Software Assurance (SwA) Checklist for Software Supply Chain Risk Management

SwA Forum

Processes and Practices Working Group

December 14, 2010

Challenges

- Organizations that are ready to improve their assurance capabilities may not be aware of how to begin an organized security initiative.
- Several maturity models are publicly available, but:
  - Learning curves may inhibit adoption
  - Finding the right model(s) can be time consuming
  - Selecting model components can be complicated
  - Each model has a different approach and level of granularity

Maturity Model Crosswalk

- Performed a model-agnostic analysis of several publicly available maturity models
- Created a consolidated view of current software assurance goals and best practices in the context of an organized SwA initiative
- This consolidated view evolved into the

  SwA Checklist for Software Supply Chain Risk Management

Mapped Maturity Models

- The crosswalk includes mappings between the SwA Checklist practices and practices identified in existing SwA maturity models and related capability maturity models.
- The maturity models mapped within the framework include:
  - Building Security In Maturity Model (BSIMM)
  - Software Engineering Institute (SEI) Capability Maturity Model Integration (CMMI) for Acquisitions
  - OWASP Open Software Assurance Maturity Model (SAMM)
  - SwA Forum Processes and Practices Working Group Assurance Process Reference Model (PRM)
  - CERT Resilience Management Model (RMM)

BSIMM

- Scientific observation-based descriptive model
- Uniquely qualified to be used as a measuring stick for software security
- Based upon analysis of the software security initiatives of 30+ organizations

www.bsimm.com

CMMI for Acquisitions

- CMMI-ACQ provides guidance to acquisition organizations for initiating and managing the acquisition of products and services
- Used to guide process improvement initiatives across a project, a division, or an entire organization.

www.sei.cmu.edu/cmmi/

www.sei.cmu.edu/cmmi/

OpenSAMM

Software Assurance Maturity Model

A guide to building security into software development

VERSION - 1.0

- Open framework to help organizations formulate and implement a strategy for software security that is tailored to the specific risks facing the organization.
- Can be applied organization-wide, for a single line-of-business, or individual projects.

www.opensamm.org

Assurance PRM

- SwA Forum Processes & Practices Working Group synthesized from the contributions of leading government and industry experts.
- Assurance for CMMI® defines the Assurance Thread for Implementation and Improvement of Assurance Practices that are assumed when using the CMMI-DEV.

https://buildsecurityin.us-cert.gov/swa/proself_assm.html

Assurance PRM Self-Assessment

- The Assurance PRM Self-Assessment provides an assessment framework of the implementation of assurance practices
- Incorporates the Assurance PRM goals and practices
- Contains mappings to other freely available maturity models

https://buildsecurityin.us-cert.gov/swa/proself_assm.html

CERT RMM

- Process improvement model
- Addresses the convergence of security, business continuity, and IT operations
- Focus on managing operational risk and establish operational resilience
- Supplies a process improvement approach through the definition and application of a capability level scale

www.cert.org/resilience/rmm.html

www.cert.org

SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

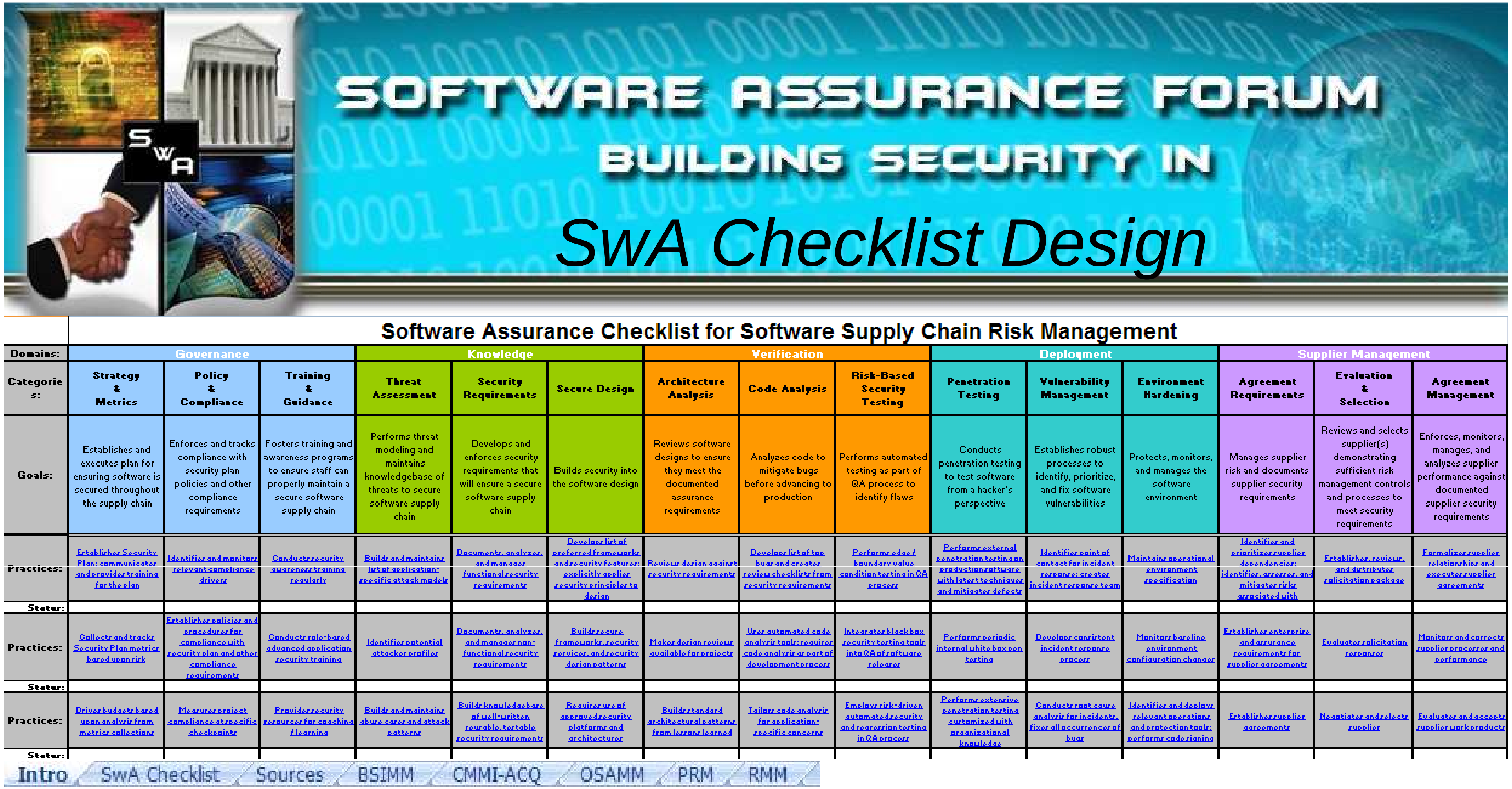
	Governance			Knowledge			Verification			Deployment			Supplier Management		
	Strategy & Metrics	Policy & Compliance	Training & Guidance	Threat Assessment	Security Requirements	Secure Design	Architecture Analysis	Code Analysis	Risk-Based Security Testing	Penetration Testing	Vulnerability Management	Environment Hardening	Agreement Requirements	Evaluation & Selection	Agreement Management
Practices:	Establishes Security Plan; communicates and provides training for the plan	Identifies and monitors relevant compliance drivers	Conducts security awareness training regularly	Builds and maintains list of application-specific attack models	Documents, analyzes, and manages functional security requirements	Develops list of preferred frameworks and security features; explicitly applies security principles to design	Reviews design against security requirements	Develops list of top bugs and creates review checklists from security requirements	Performs edge / boundary value condition testing in QA process	Performs external penetration testing on production software with latest techniques and mitigates	Identifies point of contact for incident response; creates incident response team	Maintains operational environment specification	Identifies and prioritizes supplier dependencies; identifies, assesses, and mitigates risks associated with supplier dependencies	Establishes, reviews, and distributes solicitation package	Formalizes supplier relationships and executes supplier agreement
BSIMM	SM1.1	CP1.1	T1.1	AM1.1	SR1.1	SFD1.1	AA1.1 - AA1.3	CR1.1	ST1.1 - ST1.2	PT1.1 - PT1.2	CMVM2.1	SE1.1	SR3.1	-	-
	-	CP1.2	T3.4	AM1.4	-	SFD1.2	SFD3.1	-	-	-	-	SE1.2	-	-	-
CMMI-ACQ	PP SG2 - SG3	OPF SG1	OT SG2	RSKM SG1 - SG2	ARD SG1, SG3	ATM SG2	ATM SG1	AVER SG3	AVER SG3	AVER SG3	CAR SG1	CM SG2 - SG3	RSKM SG2-SG3	SSAD SG1	AM SG1
	-	-	-	-	REQM SG1	AVAL SG2	AVAL SG1 - SG2	-	-	CAR SG1 - SG2	OPD SG1	-	PP SG1	-	SSAD SG3
OSAMM	SM1B	PC1A	EG1A	TA1A	SR1A	SA1A	DR1B	CR1A	ST2B	ST1B	VM1A	EH1A	-	-	-
	-	PC1B	-	-	SR2B	SA1B	-	-	-	-	VM1B	-	-	-	-
PRM	SG 2.1	SG 3.1	SG 1.3	SG 3.2	SG 3.1	SG 3.2	SG 3.4	SG 3.4	SG 3.4	SG 3.4	SG 4.3	SG 4.3	SG 2.3	SG 2.3	SG 2.3
	SG 1.3	-		-	-	-	-	-	-	-	-	-	SG 3.1	-	-
RMM	RTSE:SG2 - SG3	COMP:SG2	OTA:SG1 - SG2	RISK:SG1 - SG4	RRD:SG1 - SG3	RTSE:SG1 - SG2	-	VAR:SG2	RTSE:SG3	RTSE:SG3	VAR:SG1	ADM:SG3	EXD:SG1 - SG2	EXD:SG3	EXD:SG3
	MON:SG1	MON:SG1 - SG2	-	KIM:SG6	RRM:SG1	KIM:SG2, SG6	-	KIM:SG6	-	-	MON:SG1	KIM:SG5	RISK:SG3 - SG6	-	-
Practices:	Collects and tracks security plan metrics based upon risk	Establishes policies and procedures for compliance with security plan and other compliance requirements	Conducts role-based advanced application security training	Identifies potential attacker profiles	Documents, analyzes, and manages non-functional security requirements	Builds secure frameworks, security services, and security design patterns	Makes design reviews available for projects	Uses automated code analysis tools; requires code analysis as part of development	Integrates black box security testing tools into QA of software releases	Performs periodic internal white box pen testing	Develops consistent incident response process	Monitors baseline environment configuration changes	Establishes enterprise and assurance requirements for supplier agreement	Evaluates solicitation responses	Monitors and corrects supplier processes and performance
BSIMM	SM1.5	CP1.3	T2.1	AM1.3	SR1.3	SFD2.1	AA2.1	CR1.4	ST2.1	PT2.1 - PT2.3	CMVM1.1	SE1.1	SR2.1, SR2.5	-	-
	SM2.1	CP3.2	-	-	-	SFD2.3	AA2.3	CR2.3	-	-	-	-	-	-	-
CMMI-ACQ	MA SG1 - SG2	OPF SG2 - SG3	OT SG2	RSKM SG1 - SG2	ARD SG1, SG3	ATM SG2	AVAL SG1	AVER SG3	AVER SG3	AVER SG3	CAR SG1	CM SG2 - SG3	REQM SG1	SSAD SG2	AM SG1
	PMC SG1	-	-	-	REQM SG1	AVAL SG2	PMC SG1 - SG2	-	-	-	OPD SG1	-	ARD SG2	-	REQM SG1
OSAMM	SM1B	PC2A	EG2A	TA1B	SR1B	SA2A	DR2A	CR2A	ST1B	ST1A	VM2A	EH2B	SR3A	-	-
	-	-	EG3B	-	-	SA2B	DR2B	CR2B	-	ST1B	-	-	-	-	-
PRM	SG 1.1	SG 1.2	SG 1.3	SG 3.2	SG 3.1	SG 3.2	SG 3.4	SG 3.4	SG 3.4	SG 3.4	SG 4.3	SG 4.3	SG 3.1	SG 2.3	SG 2.3
	SG 2.2	-	-	-	-	-	-	-	-	-	-	-	-	-	SG 3.5
RMM	MA:SG2	RTSE:SG2	OTA:SG3 - SG4	RISK:SG1 - SG4	COMP:SG2	RTSE:SG3	-	RTSE:SG3	RTSE:SG3	RTSE:SG3	VAR:SG1	ADM:SG3	EXD:SG3	EXD:SG3	EXD:SG4
	MON:SG2	COMP:SG1	-	KIM:SG6	RRM:SG1	-	-	-	-	-	MON:SG1	KIM:SG5	RRD:SG2 - SG3	-	RRM:SG1
Practices:	Drives budgets based upon analysis from metrics collections	Measures project compliance at specific checkpoints	Provides security resources for coaching / learning	Builds and maintains abuse cases and attack patterns	Builds repository of well written testable and reusable security requirements	Requires use of approved security platforms and architectures	Builds standard architectural patterns from lessons learned	Tailors code analysis for application-specific concerns	Employs risk-driven automated security and regression testing in QA process	Performs extensive penetration testing customized with organizational knowledge	Conducts root cause analysis for incidents, fixes all occurrences of bugs	Identifies and deploys relevant operations and protection tools; performs code signing	Establishes supplier agreement	Negotiates and selects supplier	Evaluates and accepts supplier work products
BSIMM	SM1.5	CP2.3	T1.3 - T1.4	AM2.1	SR1.2	SFD3.2	AA3.2	CR3.1	ST3.1	PT3.1 - PT3.2	CMVM3.1 - 3.2	SE2.3	CP2.4	-	-
	-	CP3.3	T2.4 - T2.5	AM 2.2	SR2.3	-	-	-	-	-	-	-	CP3.2	-	-
CMMI-ACQ	PMC SG2	OPP SG1	OT SG2	RSKM SG2	-	CM SG1	AVAL SG2	AVER SG3	AVER SG3	AVER SG3	CAR SG1 - SG2	OID SG1 - SG2	SSAD SG3	SSAD SG2	AM SG1
	-	-	-	-	-	-	-	-	-	-	-		-		PPQA SG1
OSAMM	SM3A	PC3A	EG1B - EG2B	TA2A	SR2A	SA3A	DR3A	CR3A	ST1A	ST1B	VM3A	EH3A	-	-	-
	SM3B	-	EG3A	-	-	SA3B	-	-	ST2A	-	-	OE3B	-	-	-
PRM	SG 3.1	SG 4.1	SG 1.3	SG 3.1	-	SG 3.2	SG 3.4	SG 3.4	SG 3.4	SG 3.4	SG 4.2	SG 4.3	SG 2.3	SG 2.3	SG 2.3
	-	-	-	-	-	-	-	-	-	-	SG 3.5	-	-	-	-
RMM	RTSE:SG3.SP1	RTSE:SG2	OTA:SG2	RISK:SG1 - SG4	KIM:SG6	KIM:SG2	KIM:SG6	RTSE:SG2	RTSE:SG3	RTSE:SG3	VAR:SG2 - SG4	RISK:SG5	EXD:SG3	EXD:SG3	EXD:SG4
	MON:SG2	COMP:SG3 - SG4	OTA:SG4	KIM:SG6	-	-	-	RTSE:SG3	-	-	MON:SG2	-	-	-	RRM:SG1

Intended Use of the SwA Checklist

- Useful to any organization that is currently or will soon be acquiring or developing software
- Organizations can use the SwA Checklist to:
  - Guide their own development
  - Evaluate vendor capabilities
- Organizations can establish an assurance baseline using the SwA Checklist
- Learn more about current software assurance best practices
- Guide the selection of the most appropriate model components

Design of the SwA Checklist

- Currently implemented as a “hot linked” Microsoft Excel spreadsheet
- Provides a cross-reference of SwA goals and practices with side-by-side mappings to several freely available maturity models
- The consolidated format simplifies identification of the model components best suited for use



SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

SwA Checklist Design

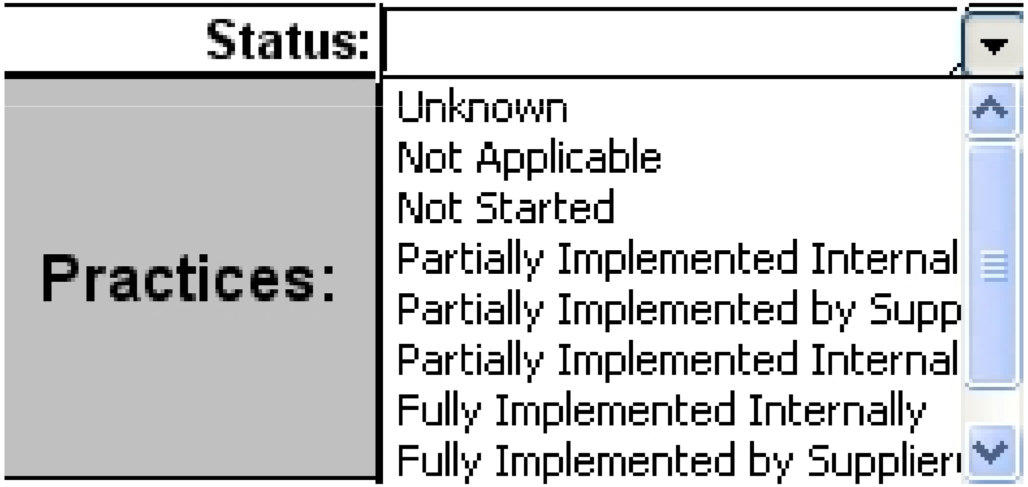
Software Assurance Checklist for Software Supply Chain Risk Management

Domains:	Governance			Knowledge			Verification			Deployment			Supplier Management		
Categories:	Strategy & Metrics	Policy & Compliance	Training & Guidance	Threat Assessment	Security Requirements	Secure Design	Architecture Analysis	Code Analysis	Risk-Based Security Testing	Penetration Testing	Vulnerability Management	Environment Hardening	Agreement Requirements	Evaluation & Selection	Agreement Management
Goals:	Establishes and executes plan for ensuring software is secured throughout the supply chain	Enforces and tracks compliance with security plan policies and other compliance requirements	Fosters training and awareness programs to ensure staff can properly maintain a secure software supply chain	Performs threat modeling and maintains knowledgebase of threats to secure software supply chain	Develops and enforces security requirements that will ensure a secure software supply chain	Builds security into the software design	Reviews software designs to ensure they meet the documented assurance requirements	Analyzes code to mitigate bugs before advancing to production	Performs automated testing as part of QA process to identify flaws	Conducts penetration testing to test software from a hacker's perspective	Establishes robust processes to identify, prioritize, and fix software vulnerabilities	Protects, monitors, and manages the software environment	Manages supplier risk and documents supplier security requirements	Reviews and selects supplier(s) demonstrating sufficient risk management controls and processes to meet security requirements	Enforces, monitors, manages, and analyzes supplier performance against documented supplier security requirements
Practices:	Establishes Security Plan; communicates and provides training for the plan	Identifies and monitors relevant compliance drivers	Conducts security awareness training regularly	Builds and maintains list of application-specific attack models	Documents, analyzes, and manages functional security requirements	Develops list of preferred frameworks and security features; explicitly applies security principles to design	Reviews design against security requirements	Develops list of top bugs and creates review checklists from security requirements	Performs edge/boundary value condition testing in QA process	Performs external penetration testing on production software with latest techniques and mitigates defects	Identifies point of contact for incident response; creates incident response team	Maintains operational environment specification	Identifies and prioritizes supplier dependencies; identifies, assesses, and mitigates risks associated with	Establishes, reviews, and distributes solicitation package	Formalizes supplier relationships and executes supplier agreements
Status:															
Practices:	Collects and tracks Security Plan metrics based upon risk	Establishes policies and procedures for compliance with security plan and other compliance requirements	Conducts role-based advanced application security training	Identifies potential attacker profiles	Documents, analyzes, and manages non-functional security requirements	Builds secure frameworks, security services, and security design patterns	Makes design reviews available for projects	Uses automated code analysis tools; requires code analysis as part of development process	Integrates black box security testing tools into QA of software releases	Performs periodic internal white box pen testing	Develops consistent incident response process	Monitors baseline environment configuration changes	Establishes enterprise and assurance requirements for supplier agreements	Evaluates solicitation responses	Monitors and corrects supplier processes and performance
Status:															
Practices:	Drives budgets based upon analysis from metrics collections	Measures project compliance at specific checkpoints	Provides security resources for coaching / learning	Builds and maintains abuse cases and attack patterns	Builds knowledgebase of well-written reusable, testable security requirements	Requires use of approved security platforms and architectures	Builds standard architectural patterns from lessons learned	Tailors code analysis for application-specific concerns	Employs risk-driven automated security and regression testing in QA process	Performs extensive penetration testing customized with organizational knowledge	Conducts root cause analysis for incidents, fixes all occurrences of bugs	Identifies and deploys relevant operations and protection tools; performs code signing	Establishes supplier agreements	Negotiates and selects supplier	Evaluates and accepts supplier work products
Status:															

- All fields are hyperlinked to specifically related areas in other tabs in the spreadsheet
- This linking allows the user to read how different models address similar assurance goals and practices

Establishing a Baseline

- There is a “Status” cell under each practice in which to select an implementation status.



Status:

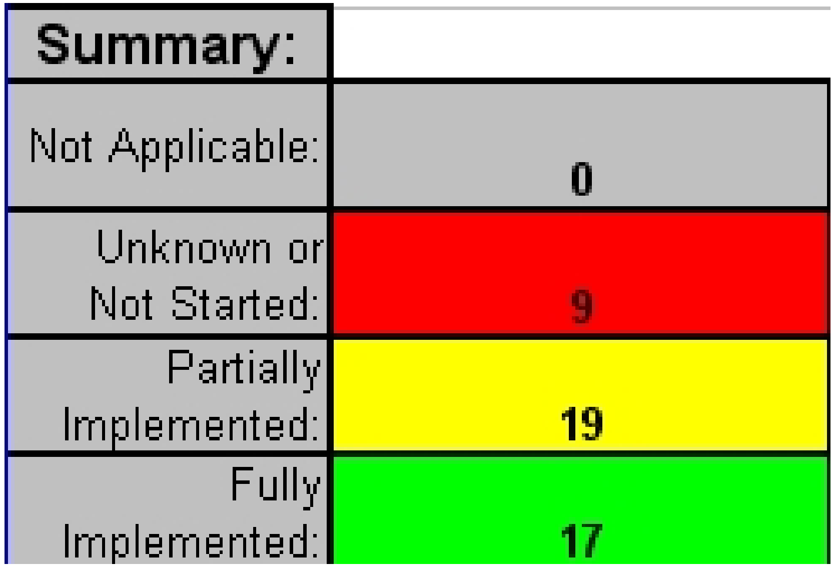
Practices:

Unknown
Not Applicable
Not Started
Partially Implemented Internal
Partially Implemented by Supp
Partially Implemented Internal
Fully Implemented Internally
Fully Implemented by Supplier

- The aggregation of the status of each practice helps organizations understand their ability to execute on software assurance activities.

Baseline Summary

- After establishing a baseline, a summary displays at the bottom
- This system provides an easy-to-view dashboard for an organization's overall implementation of assurance practices



Summary:	
Not Applicable:	0
Unknown or Not Started:	9
Partially Implemented:	19
Fully Implemented:	17

Feedback

- The SwA Checklist is available on the DHS SwA Community Resources and Information Clearinghouse website alongside the Assurance PRM Self-Assessment:

  https://buildsecurityin.us-cert.gov/swa/proself_assm.html

- The Processes and Practices Working Group welcomes feedback on your experiences using the SwA Checklist in the field.

Plans

- The SwA Forum Processes & Practices Working Group plans to add mappings to additional models and update the SwA Checklist as newer versions of mapped models are released.
- CrossTalk journal article on the SwA Checklist in March

Contacts

Ed Wotring

Information Security Solutions, LLC

ed.wotring@informationsecuritysolutionsllc.com

Sammy Migues

Cigital, Inc

smigues@cigital.com