OWASP SAMM
软件保证成熟度模型

王颉

OWASP

关于我

- 王颉（http://www-staff.lboro.ac.uk/~cojw8/index.htm）
- 现英国拉夫堡大学（Loughborough University）电子工程系高速网络（High Speed Network）研究组博士生
- 主要研究方向：
  - 入侵检测系统
  - 攻击树建模
  - 计算机网络QoS分析
- OWASP贡献
  - 2010年OWASP新闻简报中文翻译
  - 2010年OWASP Top 10中文翻译组成员
  - OWASP SAMM中文翻译组成员
  - OWASP中文项目组负责人之一

SAMM项目说明

- 原项目领导人： Pravir Chandra
- 原项目类型： 文档
- 原项目赞助方： 美国Fortify公司
- 原项目参与人员：

Fabio Arciniegas	Matt Bartoldus	Sebastien Deleersnyder
Jonathan Carter	Darren Challey	Brian Chess
Dinis Cruz	Justin Derry	Bart De Win
James McGovern	Matteo Meucci	Jeff Payne
Gunnar Peterson	Jeff Piper	Andy Steingruebl
John Steven	Chad Thunberg	Colin Watson
Jeff Williams		

- 项目网站：
  - https://www.owasp.org/index.php/Category:Software_Assurance_Maturity_Model
  - http://www.opensamm.org/

SAMM中文项目说明

- 中文项目参与人员：
  - 王颉（翻译兼Alpha版本审核）
  - Yitao Wang和Lisa Wei（Beta版本审核）
- 项目类型：翻译
- 当前进度
  - 已发布Alpha版本（下载地址：https://www.owasp.org/images/c/c2/%E8%BD%AF%E4%BB%B6%E4%BF%9D%E8%AF%81%E6%88%90%E7%86%9F%E5%BA%A6%E6%A8%A1%E5%9E%8B%28Alpha%29.pdf）
  - 正在审核Beta版本

额外说明

- 欢迎指正翻译错误
- 欢迎大家提出宝贵的意见
  - 如何在国内的环境里展开应用？
  - 是否存在缺陷？
  - 是否可以改进？

目录

- SAMM介绍
- 理解SAMM
- 应用SAMM
- 具体安全实践
- 结束语

软件保证成熟度模型（SAMM）

建立模型的动力

- 一个组织的行为随着时间的推移而缓慢的改变；
  - 改变必须循序渐进得向长期目标进行
- 没有单一的方法可作用于所有的组织；
  - 一个解决方案必须允许组织根据风险而选择
- 与安全措施相关的指导必须是规范的；
  - 一个解决方案必须为非安全人员提供足够的细节信息
- 总的来说，建立后的成果必须简单、明确定义、可衡量。

什么是软件保证成熟度模型（SAMM）？

- 软件保证成熟度模型；
  Software Assurance Maturity Model (SAMM)
- 一个开放的框架；
- 帮助组织制定并实施针对组织所面临来自软件安全的特定风险的策略。

SAMM的目标

- 创建明确定义和可衡量的目标；
- 涉及到软件开发的任何业务；
- 可用于小型、中型和大型组织。

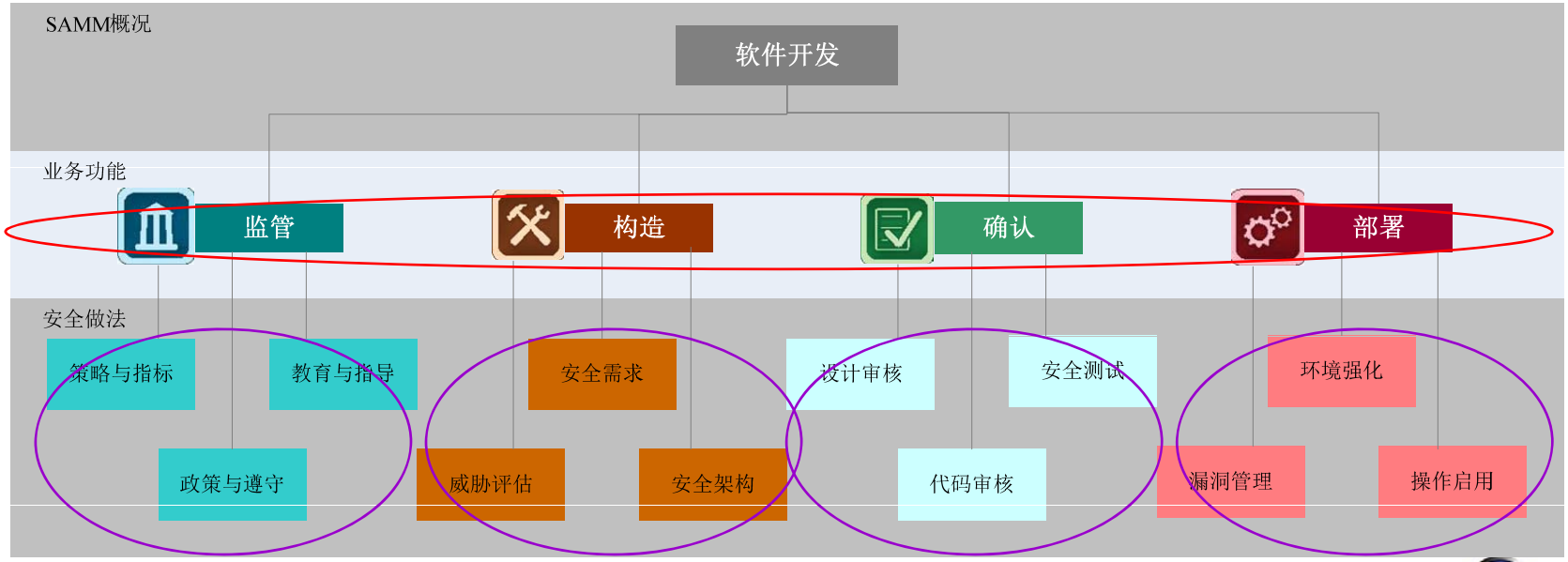
SAMM的目的

- 评估一个组织已有的软件安全实践；
- 建立一个迭代的权衡的软件安全保证计划；
- 证明安全保证计划带来的实质性改善；
- 定义并衡量组织中与安全相关的措施。

理解SAMM模型

SAMM的业务功能

- 从企业组织与软件开发的核心活动开始；
- 在最高等级上，SAMM设置了四种关键业务功能；
- 对于每一个业务功能，SAMM设置了三个安全措施；
- 对于每一个安全措施，SAMM设置了三个成熟度等级。



成熟度等级

- 每一个安全措施定义了三个等级。
  - 以证明组织是如何随着时间而改变的。
- 一个措施的三个等级：
  - 0:隐起点，措施尚未实现；
  - 1:对安全实践有了初步了解并有所专门的提供；
  - 2:提高了安全实践的效率和（或）有效性；
  - 3:在一定规模上综合掌握了安全实践。

循序渐进改善的方法

- 每一个安全实践都是一个成熟度领域。
- 一个目标的成功，代表了一系列安全实践得采用。
- 简单地说，以分阶段的方式改善一个保证计划：
  - 选择安全实践去改善保证计划的下一个阶段；
  - 通过执行相关活动指定的成功衡量标准，以得到每个实践的下一个目的。

应用SAMM

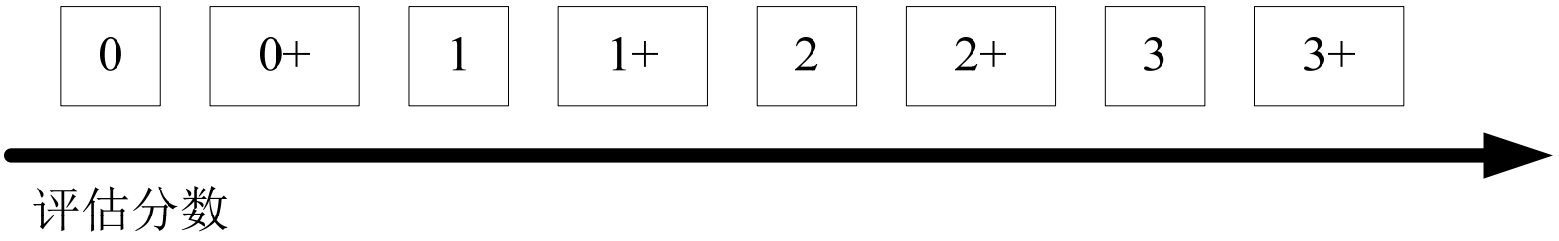
评估执行

■ SAMM的每一个安全实践，都包含了评估记录表

安全需求	是/不是	
✦大多数项目团队在开发过程中是否明确阐述的一些安全需求？		SR1
✦项目团队是否从最佳实例和遵守的指导中引导需求？		
✦大多数的业务拥有者是否审核相关项目的访问控制矩阵？		SR2
✦项目团队是否根据来自于其他安全活动的反馈来详细阐述需求？		
✦大多数的业务拥有者是否为安全需求审核厂商的协议？		SR3
✦项目团队描述的安全需求是否被审计？		

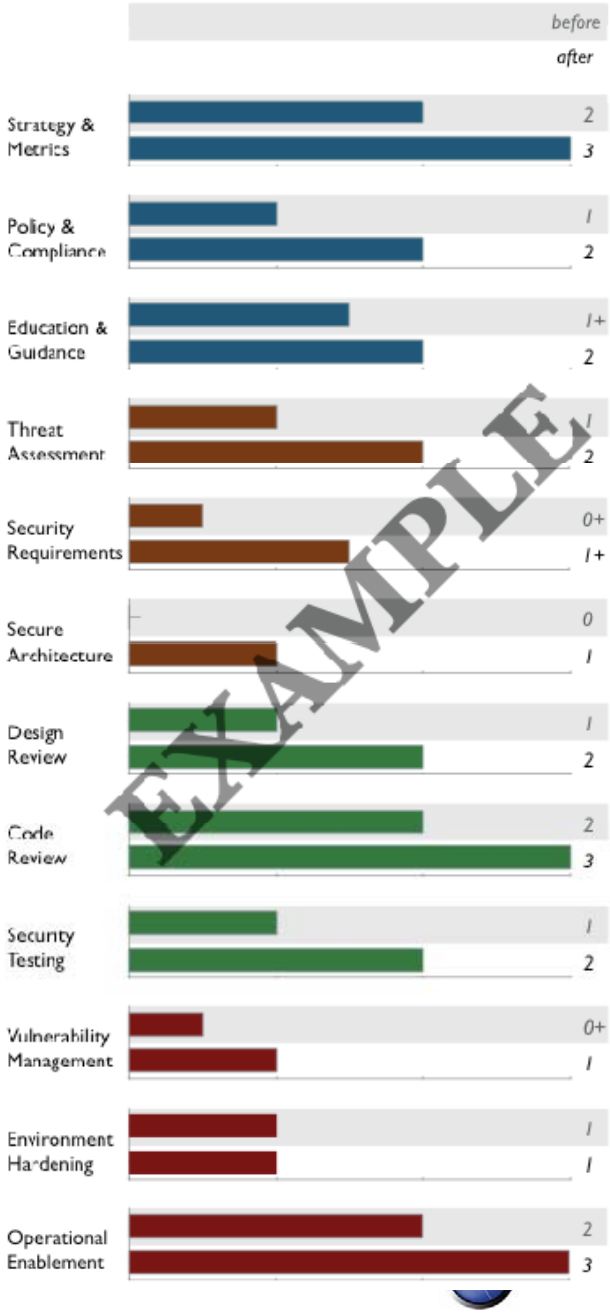
评估处理

- 支持简便评估和详细评估
- 简便方法：直接根据回答评分
- 详细方法：执行额外的审计以后，再评分
- 组织的评估得分可能处于两个级别之间，因而采用“＋”



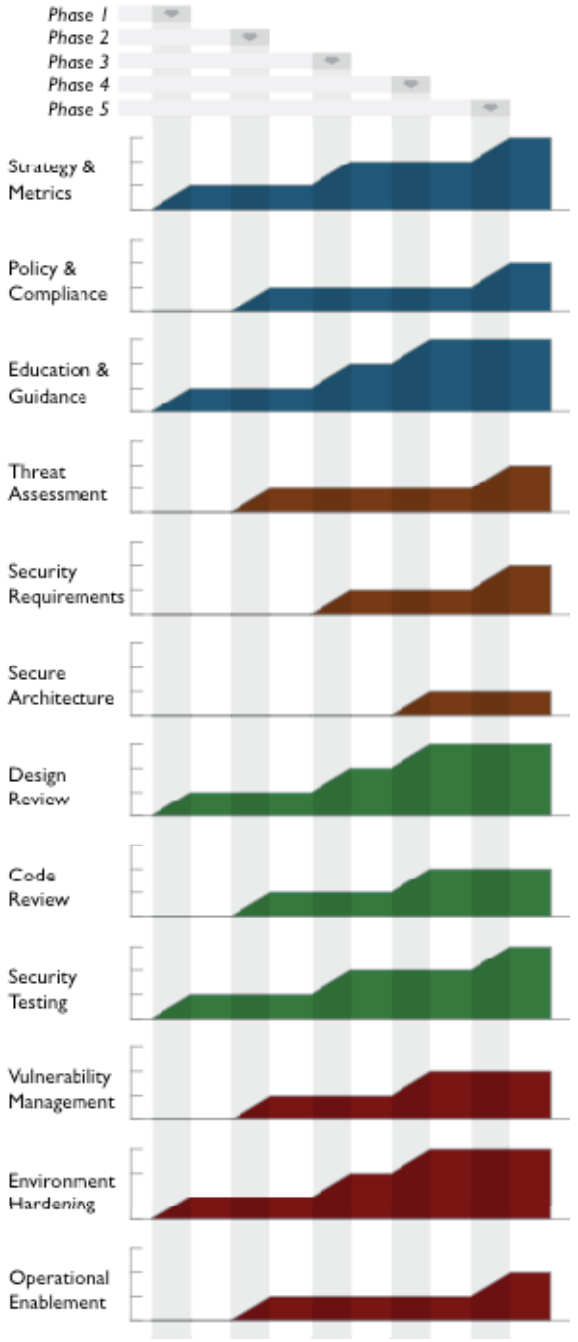
创建记分卡

- 持续衡量
  - 为一个已经就位的保证计划，在持续的时间框架内获得分数；
- 差距分析
  - 将获得的详细评估结果与预期的性能等级做比较，获得分数；
- 改善证明
  - 在一次安全计划迭代建立完成的前后，获得分数；



路线图模版

- 为使用安全实践，SAMM为以下一些有代表性的组织提供了路线图模版：
  - 独立软件供应商；
  - 在线服务提供商；
  - 金融服务机构；
  - 政府组织。
- 选择这些组织类型的原因：
  - 它们代表了常见的用例；
  - 每个组织都有对于典型软件导致的多种风险；
  - 每个组织保证计划的最优方案有所不同。



具体安全实践

策略与指标（Strategy & Metrics）

	策略与指标		
	SM1	**SM2**	**SM3**
目标	为组织内的软件安全建立统一的战略路线图。	衡量数据和软件资产的相对价值，并选择风险容忍度。	使安全成本与相关业务指标和资产价值相一致。
措施	A. 评估整体业务风险概况； B. 建立并维护保证计划路线图。	A. 根据业务风险将数据和应用程序分类； B. 建立并衡量每个分组的安全目的。	A. 引导周期性地全行业成本比较； B. 为以前的安全成本收集度量标准。

政策与遵守（Policy & Compliance）

	政策与遵守		
	PC1	**PC2**	**PC3**
目标	了解组织的相关监管和遵守要求。	建立安全和遵守的基准线，并了解每个项目的风险。	要求遵守标准，并衡量项目是否符合全组织的政策和标准。
措施	A. 确定并监控外部的遵守驱动因素； B. 建立并维护遵守指导。	A. 为安全和遵守建立政策和标准； B. 建立项目审计实践。	A. 为项目建立遵守关卡； B. 为审计数据的采集，采用解决方案。

教育与指导（Education & Guidance）

	教育与指导		
	EG1	**EG2**	**EG3**
目标	为开发人员提供关于以安全编程和部署为主题的资源。	为软件生命周期中所有的人员提供基于角色的安全开发详细指导。	实施综合的安全培训，并为员工进行基本知识的认证检验。
措施	A. 实施技术安全意识的培训； B. 建立并维护技术指导。	A. 实施针对特定角色的应用程序安全培训； B. 聘用安全指导专家增强项目团队。	A. 建立正式的应用程序安全支持门户网站； B. 建立基于角色的考试或认证制度。

威胁评估（Threat Assessment）

	威胁评估		
	TA1	**TA2**	**TA3**
目标	确定并了解组织和单个项目的高级别威胁。	提高威胁评估的准确性，并深入了解每个项目的细节。	将补偿控制与对内部和第三方软件的每个威胁具体联系起来。
措施	A. 建立并维护特定应用程序的威胁模型； B. 根据软件架构建立攻击者概况。	A. 建立并维护每个项目的滥用用例模型； B. 为威胁的度量采用一个权重系统。	A. 明确评估来自第三方组件的风险； B. 用补偿控制详细描述威胁模型。

安全需求（Security Requirements）

	安全需求		
	SR1	**SR2**	**SR3**
目标	在软件需求分析阶段明确地将安全考虑在内。	根据业务逻辑和已知风险增加安全需求的深度。	为所有软件项目和第三方的附属项目强制要求安全需求。
措施	A. 从业务功能推导出安全需求； B. 为需求评估安全和遵守指导。	A. 为资源和能力建立一个访问控制矩阵； B. 根据已知风险指定安全需求。	A. 将安全需求写入供应商协议中； B. 为安全需求扩展审计计划。

安全架构（Secure Architecture）

	安全架构		
	SA1	**SA2**	**SA3**
目标	将主动安全指导的想法引入到软件设计过程中。	将软件设计过程引导向已知安全服务和默认安全设计。	正式控制软件设计过程并验证安全部件的使用。
措施	A. 维护推荐的软件框架列表； B. 将安全原则明确运用到设计中。	A. 明确并促进安全服务和基础设施； B. 明确来自架构的安全设计模式。	A. 建立正式的参照架构和平台； B. 验证框架、模式和平台的使用。

设计审核（Design Review）

	设计审核		
	DR1	**DR2**	**DR3**
目标	为软件设计提供专门的审核，以确保排除已知风险的最低线。	根据安全的最佳实践为软件设计审核提供评估服务。	需求评估并验证已完成部分，以详细了解保护机制。
措施	A. 确定软件攻击层面； B. 根据已知安全需求分析设计。	A. 检查提供安全机制的完整性； B. 为项目团队部署设计审核服务。	A. 为敏感资源开发数据流图； B. 为设计审核建立发布关卡。

代码审核（Code Review）

	代码审核		
	CR1	**CR2**	**CR3**
目标	随机查找基本的代码级漏洞和其他高风险安全问题。	通过自动化方式在开发过程中使代码审核更加准确和有效。	必须进行全面的代码审核过程，以发现语言级别和特定应用程序的风险。
措施	A. 根据已知安全需求建立审核检查列表； B. 为高风险代码执行定点审核。	A. 使用自动化的代码分析工具； B. 将代码分析集成到开发流程当中。	A. 为特定应用程序问题自定义代码分析； B. 为代码审核建立发布关卡。

安全测试（Security Testing）

	安全测试		
	ST1	**ST2**	**ST3**
目标	根据编程和软件需求，建立处理过程以执行基本的安全测试。	通过自动化使在开发过程中的安全测试更加完善和有效。	在部署前要求进行特定应用程序的安全测试以确保基本的安全。
措施	A. 从已知安全需求推出测试用例； B. 为软件发布执行渗透测试。	A. 使用自动化的安全测试工具； B. 将安全测试整合到开发过程中。	A. 为特定应用程序使用自动化的安全测试； B. 为安全测试建立发布关卡。

漏洞管理（Vulnerability Management）

	漏洞管理		
	VM1	**VM2**	**VM3**
目标	理解对于漏洞报告或事件的高级别计划。	为响应过程阐述期望，以改善一致性和交流。	在响应过程中为积极规划提供反馈，而改善分析和数据收集。
措施	A. 为安全事件确定联络点； B. 建立非正式安全响应团队。	A. 建立一致的事件响应流程； B. 采用安全事件报告流程。	A. 为事件执行根源分析； B. 收集每一事件的度量指标。

环境强化（Environment Hardening）

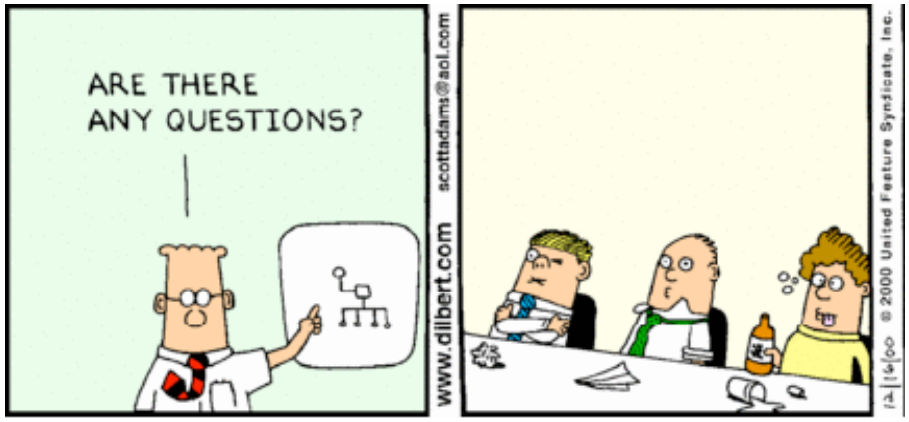
	环境强化		
	EH1	**EH2**	**EH3**
目标	了解应用程序和软件组件的基本操作环境。	通过强化操作环境提高对应用程序操作的信心。	以已知最佳实践验证应用程序的健康和操作环境状态。
措施	A. 维护操作环境说明； B. 确定并安装关键的安全软件升级和补丁。	A. 建立常规补丁管理流程； B. 监控基准基础架构配置状态。	A. 确定并部署相关操作的保护工具； B. 为环境配置扩展审计计划。

操作实现（Operational Enablement）

	操作实现		
	OE1	**OE2**	**OE3**
目标	实现开发团队和操作人员之间对于与安全相关的关键数据的沟通交流。	通过提供详细的步骤为持续的安全操作提高期望。	针对完整性而对安全信息和部件检验进行强制宣传。
措施	A. 为部署获得的重要的安全信息； B. 为典型的应用程序警报记录流程。	A. 创建每次发布的变更管理流程； B. 维护正式的操作安全指南。	A. 为操作信息扩展操作审计计划； B. 对应用程序组件执行代码签名。

结束语

- 更多信息，请访问 http://www.opensamm.org
- 中文Alpha版本文档的下载地址：
  - https://www.owasp.org/images/c/c2/%E8%BD%AF%E4%BB%B6%E4%BF%9D%E8%AF%81%E6%88%90%E7%86%9F%E5%BA%A6%E6%A8%A1%E5%9E%8B%28Alpha%29.pdf
- 预期Beta版本发布：
  - 时间：2011年8月
  - 主要变化：文档格式将与英文源文档格式相同
    存在的一些翻译错误（如果有的话）



欢迎大家交流！

Email：wangjie8578@yahoo.com.cn