BSIMM (Building Security In Maturity Model)

Framework Checklist

Version: 0.8
Author: Ivan Piskunov
Date: 2025
BSIMM Version: BSIMM15 (2025)

Document Preface

This checklist provides a structured overview of key activities and practices from the Building Security In Maturity Model (BSIMM) framework. It is designed to help security professionals, development teams, and program leads quickly reference and implement BSIMM's data-driven software security practices based on real-world observations from hundreds of organizations .

Intended Audience: AppSec Engineers, DevSecOps Teams, Security Champions, Team Leads, CISOs, BISOs.

Official Reference: BSIMM Website

Disclaimer: This checklist is derived from public BSIMM materials and is not an official Synopsys product. It represents a condensed reference guide based on published BSIMM content. For official assessments and detailed guidance, please contact Synopsys or authorized partners.

Table of Contents

1. Governance Domain Checklist

Focus: Organizing, managing, and measuring software security initiatives .

Strategy & Metrics

- [ ] **Define software security goals** aligned with business objectives
- [ ] **Establish and track security metrics** (e.g., # of assessments, MTTR, test coverage)
- [ ] **Create evangelism/internal marketing role** to build organizational support
- [ ] **Develop formal software security strategy** documented and communicated across organization
- [ ] **Assign dedicated Software Security Group (SSG)** with clear responsibilities

Compliance & Policy

- [ ] **Create security policies** addressing regulatory requirements (PCI DSS, HIPAA, etc.)
- [ ] **Establish third-party risk management** process for software suppliers
- [ ] **Develop unified compliance approach** for customer security requirements
- [ ] **Implement policy exception process** with risk assessment and approval workflow
- [ ] **Conduct regular policy reviews** and updates based on changing threats and regulations

Training

- [ ] **Provide security awareness training** for all development personnel
- [ ] **Develop role-specific security training** for developers, architects, testers
- [ ] **Incorporate company-specific attack stories** into training materials
- [ ] **Establish security champions program** across development teams
- [ ] **Create ongoing training curriculum** updated with emerging threats and technologies

---

2. Intelligence Domain Checklist

Focus: Building corporate knowledge to inform security activities .

Attack Models

- [ ] **Collect and publish attack stories** based on organizational history
- [ ] **Create and maintain catalog of attack patterns** relevant to organization
- [ ] **Perform threat modeling** during design phases of development

- [ ] **Identify potential attack vectors** for critical applications and data
- [ ] **Integrate threat intelligence** into security activities and planning

Security Features & Design

- [ ] **Build/publish reusable security components** (authn, authz, crypto, logging)
- [ ] **Define secure design principles** and patterns for development teams
- [ ] **Create security reference architectures** for common application types
- [ ] **Establish secure design review process** for architecture and design phases
- [ ] **Develop security patterns** for cloud-native and microservices architectures

Standards & Requirements

- [ ] **Create security standards** including secure coding standards
- [ ] **Define security requirements** for software projects
- [ ] **Identify open source usage** and establish risk management process
- [ ] **Develop security requirements framework** aligned with business risk
- [ ] **Maintain security knowledge base** accessible to development teams

---

3. SSDL Touchpoints Domain Checklist

Focus: Integrating security activities into software development lifecycle .

Architecture Analysis

- [ ] **Have SSG lead architecture review efforts** for critical applications
- [ ] **Perform security architecture analysis** during design phases
- [ ] **Integrate security considerations** into architecture governance
- [ ] **Establish risk-based review process** prioritizing high-impact applications
- [ ] **Develop architecture assessment criteria** covering security controls and patterns

Code Review

- [ ] **Adopt SAST tools** alongside manual reviews
- [ ] **Perform manual security code reviews** for high-risk code
- [ ] **Create tailored rule sets** for SAST tools based on organizational risk
- [ ] **Track and address security-related defects** in bug tracking system
- [ ] **Implement automated malicious code detection**

Security Testing

- [ ] **Integrate black box security tools** into QA process
- [ ] **Perform dynamic security testing** (DAST) for applications
- [ ] **Conduct fuzz testing** for protocols and interfaces
- [ ] **Implement security testing in CI/CD** pipelines

- [ ] **Develop automated security test cases** for critical security controls

---

4. Deployment Domain Checklist

Focus: Securing operational environments and managing vulnerabilities .

Penetration Testing

- [ ] **Use external pen testers** to find problems
- [ ] **Conduct regular penetration tests** of applications and environments
- [ ] **Perform red team exercises** to test defense capabilities
- [ ] **Establish remediation process** for penetration test findings
- [ ] **Integrate pen test findings** into security improvement plans

Software Environment

- [ ] **Ensure host/network security basics** in place
- [ ] **Harden runtime environments** for applications
- [ ] **Implement secure configuration management**
- [ ] **Monitor applications for security events**
- [ ] **Establish container security controls** for cloud-native deployments

Configuration Management & Vulnerability Management

- [ ] **Identify software bugs found in ops monitoring** and feed back to development
- [ ] **Maintain inventory of applications and dependencies**
- [ ] **Implement patch management process** for applications and components
- [ ] **Conduct regular vulnerability assessments**
- [ ] **Establish software bill of materials (SBOM)** process

---

5. Implementation Guidance

Maturity Levels

BSIMM activities are typically categorized into three maturity levels :

1. **Level 1 (Emerging):** Basic practices implemented inconsistently
2. **Level 2 (Maturing):** Practices formalized and implemented more consistently
3. **Level 3 (Optimizing):** Practices refined, measured, and continuously improved

Assessment Process

Official BSIMM assessment typically involves :

1. **Data collection** through interviews with key personnel
2. **Activity identification** against BSIMM framework
3. **Scoring** based on implementation of activities
4. **Benchmarking** against industry peers
5. **Reporting** with recommendations for improvement

Additional Resources

- **Official BSIMM Website:** https://www.bsimm.com
- **Latest BSIMM Report:** Download from official website
- **BSIMM Community:** Participate in community activities and benchmarking
- **Maturity Action Plan (MAP):** Develop plan for implementing improvements

---

Document Version History:

- Version 0.8 (2025): Initial checklist created by Ivan Piskunov

Acknowledgements: This checklist is based on publicly available BSIMM materials from Synopsys and the BSIMM community. Special thanks to all organizations that contribute data to the BSIMM study.