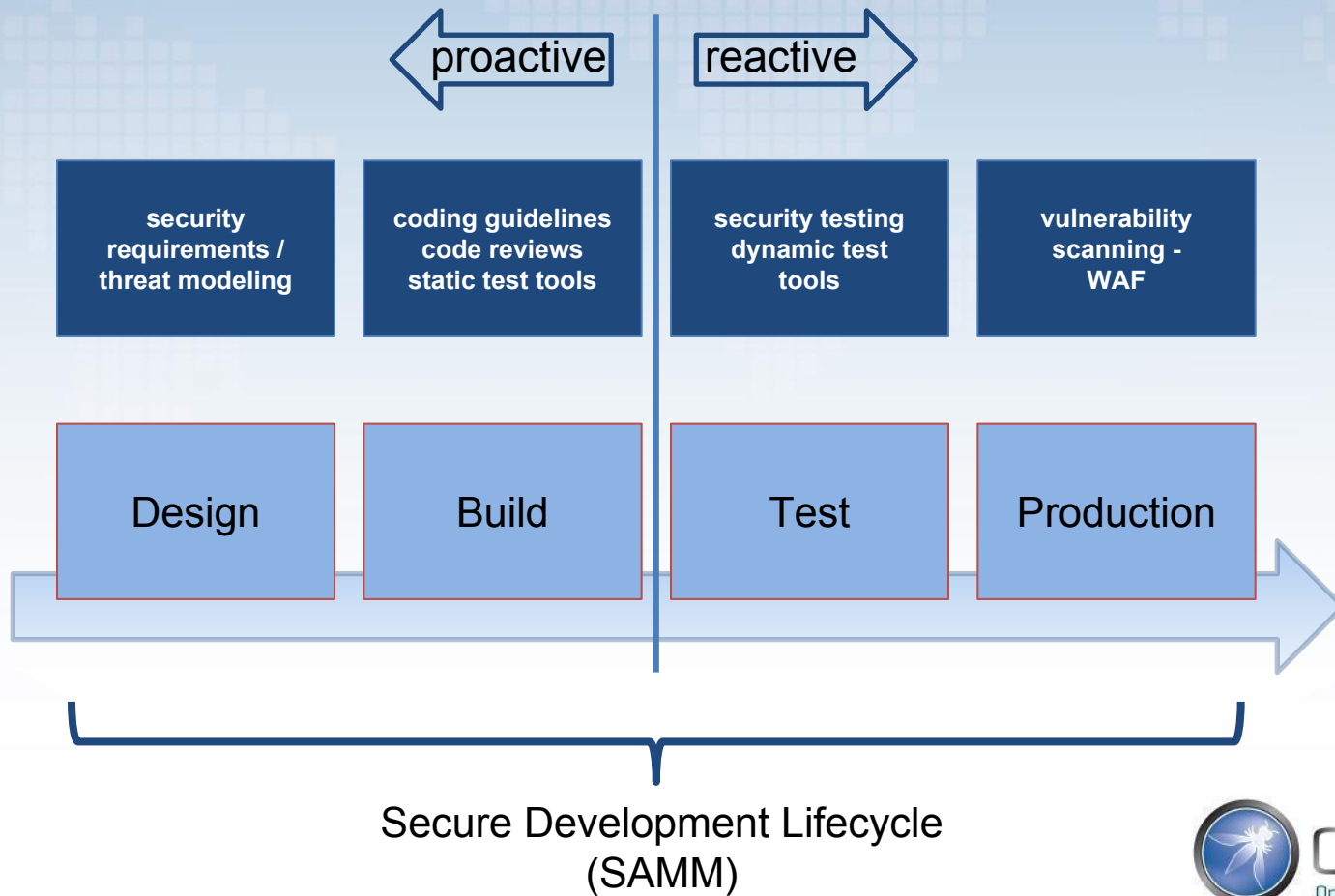
Security consultant PwC



Agenda

- Integrating software assurance?
- OpenSAMM
- Quick Start
- Lessons Learned
- Resources & Self-Assessment
- OpenSAMM Road Map

“Build in” software assurance



Secure Development Lifecycle
(SAMM)

We need a Maturity Model

An organization's behavior changes slowly over time

Changes must be iterative while working toward long-term goals

There is no single recipe that works for all organizations

A solution must enable risk-based choices tailored to the organization

Guidance related to security activities must be prescriptive

A solution must provide enough details for non-security-people

Overall, must be simple, well-defined, and measurable

OWASP Software Assurance Maturity Model (SAMM)



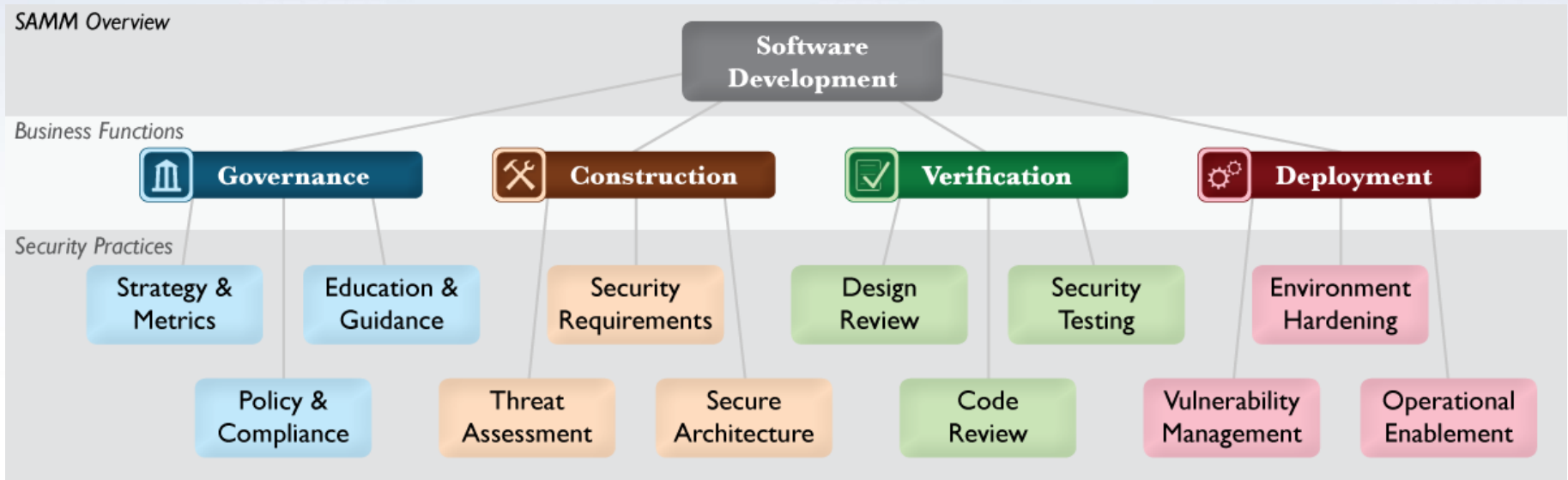
SAMM users

- Dell Inc
- KBC
- ING Insurance
- Gotham Digital Science
- HP Fortify
- ISG ...






SAMM Security Practices

- From each of the Business Functions, 3 Security Practices are defined
- The Security Practices cover all areas relevant to software security assurance
- Each one is a 'silo' for improvement



Example: Education & Guidance

	Education & Guidance ...more on page 42		
	 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	<ul style="list-style-type: none"> A. Conduct technical security awareness training B. Build and maintain technical guidelines 	<ul style="list-style-type: none"> A. Conduct role-specific application security training B. Utilize security coaches to enhance project teams 	<ul style="list-style-type: none"> A. Create formal application security support portal B. Establish role-based examination/certification

Per Level, SAMM defines...

- Objective
- Activities
- Results
- Success Metrics
- Costs
- Personnel
- Related Levels

Education & Guidance

Offer development staff access to resources around the topics of secure programming and deployment

ACTIVITIES

A. Conduct technical security awareness training

Either internally or externally sourced, conduct security training for technical staff that covers the basic tenets of application security. Generally, this can be accomplished via instructor-led training in 1-2 days or via computer-based training with modules taking about the same amount of time per developer.

Course content should cover both conceptual and technical information. Appropriate topics include high-level best practices surrounding input validation, output encoding, error handling, logging, authentication, authorization. Additional coverage of commonplace software vulnerabilities is also desirable such as a Top 10 list appropriate to the software being developed (web applications, embedded devices, client-server applications, back-end transaction systems, etc.). Whenever possible, use code samples and lab exercises in the specific programming language(s) that applies.

To rollout such training, it is recommended to mandate annual security training and then hold courses (either instructor-led or computer-based) as often as required based on development head-count.

B. Build and maintain technical guidelines

For development staff, assemble a list of approved documents, web pages, and technical notes that provide technology-specific security advice. These references can be assembled from many publicly available resources on the Internet. In cases where very specialized or proprietary technologies permeate the development environment, utilize senior, security-savvy staff to build security notes over time to create such a knowledge base in an ad hoc fashion.

Ensure management is aware of the resources and briefs oncoming staff about their expected usage. Try to keep the guidelines lightweight and up-to-date to avoid clutter and irrelevance. Once a comfort-level has been established, they can be used as a qualitative checklist to ensure that the guidelines have been read, understood, and followed in the development process.

RESULTS

- ◆ Increased developer awareness on the most common problems at the code level
- ◆ Maintain software with rudimentary security best-practices in place
- ◆ Set baseline for security know-how among technical staff
- ◆ Enable qualitative security checks for baseline security knowledge

SUCCESS METRICS

- ◆ >50% development staff briefed on security issues within past 1 year
- ◆ >75% senior developers/architect staff briefed on security issues within past 1 year
- ◆ Launch technical guidance within 3 months of first training

COSTS

- ◆ Training course buildout or license
- ◆ Ongoing maintenance of technical guidance

PERSONNEL

- ◆ Developers (1-2 days/yr)
- ◆ Architects (1-2 days/yr)

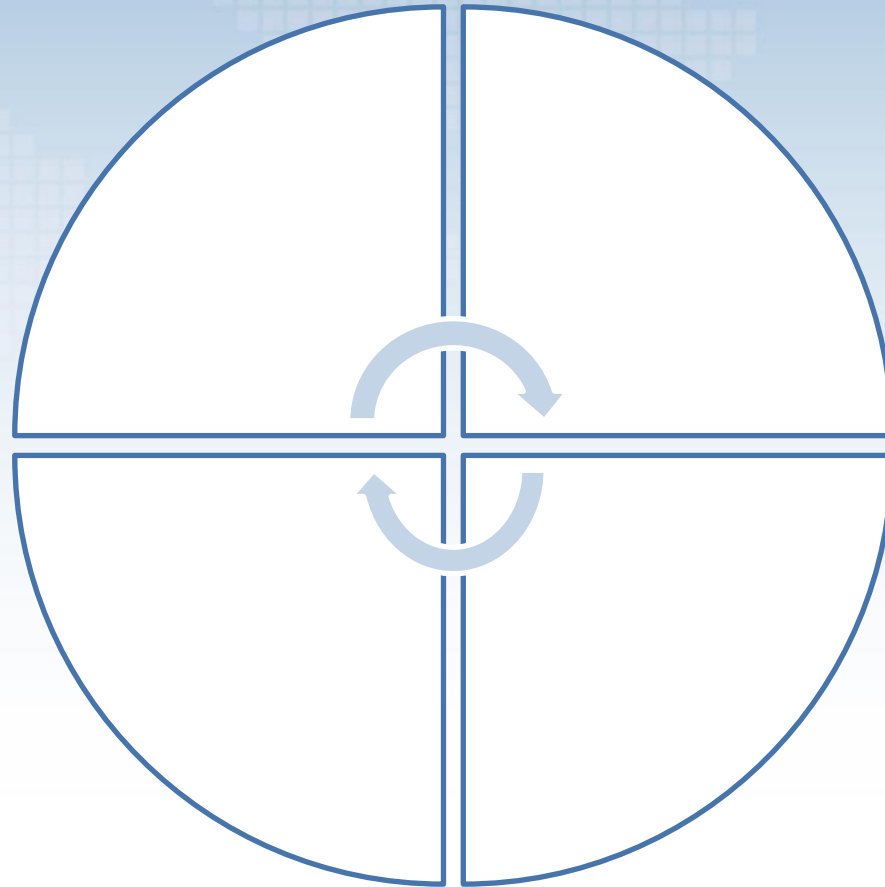
RELATED LEVELS

- ◆ Policy & Compliance - 2
- ◆ Security Requirements - 1
- ◆ Secure Architecture - 1

4 SAMM/The Source Practices - v1.0

 Open Web Application Security Project

SAMM Quick Start



Assess

- SAMM includes assessment worksheets for each Security Practice

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?	
◆ 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?	
◆ 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?	



Lessons Learned – Organisation Specific

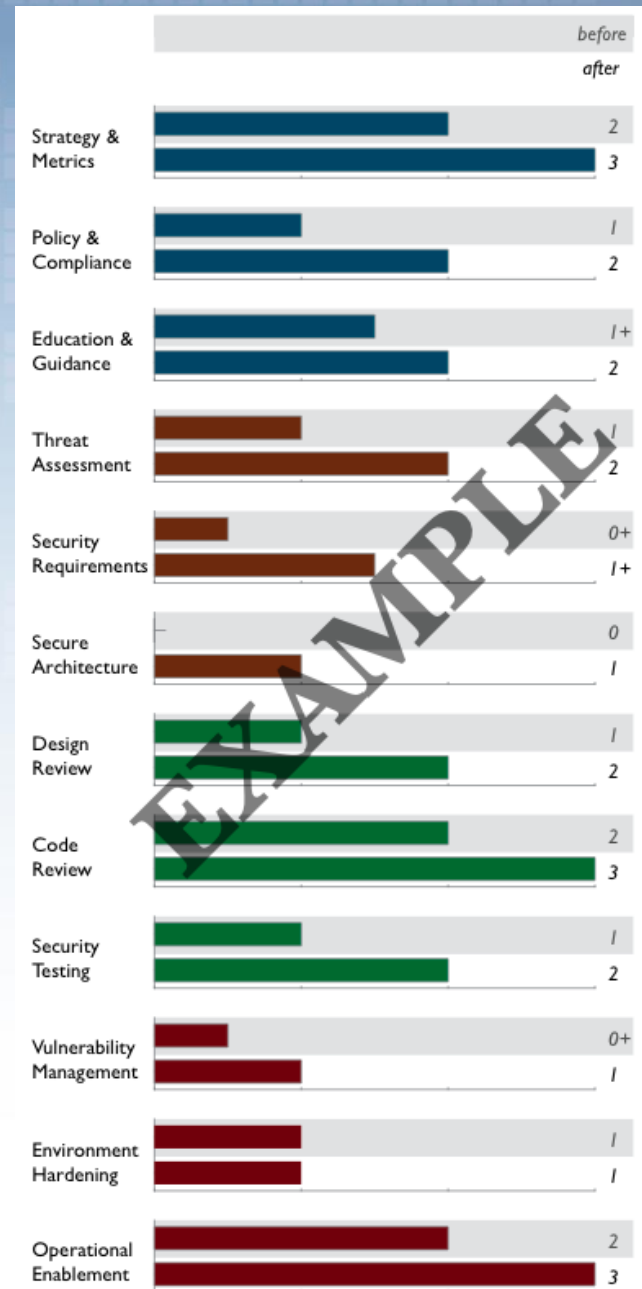
- Pre-screen general software development maturity
- Define assessment scope in organisation:
 - Organisation wide
 - Selected Business Units
 - Development Groups (internal, supplier)
 - IT infrastructure Groups (hosting internal, cloud)
- Involve key stakeholders
Invaluable for awareness & education
- Apply CONSISTENT (same interviewers) within same organisation

Lessons Learned – Interview / Scoring

- Adapt & select subset questionnaire per profile
(risk management, development, IT infrastructure, ...)
- Try different formats: interview style, workshops
- Capture more details:
 - “Adjusted” scoring
 - Ask percentage instead of Yes/No
 - If Yes: request CMM level for activity
 - Ask about strengths & weaknesses
- Validate results:
 - Repeat questions to several people
 - Lightweight vs full approach
 - Anonymous interviews
 - Aggregate gathered information

Goal

- Gap analysis
 - Capturing scores from detailed assessments versus expected performance levels
- Demonstrating improvement
 - Capturing scores from before and after an iteration of assurance program build-out
- Ongoing measurement
 - Capturing scores over consistent time frames for an assurance program that is already in place



Goal – Lessons Learned

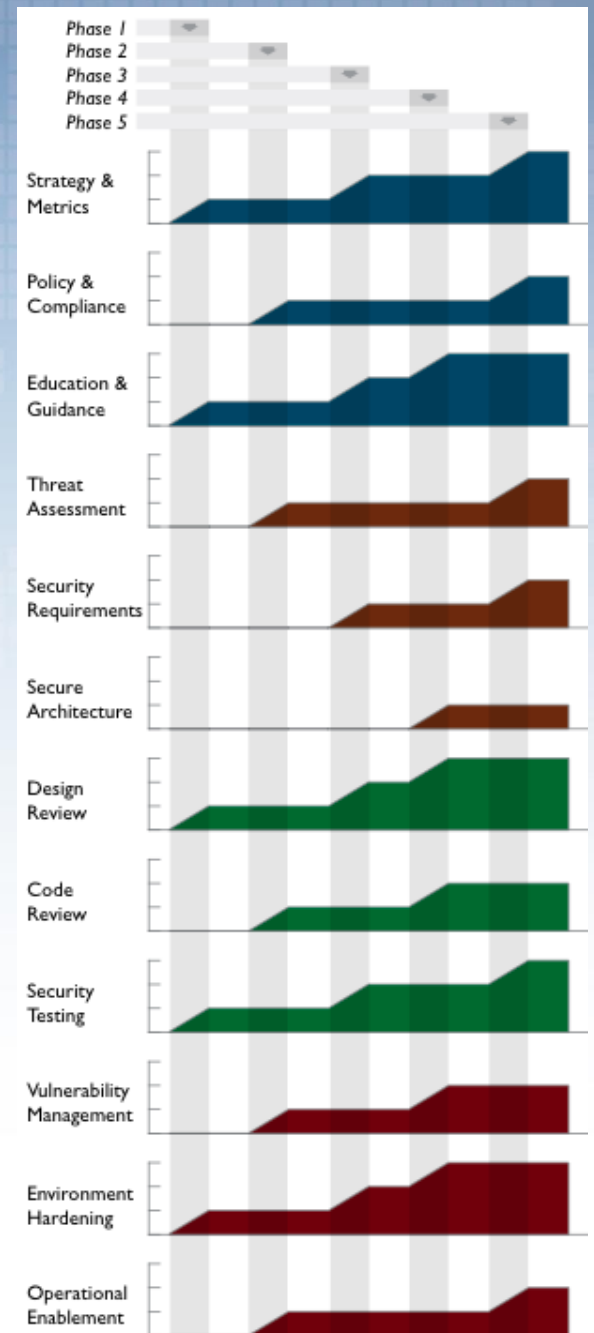
- Link to the organisational context
 - Specific Business Case (ROI)
 - Organisation objectives / risk profile
- Think carefully about selection
 - So you want to achieve all 3's. Hmm. Who are you, NSA ?
 - Link to industry level
 - Respect practice dependencies
 - It can make sense not to include particular low-level activities, or to lower a current level

Goal – Lessons Learned

- Get consensus, management support
- Be ready for budget questions (linked to Plan phase)
 - MD, CAPEX, OPEX
 - General stats about %'s
- Create & reuse own organisation template

Plan

- Roadmaps: to make the “building blocks” usable
- Roadmaps templates for typical kinds of organizations
 - Independent Software Vendors
 - Online Service Providers
 - Financial Services Organizations
 - Government Organizations
- Tune these to your own targets / speed



Plan – Lessons Learned

- Identify quick wins (focus on success cases)
- Start with awareness / training
- Adapt to upcoming release cycles / key projects
- Spread effort & “gaps to close” over realistic iterations

- Spread work, roles & responsibilities
 - SW security competence centre, development, security, operations
 - For instance service portfolio and guidelines: when and who ?
- Take into account dependencies

- Be ready to adapt planning

Plan – Budgeting

- Average budget impact 5%-15% on project
- Cost of tooling
 - Central procurement vs per development group
- Cost of training
 - Do not forget internal/external time spent
- Cost of external suppliers / outsourcing
- Different technology stacks will impact budget

Implement: 150+ OWASP resources

Blank rounded rectangular boxes for content input, arranged in a list structure with varying indentation.

Implement – Lessons Learned

- Adapt & reuse SAMM to your organisation
 - Categorize applications: High, Medium, Low based on risk: e.g. Internet facing, transactions, ...
 - Recheck progress & derive lessons learned at each iteration
 - Create & improve reporting dashboard
 - Application & process metrics
 - Treat new & legacy code bases differently
-
- Agile: differentiate between Every Sprint, Bucket & one-time AppSec activities
 - Balance planning on people, process, knowledge and tools

Lessons Learned – AppSec Competence Centre

- Inject & spread best practices
- “market & promote” – do not become risk/audit function
- Do not become operational bottle-neck
- Spread/hand-over knowledge to champions throughout organisation
- Create & nurture AppSec community

SAMM Resources

www.opensamm.org

- Presentations
- Quick Start (to be released)
- Assessment worksheets / templates
- Roadmap templates
- Translations (Spanish, Japanese, ...)
- SAMM mappings to ISO/EIC 27034 – BSIMM – PCI (to be released)
- NEW: Training material

NEW: Self-Assessment Online

SAMM Self Assessment [About](#) [Login](#) [Sign Up](#)

Welcome to the SAMM Self Assessment tool
Want to read more about OpenSAMM? [Click here](#)

[Governance](#) [Construction](#) [Verification](#) [Deployment](#) [Scorecard](#)

Strategy & Metrics

Question	Response	Rating
Is there a software security assurance program already in place?	<input checked="" type="checkbox"/> Yes	2
Do most of the business stakeholders understand your organization's risk profile?	<input checked="" type="checkbox"/> Yes	
Is most of your development staff aware of future plans for the assurance program?	<input checked="" type="checkbox"/> Yes	
Are most of your applications and resources categorized by risk?	<input checked="" type="checkbox"/> Yes	
Are risk ratings used to tailor the required assurance activities?	<input checked="" type="checkbox"/> Yes	
Does most of the organization know about what's required based on risk ratings?	<input checked="" type="checkbox"/> Yes	
Is per-project data for cost of assurance activities collected?	<input type="checkbox"/> Yes	
Does your organization regularly compare your security spend with other organizations?	<input type="checkbox"/> Yes	

Policy & Compliance

Question	Response	Rating
Do most project stakeholders know their project's compliance status?	<input checked="" type="checkbox"/> Yes	1
Are compliance requirements specifically considered by project teams?	<input checked="" type="checkbox"/> Yes	
Does the organization utilize a set of policies and standards to control software development?	<input type="checkbox"/> Yes	
Are project teams able to request an audit for compliance with policies and standards?	<input type="checkbox"/> Yes	
Are projects periodically audited to ensure a baseline of compliance with policies and standards?	<input type="checkbox"/> Yes	
Does the organization systematically use audits to collect and control compliance evidence?	<input type="checkbox"/> Yes	

<https://ssa.asteriskinfosec.com.au>

SAMM Roadmap

Build the SAMM community:

- Grow list of SAMM adopters
- Workshops at conferences
- Dedicated SAMM summit

V1.1:

- Incorporate Quick Start / tools / guidance / OWASP projects
- Revamp SAMM wiki

V2.0:

- Revise scoring model
- Model revision necessary ? (12 practices, 3 levels, ...)
- Application to agile
- Roadmap planning: how to measure effort ?
- Presentations & teaching material
- ...

Get involved

- Project mailing list / work packages
- Use and donate (feed)back!
- Donate resources
- Sponsor SAMM



Critical Success Factors

- Get initiative buy-in from all stakeholders
- Adopt a risk-based approach
- Awareness / education is the foundation
- Integrate security in your development / acquisition and deployment processes
- Measure: Provide management visibility

Measure & Improve!

OpenSAMM.org

Mapping Projects / SAMM

Project	Type	Level	SAMM Practice	Remarks
AntiSamy	Code	Flagship	SA2	
Enterprise Security API	Code	Flagship	SA3	
ModSecurity Core Rule Set	Code	Flagship	EH3	
CSRFGuard	Code	Flagship	SA2	
Web Testing Environment	Tools	Flagship	ST2	
WebGoat	Tools	Flagship	EG2	
Zed Attack Proxy	Tools	Flagship	ST2	
Application Security Verification Standard	Documentation	Flagship	DR2	ASVS-L4
Application Security Verification Standard	Documentation	Flagship	CR3	ASVS-L4
Application Security Verification Standard	Documentation	Flagship	ST3	ASVS-L4
Code Review Guide	Documentation	Flagship	CR1	
Codes of Conduct	Documentation	Flagship		not applicable
Development Guide	Documentation	Flagship	EG1	
Secure Coding Practices - Quick Reference Guide	Documentation	Flagship	SR1	
Software Assurance Maturity Model	Documentation	Flagship	SM1	Recursiveness :-)
Testing Guide	Documentation	Flagship	ST1	
Top Ten	Documentation	Flagship	EG1	

	Type	Level	SAMM Practice	Remarks
	Tools	Labs	EG1	
	Tools	Labs	ST1	
	Tools	Labs	ST1	
	Tools	Labs	ST1	
	Tools	Labs	EG1	
	Tools	Labs	ST1	
	Tools	Labs	ST1	
	Tools	Labs	ST1	
	Tools	Labs	SA2	
	Tools	Labs		not applicable
	Tools	Labs	ST1	
	Tools	Labs	CR2	
	Tools	Labs	ST1	
	Tools	Labs	EG1	
	Tools	Labs	ST2	
	Tools	Labs	CR2	
	Tools	Labs	ST1	
Virtual Worlds	Tools	Labs	ST1	
	Tools	Labs	EG1	
	Tools	Labs	ST1	
	Tools	Labs	ST1	
	Tools	Labs	ST1	
	Tools	Labs	ST1	
	Tools	Labs	CR2	
	Documentation	Labs	EG1	
	Documentation	Labs	EH3	
	Documentation	Labs	SA2	
	Documentation	Labs	EG1	
	Documentation	Labs	EG1	
	Documentation	Labs	ST1	
	Documentation	Labs	SR3	
	Documentation	Labs	EG1	
	Documentation	Labs	EH3	

Wapiti	Tools	Labs	ST1	
Web Browser Testing System	Tools	Labs	ST1	
WebScarab	Tools	Labs	ST1	
Webslayer	Tools	Labs	ST1	
WSFuzzer	Tools	Labs	ST1	
Yasca	Tools	Labs	CR2	
AppSec Tutorials	Documentation	Labs	EG1	
AppSensor	Documentation	Labs	EH3	
AppSensor	Documentation	Labs	SA2	
Cloud 10	Documentation	Labs	EG1	
CTF	Documentation	Labs	EG1	
Fuzzing Code	Documentation	Labs	ST1	
Legal	Documentation	Labs	SR3	
Podcast	Documentation	Labs	EG1	
Virtual Patching Best Practices	Documentation	Labs	EH3	

OWASP Projects Coverage

Governance						
Strategy & Metrics		Policy & Compliance		Education & Guidance		
SM1	1	PC1	0	EG1	10	
SM2	0	PC2	0	EG2	1	
SM3	0	PC3	0	EG3	0	
	1		0		11	12
Construction						
Threat Assessment		Security Requirements		Security Architecture		
TA1	0	SR1	1	SA1	0	
TA2	0	SR2	0	SA2	4	
TA3	0	SR3	1	SA3	1	
	0		2		5	7
Verification						
Design Review		Code Review		Security Testing		
DR1	0	CR1	1	ST1	18	
DR2	1	CR2	3	ST2	3	
DR3	0	CR3	1	ST3	1	
	1		5		22	28
Deployment						
Vulnerability Management		Environment Hardening		Operational Hardening		
VM1	0	EH1	0	OE1	0	
VM2	0	EH2	0	OE2	0	
VM3	0	EH3	3	OE3	0	
	0		3		0	3

SDLC Cornerstones (recap)

