Application Security Testing Policy and Procedures

1. Purpose

This document defines the organization's requirements, criteria, and processes for conducting application security testing and reviews. The goal is to ensure that all applications meet stringent security standards before deployment and throughout their lifecycle.

---

2. Scope

This policy applies to all applications developed, maintained, or used by DocuPanda, including internal and third-party software. It covers all stages of the software lifecycle, from development to deployment and maintenance.

---

3. Application Security Testing Requirements

3.1 Types of Security Testing

All applications must undergo the following types of security testing:

1. **Static Application Security Testing (SAST)**
    - Performed during the development phase.
    - Analyzes source code for vulnerabilities (e.g., insecure coding patterns, hardcoded credentials).
2. **Dynamic Application Security Testing (DAST)**
    - Conducted on running applications.
    - Identifies vulnerabilities in runtime behavior (e.g., SQL injection, XSS).
3. **Interactive Application Security Testing (IAST)**
    - Combines SAST and DAST.
    - Detects vulnerabilities in real-time during testing or manual QA.
4. **Penetration Testing**
    - Simulates real-world attacks.
    - Identifies vulnerabilities beyond automated testing capabilities.

5. **Third-Party Dependency Scanning**
    - Scans external libraries and frameworks for known vulnerabilities.
6. **Configuration and Infrastructure Security Testing**
    - Ensures that application environments (e.g., cloud configurations, APIs) are securely configured.

---

4. Testing Criteria

4.1 Risk-Based Testing

The level and frequency of security testing are determined based on the application's risk level:

- **High-Risk Applications**
    - Critical systems handling sensitive data (e.g., personal user data, financial information).
    - Require comprehensive testing, including SAST, DAST, and penetration testing for every major release.
- **Medium-Risk Applications**
    - Applications with moderate sensitivity (e.g., internal tools with limited sensitive data).
    - Require SAST and DAST for each release and annual penetration tests.
- **Low-Risk Applications**
    - Applications that handle no sensitive data.
    - Require basic SAST and DAST for each release.

4.2 Pass/Fail Criteria

- No critical or high vulnerabilities should exist in production.
- Medium vulnerabilities must be addressed or have compensating controls in place.
- Low vulnerabilities should be documented and fixed based on priority.

---

5. Security Review Process

5.1 Pre-Development

1. **Security Requirements Definition**
    - Security requirements must be defined in the initial project phase.
    - Include access control, data protection, and audit logging.
2. **Threat Modeling**
    - Identify potential threats and define mitigation strategies.

5.2 During Development

1. **Secure Code Reviews**
    - Conduct peer reviews with a focus on security.
    - Use automated SAST tools to identify vulnerabilities early.
2. **Automated Security Testing**
    - Integrate security testing tools (SAST/DAST) into the CI/CD pipeline.
    - Ensure vulnerabilities are fixed before code merges.

5.3 Pre-Deployment

1. **Penetration Testing**
    - Conduct manual penetration testing on pre-production environments.
    - Verify that no critical or high vulnerabilities exist.
2. **Security Configuration Review**
    - Verify secure configuration of production environments.
    - Ensure adherence to secure deployment practices.

5.4 Post-Deployment

1. **Continuous Monitoring and Testing**
    - Regularly scan for new vulnerabilities using automated tools.
    - Conduct periodic penetration tests (at least annually for high-risk applications).
2. **Patch and Update Management**

- Apply patches to fix vulnerabilities promptly.
- Re-test after applying patches to ensure no regression.

---

6. Documentation and Reporting

6.1 Test Reports

- Document all findings from security tests, including:
  - Vulnerability description.
  - Risk level (critical, high, medium, low).
  - Remediation steps.

6.2 Remediation Tracking

- Track the status of vulnerabilities through issue tracking systems.
- Ensure timely remediation based on the risk level.

6.3 Audit Logs

- Maintain logs of all security testing activities.
- Logs should include details such as test type, date, tester, and findings.

---

7. Roles and Responsibilities

7.1 Security Team

- Own the security testing process.
- Provide tools and frameworks for security testing.
- Conduct manual penetration testing and audits.

7.2 Developers

- Write secure code following industry best practices.
- Address vulnerabilities identified during security testing.

7.3 QA Engineers

- Incorporate security tests into the QA process.
- Validate fixes for identified vulnerabilities.

7.4 Product Managers

- Ensure security requirements are prioritized and addressed.
- Ensure applications pass security reviews before deployment.

---

8. Tools and Resources

8.1 Recommended Tools

- **SAST**: SonarQube, Checkmarx
- **DAST**: OWASP ZAP, Burp Suite
- **Penetration Testing**: Metasploit, Kali Linux
- **Dependency Scanning**: Snyk, Dependabot
- **CI/CD Integration**: GitHub Actions, Jenkins

---

9. Policy Compliance

- All applications must comply with this policy.
- Non-compliance will result in immediate review and potential suspension of the application's development or deployment until compliance is achieved.

---

10. Policy Review

This policy will be reviewed annually or as necessary to ensure effectiveness and alignment with evolving security standards.

Approved By: Uri Merhav (CIO) **Effective Date**: 10/01/2024 **Next Review Date**: 01/01/2025