Level 1 Activities

**Governance**

*Strategy & Metrics (SM)*
- Publish process (roles, responsibilities, plan), evolve as necessary. [SM1.1]
- Create evangelism role and perform internal marketing. [SM1.2]
- Educate executives. [SM1.3]
- Identify gate locations, gather necessary artifacts. [SM1.4]

*Compliance & Policy (CP)*
- Unify regulatory pressures. [CP1.1]
- Identify PII obligations. [CP1.2]
- Create policy. [CP1.3]

*Training (T)*
- Provide awareness training. [T1.1]
- Deliver role-specific advanced curriculum (tools, technology stacks, and bug parade). [T1.5]
- Create and use material specific to company history. [T1.6]
- Deliver on-demand individual training. [T1.7]

**Intelligence**

*Attack Models (AM)*
- Create a data classification scheme and inventory. [AM1.2]
- Identify potential attackers. [AM1.3]
- Gather and use attack intelligence. [AM1.5]

*Security Features & Design (SFD)*
- Build and publish security features. [SFD1.1]
- Engage SSG with architecture. [SFD1.2]

*Standards & Requirements (SR)*
- Create security standards. [SR1.1]
- Create a security portal. [SR1.2]
- Translate compliance constraints to requirements. [SR1.3]
SSDL Touchpoints

Architecture Analysis (AA)
- **Perform security feature review.** [AA1.1]
- Perform design review for high-risk applications. [AA1.2]
- Have SSG lead design review efforts. [AA1.3]
- Use a risk questionnaire to rank applications. [AA1.4]

Code Review (CR)
- Have SSG perform ad hoc review. [CR1.2]
- **Use automated tools along with manual review.** [CR1.4]
- Make code review mandatory for all projects. [CR1.5]
- Use centralized reporting to close the knowledge loop and drive training. [CR1.6]

Security Testing (ST)
- **Ensure QA supports edge/boundary value condition testing.** [ST1.1]
- Drive tests with security requirements and security features. [ST1.3]

Deployment

Penetration Testing (PT)
- **Use external penetration testers to find problems.** [PT1.1]
- Feed results to the defect management and mitigation system. [PT1.2]
- Use penetration testing tools internally. [PT1.3]

Software Environment (SE)
- Use application input monitoring. [SE1.1]
- **Ensure host and network security basics are in place.** [SE1.2]

Configuration Management & Vulnerability Management (CMVM)
- Create or interface with incident response. [CMVM1.1]
- **Identify software defects found in operations monitoring and feed them back to development.** [CMVM 1.2]

Level 2 Activities

Governance

Strategy & Metrics (SM)
- Publish data about software security internally. [SM2.1]
- Enforce gates with measurements and track exceptions. [SM2.2]
- Create or grow a satellite. [SM2.3]
- Identify metrics and use them to drive budgets. [SM2.5]
- Require security sign-off. [SM2.6]
Compliance & Policy (CP)
• Identify PII data inventory. [CP2.1]
• Require security sign-off for compliance-related risk. [CP2.2]
• Implement and track controls for compliance. [CP2.3]
• Paper all vendor contracts with software security SLAs. [CP2.4]
• Ensure executive awareness of compliance and privacy obligations. [CP2.5]

Training (T)
• Enhance satellite through training and events. [T2.5]
• Include security resources in onboarding. [T2.6]
• Identify satellite through training. [T2.7]

Intelligence

Attack Models (AM)
• Build attack patterns and abuse cases tied to potential attackers. [AM2.1]
• Create technology-specific attack patterns. [AM2.2]
• Build and maintain a top N possible attacks list. [AM2.5]
• Collect and publish attack stories. [AM2.6]
• Build an internal forum to discuss attacks. [AM2.7]

Security Features & Design (SFD)
• Build secure-by-design middleware frameworks and common libraries. [SFD2.1]
• Create SSG capability to solve difficult design problems. [SFD2.2]

Standards & Requirements (SR)
• Create a standards review board. [SR2.2]
• Create standards for technology stacks. [SR2.3]
• Identify open source. [SR2.4]
• Create a SLA boilerplate. [SR2.5]
• Use secure coding standards. [SR2.6]

SSDL Touchpoints

Architecture Analysis (AA)
• Define and use AA process. [AA2.1]
• Standardize architectural descriptions (including data flow). [AA2.2]
• Make SSG available as AA resource or mentor. [AA2.3]

Code Review (CR)
• Assign tool mentors. [CR2.5]
• Use automated tools with tailored rules. [CR2.6]
• Use a top N bugs list (real data preferred). [CR2.7]

Security Testing (ST)
• Integrate black box security tools into the QA process. [ST2.1]
• Share security results with QA. [ST2.4]
• Include security tests in QA automation. [ST2.5]
• Perform fuzz testing customized to application APIs. [ST2.6]
Deployment

**Penetration Testing (PT)**
- Provide penetration testers with all available information. [PT2.2]
- Schedule periodic penetration tests for application coverage. [PT2.3]

**Software Environment (SE)**
- Publish installation guides. [SE2.2]
- Use code signing. [SE2.4]

**Configuration Management & Vulnerability Management (CMVM)**
- Have emergency codebase response. [CMVM2.1]
- Track software bugs found in operations through the fix process. [CMVM2.2]
- Develop an operations inventory of applications. [CMVM2.3]

Level 3 Activities

**Governance**

**Strategy & Metrics (SM)**
- Use an internal tracking application with portfolio view. [SM3.1]
- Run an external marketing program. [SM3.2]

**Compliance & Policy (CP)**
- Create regulator eye candy. [CP3.1]
- Impose policy on vendors. [CP3.2]
- Drive feedback from SSDL data back to policy. [CP3.3]

**Training (T)**
- Reward progression through curriculum (certification or HR). [T3.1]
- Provide training for vendors or outsourced workers. [T3.2]
- Host external software security events. [T3.3]
- Require an annual refresher. [T3.4]
- Establish SSG office hours. [T3.5]

**Intelligence**

**Attack Models (AM)**
- Have a science team that develops new attack methods. [AM3.1]
- Create and use automation to do what attackers will do. [AM3.2]

**Security Features & Design (SFD)**
- Form a review board or central committee to approve and maintain secure design patterns. [SFD 3.1]
- Require use of approved security features and frameworks. [SFD3.2]
- Find and publish mature design patterns from the organization. [SFD3.3]

**Standards & Requirements (SR)**
- Control open source risk. [SR3.1]
- Communicate standards to vendors. [SR3.2]
SSDL Touchpoints

Architecture Analysis (AA)
- Have software architects lead design review efforts. [AA3.1]
- Drive analysis results into standard architecture patterns. [AA3.2]

Code Review (CR)
- Build a factory. [CR3.2]
- Build a capability for eradicating specific bugs from the entire codebase. [CR3.3]
- Automate malicious code detection. [CR3.4]
- Enforce coding standards. [CR3.5]

Security Testing (ST)
- Drive tests with risk analysis results. [ST3.3]
- Leverage coverage analysis. [ST3.4]
- Begin to build and apply adversarial security tests (abuse cases). [ST3.5]

Deployment

Penetration Testing (PT)
- Use external penetration testers to perform deep-dive analysis. [PT3.1]
- Have the SSG customize penetration testing tools and scripts. [PT3.2]

Software Environment (SE)
- Use code protection. [SE3.2]
- Use application behavior monitoring and diagnostics. [SE3.3]
- Use application containers. [SE3.4]

Configuration Management & Vulnerability Management (CMVM)
- Fix all occurrences of software bugs found in operations. [CMVM3.1]
- Enhance the SSDL to prevent software bugs found in operations. [CMVM3.2]
- Simulate software crisis. [CMVM3.3]
- Operate a bug bounty program. [CMVM3.4]
Interested in joining the growing BSIMM Community?

Go to www.BSIMM.com