S3M2

Scalable Software Security Maturity Model

S3M2 – Overview

- The Purple Book Community's Scalable Software Security Maturity Model (S3M2) is a framework designed to help organizations assess and improve their software security practices. It provides a structured approach to measuring and enhancing an organization's maturity in software security, focusing on scalability and community collaboration.
- S3M2 emphasizes scalability and community collaboration, meaning it aims to provide a framework that can be adapted and applied to organizations of different sizes and industries. It also encourages organizations to engage with the software security community, share knowledge, and leverage collective expertise to enhance their security practices.
- The materials from these slides was presented in 3 workshops in the 2023 AppSecCon event on June 29, 2023 as Version .5 of the model to gain community awareness and access to the model. If you are interested in joining the efforts to evolve the model and help with improving software security across the globe, please visit https://www.thepurplebook.club/s3m2

S3M2 – Applying the Model

To begin, download this worksheet to record your results and gain a visual representation of your current state of the software security practices.

S3M2 is broken down into three major categories with a varied number of sub-categories within them, and 5 levels of maturity defined on each category, as defined on Slide 4:

- **People** – Relates to the people aspect of software development organizations and addresses the needs for awareness, training, and Security Champions.
- **Process** – Describes the relative maturity across internal processes to address software security.
- **Technology** – Covers the selection, procurement, and use of software security and DevOps tools to help operate and report on the effectiveness of a software security program.

To use the model, review each of the summary slides (slides 6, 8, 10, and 11) and check off the attributes along each row that best describes the state of each sub category for your software security practices. Review the row and select the column that best represents the state of your program. Record that value (1-5) on the spreadsheet linked on this page.

At the end of your review and transfer of maturity levels you determined, you'll see a Radar Chart that will dynamically change as you add or update data. You can also use this chart for planning future iterations of your program by determining which attributes are needed to advance to the next level. These will serve as a roadmap for improvements to your program.

Purple Book
Community

S3M2: Section Summaries

S3M2 – Maturity Level Overview

Dimension	Level 1 Reactive	Level 2 Proactive	Level 3 Managed	Level 4 Optimized	Level 5 Dynamic
	Basic visibility from Ad Hoc tool execution	Prioritization of remediation efforts, automated tool execution	Processes are defined and policies followed	Processes are optimized and automated	Adaptive AppSec (e.g. Threat modeling)
People					
Process					
Technology					

People

People Section Summary

Dimension	Level 1 Reactive	Level 2 Proactive	Level 3 Managed	Level 4 Optimized	Level 5 Dynamic	Your Rating (1-5)
	Basic Visibility from Ad Hoc tool execution	Prioritization of remediation efforts, automated tool execution	Processes are defined and policies followed	Processes are optimized and automated	Adaptive AppSec (e.g. Threat modeling)	
People / Personas	□ Developer led (volunteer program) □ No dedicated software security resources □ No organizational mandate	□ Dedicated AppSec resources	□ Security Champion program. □ Security champions present on every development team	□ Security champions (if present) Community Formed and Operating	□ Security leads/champions contribute reusable code for remediations □ Best practices are documented for sharing across all development efforts	
Training and Education	□ No program in place □ No mandate from the upper management or leadership	□ Some foundational / introductory training.	□ Role-based training introduced. □ Training program needs to account for the identification of security champions	□ Role-based training refined and metrics collected □ Refresher training introduced. □ Badge of honor issue □ Just-in-Time contextual training. □ Customized CTFs (Capture the Flag)	□ Advanced degree/certification encouraged and sponsored. □ Refresher training expanded.	
Security Champions Program	□ No program in place □ No mandate from the upper management or leadership	□ Ad-hoc appearance of security leads.	□ Security Champions are formalized as part of the program.	□ Community / Network of Security Champions is functioning	□ Active contribution from Security Champions □ Reusable Code, Best Practices and Standards.	
Software Security Awareness	□ No program in place □ No mandate from the upper management or leadership	□ Initial / generic security awareness training rolled out	□ Development of tailor-made, role-specific security awareness	□ Exercises - CTFs □ Cyber-ranges □ Guest speakers from the industry □ Brown bags, □ Regular communication channels established. □ Issuing security advisories / bulletins □ Discussions about security breaches are common	□ The Security Champions become the program!	

Process

Process Section Summary

Dimension	Level 1 Reactive	Level 2 Proactive	Level 3 Managed	Level 4 Optimized	Level 5 Dynamic	Your Rating (1-5)
	Basic Visibility from Ad Hoc tool execution	Prioritization of remediation efforts, automated tool execution	Processes are defined and policies followed	Processes are optimized and automated	Adaptive AppSec (e.g. Threat modeling)	
Governance	❏ Totally reactive, fighting fires	❏ Focus on finding 'low hanging fruit' external attack vulnerabilities	❏ Governance policies (SLAs) and risk methodology defined	❏ Automated gates to control push to production	❏ Data driven decision making process to drive feedback to improve the workflow	
Asset Inventory and Categorization	❏ Inconsistent tracked inventory, limited thought process and planning	❏ Partial Asset Inventory. Effort continues towards automation	❏ Complete portfolio visibility	❏ Partial Asset Inventory. Effort continues towards automation	❏ 100% correlation with asset inventory systems of record and business functions	
Prioritization	❏ Usual approach is to fight the fire, every single time as a snowflake	❏ Prioritization is done using the scanning solution	❏ Internal application's business context is used for prioritization	❏ Prioritization is done using the scanning solution	❏ Ongoing threat modeling drives updates to prioritization	
Remediation	❏ No established strategy, no established guidance	❏ SLAs defined, strategizing remediation activities, not strictly enforced, not universally applied	❏ Established formalized strategy for remediation with rigor and policy compliance	❏ Enforce SLA compliance to drive down MTTR, also leveraging and integration Threat Intel feeds	❏ Ongoing threat modeling drives updates to prioritization	
Security Debt	❏ No visibility into the technical debt	❏ There is a visibility into the technical debt, but the much needed focus does not exist	❏ Visibility, All new debt is managed/under control aka "stop the bleeding"	❏ Visibility, Debt significantly reduced/managed + new under control	❏ acceptance of prioritization of debt reduction as part of backlog	
Metrics	❏ Reporting done on an ad-hoc basis, not an ideal and usually prone to human error	❏ Reporting done on an ad-hoc basis, not an ideal and usually prone to human error	❏ Consolidation of reporting of security posture on a regular basis	❏ Reporting/Dashboarding "on demand" + self service for specific roles	❏ Operationalizing tool selection/optimization/ rationalization	

Purple Book Community

Technology

Technology Tool Summary

Dimension	Level 1 Reactive	Level 2 Proactive	Level 3 Managed	Level 4 Optimized	Level 5 Dynamic	Your Rating (1-5)
	Basic Visibility from Adhoc tool execution	Prioritization of remediation efforts. Automated tool execution.	Processes are defined and policies followed	Processes are optimized and automated	Adaptive AppSec (Threat modelling etc)	
Tool portfolio/ security stack	❑ Tools are Open-Source non-enterprise versions (no paid support, all functions not available) ❑ Usage of SCA tools on high value applications ❑ Usage of SAST on high value apps (no automation) ❑ Start of usage of Code Coverage tools ❑ Lack of Testing tools, lack of testing regime (Ad hoc)	❑ Paid Licenses on primary tools. ❑ Visibility into tools for SCA, SAST, PenTesting, Secrets Detection ❑ Mandated tool sets without App Teams input creating usability issues. ❑ Use of CIS hardening Standards ❑ Use of Security Framework and Libraries ❑ Use of Testing tools, start of formalizing of testing Regime. ❑ Start/planning of formal asset catalog.	❑ Use of Security Frameworks and Libraries ❑ Track Dependency tools for 3rd party code ❑ Infrastructure as code, and automation of deployment ❑ Risk management tool ❑ Automated bug tracking ❑ Tools for tracking dependency in 3rd party code ❑ Tools for Identification/classification of data ❑ Normalization of tools for development/testing	❑ Strict Policy Enforcement ❑ Mix of testing methods ❑ Automation of risk management ❑ Attack Surface Management ❑ Threat Model workflow Standardized ❑ Vulnerabilities/Risk/Findings actively managed	❑ Automated Remediation (SOAR) ❑ Automated policy enforcement ❑ Orchestration of testing/development process	

Technology Secure Design Summary

Dimension	Level 1 Reactive	Level 2 Proactive	Level 3 Managed	Level 4 Optimized	Level 5 Dynamic	Your Rating (1-5)
	Basic Visibility from Adhoc tool execution	**Prioritization of remediation efforts. Automated tool execution.**	**Processes are defined and policies followed**	**Processes are optimized and automated**	**Adaptive AppSec (Threat modelling etc)**	
Secure Design	□ No specific and dedicated secure design practices □ Adhoc threat modeling, if it occurs □ Constrained by Developers knowledge base □ Security requirements reactive	□ Basic threat modelling for critical designs/applications. □ The model document will be any convenient form (data flow diagram with attacks and mitigations, picture of white board, etc.) □ There will typically be significant secure design expertise gaps and lack of coverage.	□ Complex and security critical designs must have a threat model. the security requirements from the model are prioritized and built. □ There is a process for engaging secure design expertise. threat modelling is required for security significant changes. □ KPIs will be introduced. □ Secure design expertise is available at least critical design work. □ Security requirements are validated Training program needs to account for the identification of security champions	□ Trainings occur at predictable periodicity. □ There is a standard model document form and modelling methodology. Governance ensures model quality and completeness. □ Security requirement validation must be included in test regime. □ Modelling is widely adopted (nearly all teams), with modelling and design expertise readily available. □ There are secure design patterns, checklists, or standards. modelling is required for security significant changes. □ There is a risk rating methodology. □ There is a way to protect models and to archive them. □ Models are used as a critical input to design decisions	□ Levels of secure design skill: leaders, practitioners, and those who are learning. □ Every development effort/team practices secure design: threat modelling happens as an organic part of the development process. □ Models go through a governance review to ensure quality. There may be automation applied to modeling. □ Models are documented and available influence security decisions during design sessions. □ Penetration test/bug bounty/test regime findings help refine threat models	