W10 Application Security & Assessment

Here is a comprehensive, well-structured Markdown guide covering:

- Static Application Security Testing (SAST)
- Dynamic Application Security Testing (DAST)
- Technology Security & Risk Assessment
- Product Security & Risk Assessment

This guide includes key concepts, tools, assessment processes, common risks, mitigation strategies, and relevant standards/frameworks.

🛡️ Enterprise Security Testing & Risk Assessment Guide

---

1. 🧩 Static Application Security Testing (SAST)

🔍 Overview

- **Definition**: White-box testing method analyzing source code, bytecode, or binaries without executing the program.
- **Purpose**: Identify security vulnerabilities early in the Software Development Life Cycle (SDLC).
- **Benefits**:
  - Early detection of vulnerabilities.
  - Integration into CI/CD pipelines.
  - Compliance with security standards.(Jit, OX Security)

🛠️ Tools

- SonarQube
- Fortify Static Code Analyzer
- Checkmarx
- Veracode
- CodeQL(Investopedia, LeanIX, DevTools)

🔄 Assessment Process

1. Integrate SAST tools into development workflows.
2. Configure rulesets based on coding standards.
3. Perform automated scans during code commits or builds.

4. Analyze results and prioritize findings.
5. Remediate identified vulnerabilities.(Reuters)

⚠️ Common Risks Detected

- SQL Injection
- Cross-Site Scripting (XSS)
- Hardcoded credentials
- Buffer overflows
- Insecure API usage(Secureframe, WSJ, Hyperproof)

🛡️ Mitigation Strategies

- Adopt secure coding practices.
- Regular code reviews.
- Developer training on security principles.
- Use of coding standards (e.g., OWASP Secure Coding Guidelines).(U.S. Department of Education, Bright Security, Black Duck)

📚 Standards & Frameworks

- OWASP Top 10
- ISO/IEC 27001:2022 (A.5.24 Secure coding)
- NIST SP 800-218 (Secure Software Development Framework)(OX Security)

---

2. 🌐 Dynamic Application Security Testing (DAST)

🔍 Overview

- **Definition**: Black-box testing method that analyzes applications in their running state to identify vulnerabilities.
- **Purpose**: Simulate real-world attacks to uncover security issues in live applications.
- **Benefits**:
  - Identifies runtime vulnerabilities.
  - No access to source code required.
  - Complements SAST by covering different vulnerability types.(BreachLock)

🛠️ Tools

- OWASP ZAP
- Burp Suite
- Acunetix
- Netsparker

- IBM AppScan

🔄 Assessment Process

1. Set up testing environment mirroring production.
2. Configure DAST tools with target URLs and authentication details.
3. Perform automated scans to detect vulnerabilities.
4. Analyze results and validate findings.
5. Implement fixes and retest.(Black Duck)

⚠️ Common Risks Detected

- Cross-Site Scripting (XSS)
- SQL Injection
- Security misconfigurations
- Authentication and session management issues
- Sensitive data exposure

🛡️ Mitigation Strategies

- Implement input validation and output encoding.
- Use secure authentication mechanisms.
- Configure security headers properly.
- Regularly update and patch systems.

📚 Standards & Frameworks

- OWASP Top 10
- NIST SP 800-115 (Technical Guide to Information Security Testing and Assessment)
- PCI DSS Requirement 11.3 (Vulnerability scanning)

---

3. 🏗️ Technology Security & Risk Assessment

🔍 Overview

- **Definition**: Systematic evaluation of an organization's technology infrastructure to identify and mitigate security risks.
- **Purpose**: Ensure the confidentiality, integrity, and availability of technological assets.
- **Benefits**:
  - Proactive risk identification.
  - Informed decision-making for security investments.
  - Compliance with regulatory requirements.

🛠️ Tools

- Nessus
- Qualys
- OpenVAS
- Nmap
- Shodan(Reuters, SentinelOne)

🔄 Assessment Process

1. Asset identification and classification.
2. Threat and vulnerability analysis.
3. Risk evaluation and prioritization.
4. Implementation of mitigation strategies.
5. Continuous monitoring and review.(NIST Publications)

⚠️ Common Risks

- Unpatched software vulnerabilities.
- Misconfigured systems and networks.
- Inadequate access controls.
- Lack of encryption for sensitive data.

🛡️ Mitigation Strategies

- Regular patch management.
- Implementing least privilege access.
- Network segmentation.
- Encryption of data at rest and in transit.(LeanIX)

📚 Standards & Frameworks

- NIST Cybersecurity Framework (CSF)
- ISO/IEC 27001:2022
- CIS Controls v8

---

4. 🧪 Product Security & Risk Assessment

🔍 Overview

- **Definition**: Evaluation of a product's security posture throughout its lifecycle, from design to deployment.
- **Purpose**: Identify and mitigate security risks in products to protect users and data.
- **Benefits**:
  - Enhanced product security.

- Reduced risk of breaches.
- Improved customer trust.(DevTools, SentinelOne, Investopedia)

🛠️ Tools

- Threat modeling tools (e.g., Microsoft Threat Modeling Tool)
- Dependency scanning tools (e.g., Snyk, OWASP Dependency-Check)
- Security testing tools (e.g., SAST and DAST tools)
- Vulnerability management platforms(Investopedia)

🔄 Assessment Process

1. Conduct threat modeling during design phase.
2. Perform security testing (SAST/DAST) during development.
3. Assess third-party components for vulnerabilities.
4. Implement security controls and best practices.
5. Monitor and respond to security incidents post-deployment.

⚠️ Common Risks

- Use of vulnerable third-party libraries.
- Insecure default configurations.
- Lack of secure update mechanisms.
- Insufficient logging and monitoring.(WSJ, LeanIX)

🛡️ Mitigation Strategies

- Maintain a Software Bill of Materials (SBOM).
- Regularly update and patch components.
- Implement secure coding standards.
- Establish incident response plans.(Reuters)

📚 Standards & Frameworks

- OWASP Application Security Verification Standard (ASVS)
- ISO/IEC 27034 (Application Security)
- NIST Secure Software Development Framework (SSDF)

This guide serves as a foundational resource for enterprise security teams to implement and enhance their security testing and risk assessment practices.
