
SAMM (2)

AN INTRODUCTION

About

The security problem

What is SAMM

Use cases

Questions



About

Esfandiar Behrouz

- Cybersecurity and Technology Practitioner and Consultant 20+ years
- Specializing in adaptive cybersecurity solutions and building programs from ground up
- Founder at Comply Frame consulting and Cybersecurity and Risk Advisor at Clearview Systems
- Active member of the cybersecurity communities (OWASP, ISACA, ISC2, ISSA)

Email: esfandb@complyframe.com, esfandiar.behrouz@owasp.org

The Security Problem

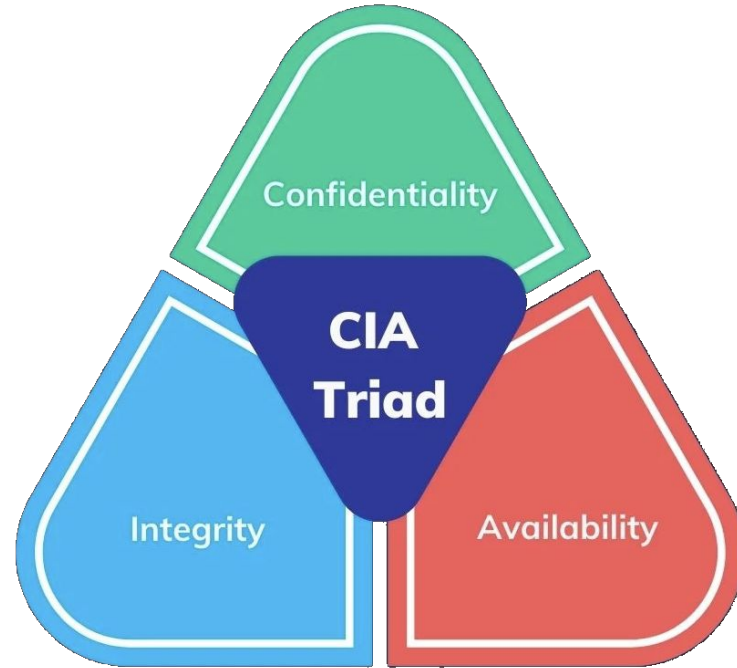


Security

Confidentiality

Integrity

Availability





Security is not one thing nor three things

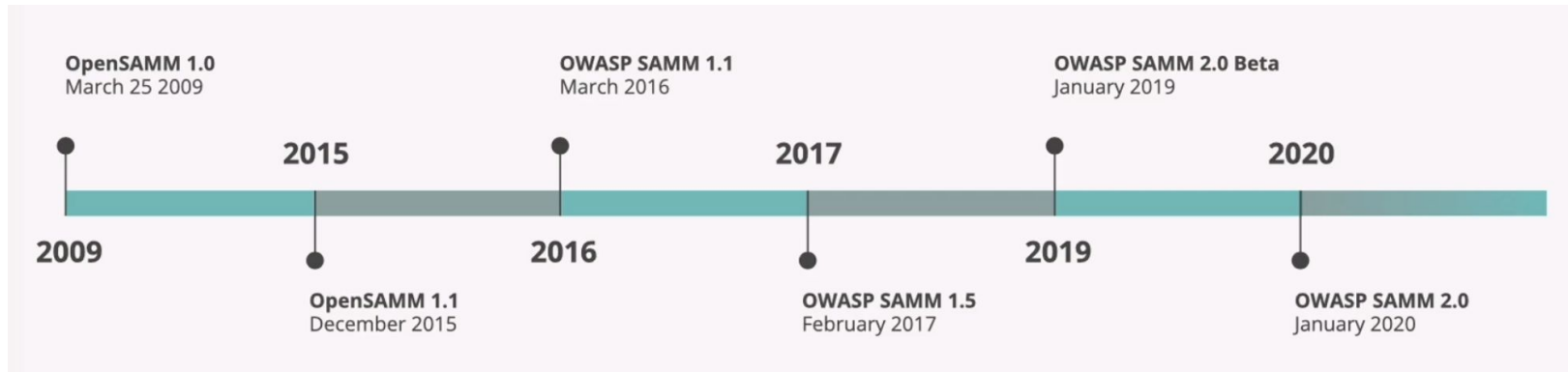
- Governance → Strategy & Metrics, Policy & Compliance, Education and Guidance
- Design → Threat Assessment, Security Requirements, Security Architecture
- Implementation → Secure Build, Secure Deployment, Defect Management
- Verification → Verification, Architecture Assessment, Req. Testing, Sec. Testing
- Operations → Operations, Incident Management, Env. Management, Ops. Management

What is SAMM



SAMM (Software Assurance Maturity Model)

- Community effort.
- Open-source framework helping organizations improve their software security.
- OWASP a flagship project supported by a team of passionate technology and security experts



SAMM Team

- Sebastien Deleersnyder
- Bart De Win
- Maxim Baele
- Aram Hovsepyan
- Nessim Kisserli
- Romuald Szkudlarek
- Daniel Kefer
- John DiLeo
- John Kennedy
- Chris Cooper
- Patricia Duarte
- John Ellingworth
- Brian Glas
- Bruce Jenkins





Helpful features

- Risk-based and helpful to communicate security to staff and business
- Continuous improvement and adaptable
- Helps staff to explain why security is important and everyone's responsibility
- Provides predictability and planning capability
- Evaluation of third-party



SAMM Structure

15 security practices across 5 business functions, each with activities structured into 3 maturity levels. Lower maturity activities are easier to implement and less formalized than those at higher levels.

1. Foundational: The starting point with an unfulfilled security practice
2. Mature: A structured realization with increased efficiency and effectiveness
3. Advanced: A comprehensive mastery of the security practice

Governance

Strategy and Metrics	
Create and promote	Measure and improve
Stream A	Stream B

Policy and Compliance	
Policy & standards	Compliance management
Stream A	Stream B

Education and Guidance	
Training and awareness	Organization and culture
Stream A	Stream B

Design

Threat Assessment	
Application risk profile	Threat modeling
Stream A	Stream B

Security Requirements	
Software requirements	Supplier security
Stream A	Stream B

Secure Architecture	
Architecture design	Technology management
Stream A	Stream B

Implementation

Secure Build	
Build process	Software dependencies
Stream A	Stream B

Secure Deployment	
Deployment process	Secret management
Stream A	Stream B

Defect Management	
Defect tracking	Metrics and feedback
Stream A	Stream B

Verification

Architecture Assessment	
Architecture validation	Architecture mitigation
Stream A	Stream B

Requirements-driven Testing	
Control verification	Misuse/abuse testing
Stream A	Stream B

Security Testing	
Scalable baseline	Deep understanding
Stream A	Stream B

Operations

Incident Management	
Incident detection	Incident response
Stream A	Stream B

Environment Management	
Configuration hardening	Patch and update
Stream A	Stream B

Operational Management	
Data protection	Legacy management
Stream A	Stream B



Maturity Levels

Level 0: Inactive, with no or minimal security practices


Level 1: Initial, with ad-hoc security practices

Level 2: Defined, with documented and increased security practices

Level 3: Mastery, with continuously improved and quantitatively measured security practices



Use case



Analyze security posture (int or ext): Use a maturity measurement tool to assess an organization's current security posture

Identify areas for improvement: Prioritize areas for improvement in an organization's security posture

Establish a baseline: Set a baseline to measure the effectiveness of a security program over time

Build a security assurance program: Use a maturity measurement tool to assess an organization's current security posture

Define and measure security activities: Define and measure security-related activities throughout an organization

Define targets: Set targets for improvement

Define an implementation roadmap: Create a plan for implementation



How can SAMM help (recap)

- Evaluate an organization's existing software security practices
- Build a balanced software security assurance program in well-defined iterations
- Demonstrate concrete improvements to a security assurance program
- Define and measure security-related activities throughout an organization



References

- Join #project-samm in OWASP Slack
- Learn more about SAMM 2 at: <https://owaspsamm.org/resources/training/>
- Check out this free tool using SAMM to manage your application security: <https://sammy.codific.com/>

Questions?

