OWASP ALGIERS

Building a Secure Software Development Lifecycle (S-SDLC)

20/04/2024

OWASP ALGIERS

SPEAKER

Yasmine AZZAZ

Board Member @ OWASP Algiers

- Board Member @ OWASP Algiers Chapter
- Cyber Security Specialist
- Author of CVE-2018-1999005
- VAPT Specialist & Cybersecurity Solutions Administrator

Agenda

- Introduction to Software Development Lifecycle – SDLC
- Importance of SDLC in Software Development
- SDLC Phases Overview
- Security In The Software Development Lifecycle
- Integrating Security Throughout SDLC Phases
- Secure SDLC Methodologies
- Resources

Introduction to Software Development Lifecycle

What is SDLC ?

- SDLC is a project life cycle specifically designed for the creation, alteration, maintenance and improvement of software.
- It is a structured and methodical process used by software development teams to design, develop, test, and deploy high-quality and low-cost software products, in the shortest possible production time.

Importance of SDLC in Software Development

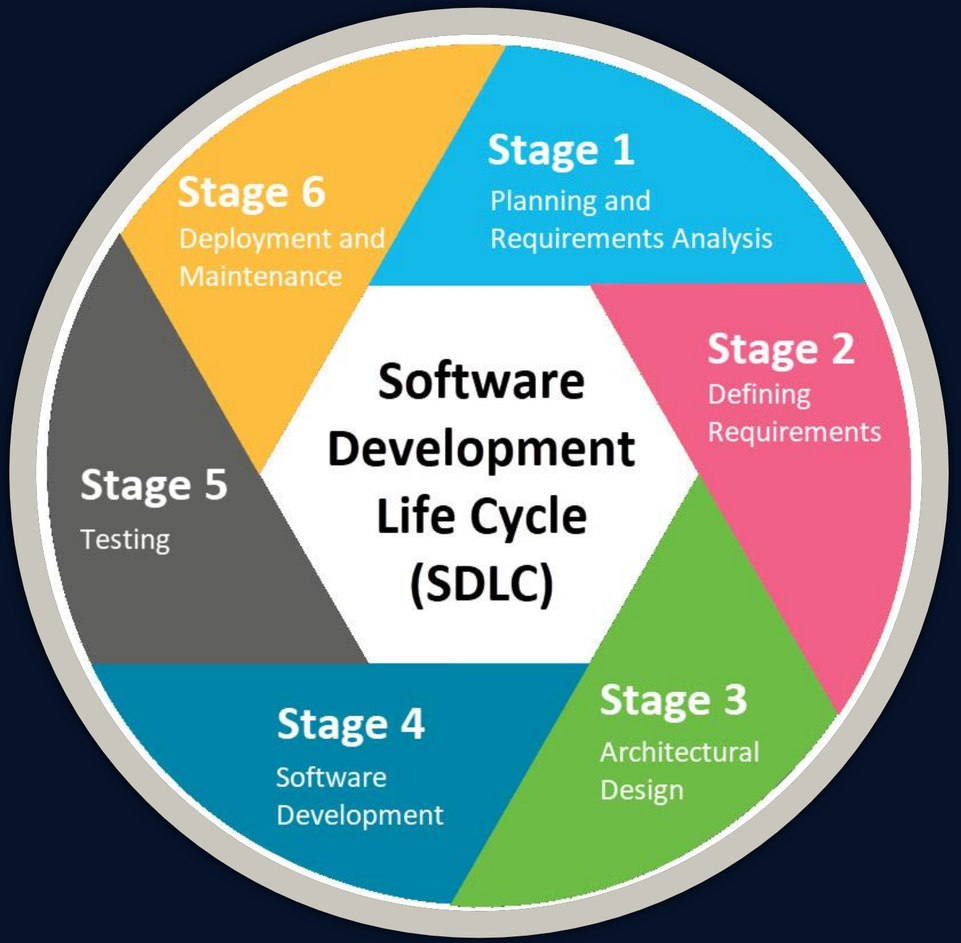
Why is SDLC Important?

- ✓ Structured approach of software development.
- ✓ Better project management (project timeline, budget, and resources).
- ✓ High-quality software that meets all customer expectations.
- ✓ Standardized framework.
- ✓ Reduce costs and risk of errors.
- ✓ Facilitates continuous improvement.

SDLC Phases Overview

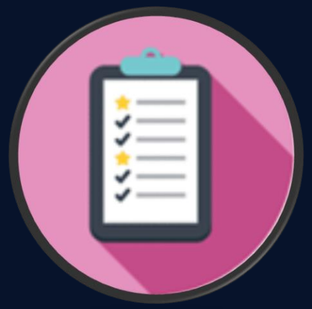
How does SDLC work?

The SDLC process consists essentially of the following phases



SDLC Phases

Planning



- ✓ Define the project scope and objectives.
- ✓ Discuss and understand client needs and expectations.
- ✓ Discuss application key elements.

Defining Requirements

- ✓ Collect and document all functional and non-functional project requirements.
- ✓ Requirements review and approval. Feasibility studies.
- ✓ Build Software Requirement Specification Document (SRS).

Design

- ✓ Create a detailed design of the software architecture, including components, modules, and interfaces based on the SRS document.
- ✓ High-level design (HLD) to describe software architecture.
- ✓ Low-Level Design (LLD) to explain how each product feature and component works.

OWASP ALGIERS

SDLC Phases

Development (Coding)

- ✓ Implementation of the software product.
- ✓ Developers develop the software according to design specifications.
- ✓ Developers build the system by writing programming codes using different programming languages.

Testing

- ✓ Verify that the software works and gives the result as per the requirements addressed in the requirement phase.
- ✓ Test and verify software functionality and identify errors and bugs based on test plan.

Deployment & Maintenance

- ✓ Deploy the software to the production environment.
- ✓ Release regular updates with enhanced features.
- ✓ Fix errors and bugs discovered on the production application.

OWASP ALGIERS

Security in the Software Development Lifecycle (S-SDLC)

What is Secure SDLC?

- Secure SDLC involves integrating security activities throughout the entire software development process.

- Including security into all SDLC phases help to build robust, secure, and resilient software that meets user's need while protecting against evolving cyber threats.

Why is Security Important in the SDLC?

- ✓ Improve software security through early detection and mitigation of vulnerabilities.
- ✓ Reduce software development costs.
- ✓ Compliance with privacy and security regulations.
- ✓ Prevent Security Incidents. Reduce data breaches risk, malware infections, etc…
- ✓ Build and maintain customer trust.
- ✓ Increase collaboration between all stakeholders.

Integrating Security Throughout SDLC Phases

Security In Planning Phase

- ✓ Stakeholders alignment on security considerations to establish a strong foundation for the development process.
- ✓ Consider key aspects of the project including project scope, plan, cost, timelines, targets…
- ✓ Define Security objectives and plan.

Security In Defining Requirements Phase

Define security requirements for system and information security

Risk Assessment

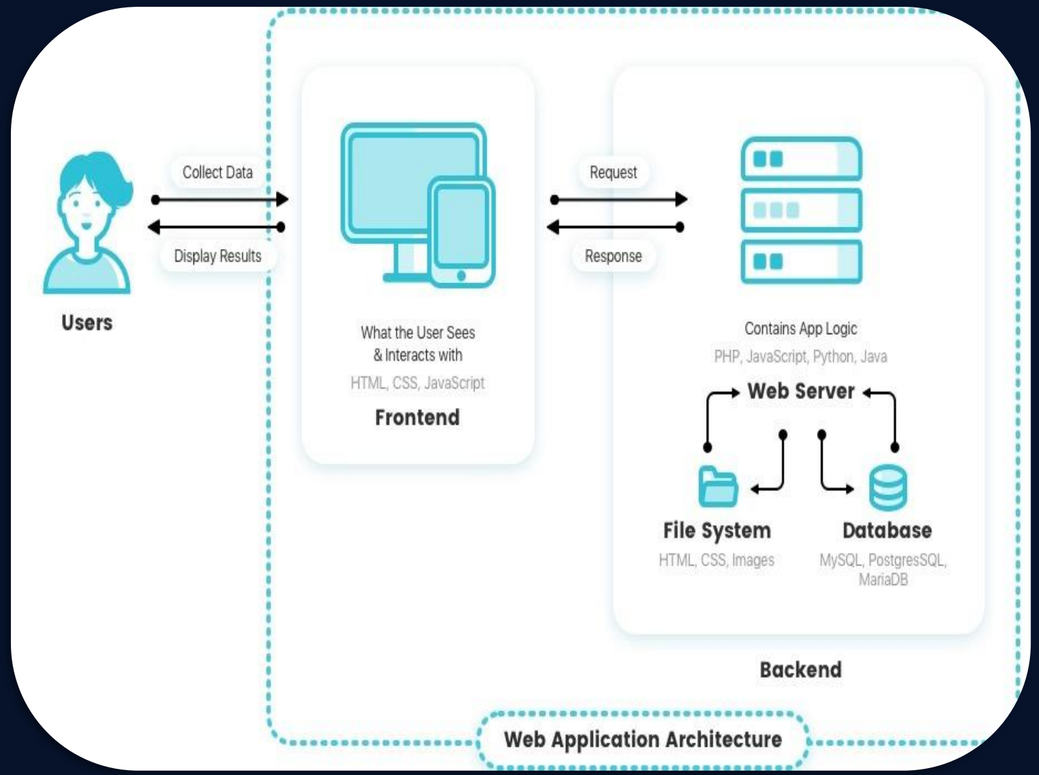
Validate requirements. Accept secure framework, languages, and technologies

Include Compliance requirements

OWASP ALGIERS

Security In Design Phase

- Secure architecture.
- Secure communication and interaction between application components and with external systems.
- Security controls implementation (access controls, authentication mechanisms, encryption, data integrity checks,…).



“The cost of removing an application security vulnerability during the design phase ranges from 30-60 times less than if removed during production.” NIST

Security In Development (CODING) Phase

- ✓ Indicate secure coding practices and guidelines (input validation implementation, proper error handling,…)
- ✓ Include Static Application Security Testing Tool (SAST) during development for real-time code review and security checks.
- ✓ Assist developer to mitigate detected vulnerabilities.

Security In Testing Phase

Perform full application testing to confirm the application of security requirements, identify and fix security weaknesses and vulnerabilities.

Some types of security testing

OS Security check

Conduct vulnerability scanning to identify vulnerable versions and apply patches

Tools :
Qualys, Tenable, Rapid7…

Configuration check

Review technologies configuration according to best practices

References :
CIS Benchmark

Application security test

Manual and automatic application assessment following the OWASP guide (Open Worldwide Application Security Project)

Manual Tools :
BurpSuite, OWASP ZAP.

Automatic Tools:
Dynamic application security testing tool (DAST).

Security In Deployment & Maintenance Phase

✓ Conduct a security checks of the deployment environment and implement appropriate controls.

✓ Monitor and manage security incidents and events.

✓ Perform regular security audits and assessments and apply patches.

SAST vs DAST?

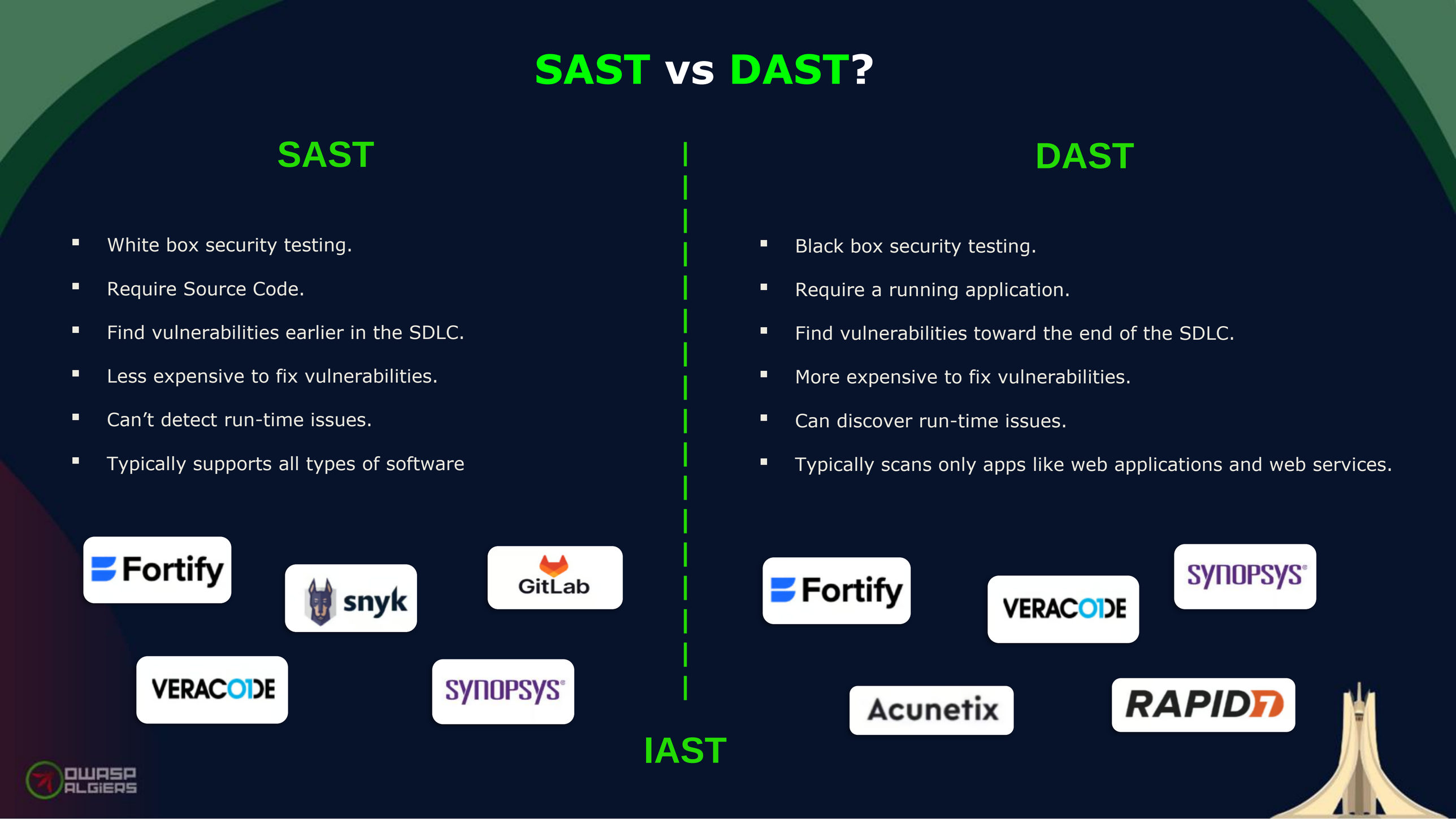
SAST

- White box security testing.
- Require Source Code.
- Find vulnerabilities earlier in the SDLC.
- Less expensive to fix vulnerabilities.
- Can’t detect run-time issues.
- Typically supports all types of software

DAST

- Black box security testing.
- Require a running application.
- Find vulnerabilities toward the end of the SDLC.
- More expensive to fix vulnerabilities.
- Can discover run-time issues.
- Typically scans only apps like web applications and web services.

IAST



OWASP - Security Testing Guides

(Open Worldwide Application Security Project)

OWASP Web Security Testing Guide (WSTG)

Provides various methodologies and techniques for evaluating and improving the web applications security posture.

OWASP Mobile Security Testing Guide (MSTG)

Guide for mobile applications security. It offers advice and best practices for testing the security of both iOS and Android applications.

API Security Testing Guide (ASTG)

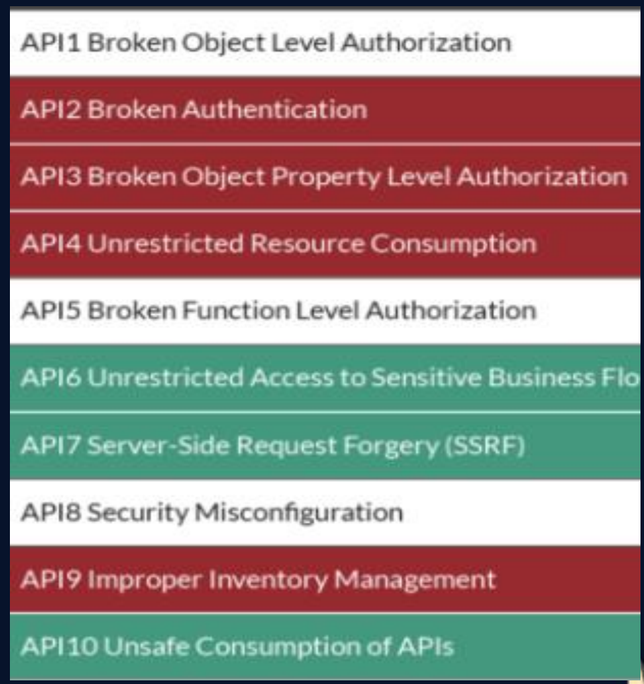
Guide designed to assist security professionals, developers, and testers in evaluating the security posture of APIs (Application Programming Interfaces).

OWASP – Top 10

Web application security
OWASP top 10

Mobile application Security
OWASP top 10

API Security
OWASP top 10



OWASP ALGIERS

Secure SDLC Methodologies

Popular Secure SDLC Methodologies

Popular frameworks used to establish secure software development lifecycles :

- NIST Secure Software Development Framework (SSDF)
- MS Security Development Lifecycle (MS SDL)
- OWASP Application Security Verification Standard (ASVS)
- OWASP Software Assurance Maturity Model (SAMM)

OWASP (ASVS)

The OWASP Application Security Verification Standard (ASVS) is a framework designed to establish a set of security requirements and guidelines for designing, building, developing, and testing web applications.

ASVS covers a wide range of security topics.

OWASP Application Security Verification Standard (ASVS)

- V1 - Architecture, Design and Threat Modeling
- V2 - Authentication
- V3 - Session management
- V4 - Access Controls
- V5 - Validations, Sanitization and Encoding
- V6 - Stored Cryptography
- V7 - Error Handling and Logging
- V8 - Data Protection
- V9 - Communication
- V10 - Malicious Code
- V11 - Business Logic
- V12 - File and Resources
- V13 - API and Web Service
- V14 - Configuration

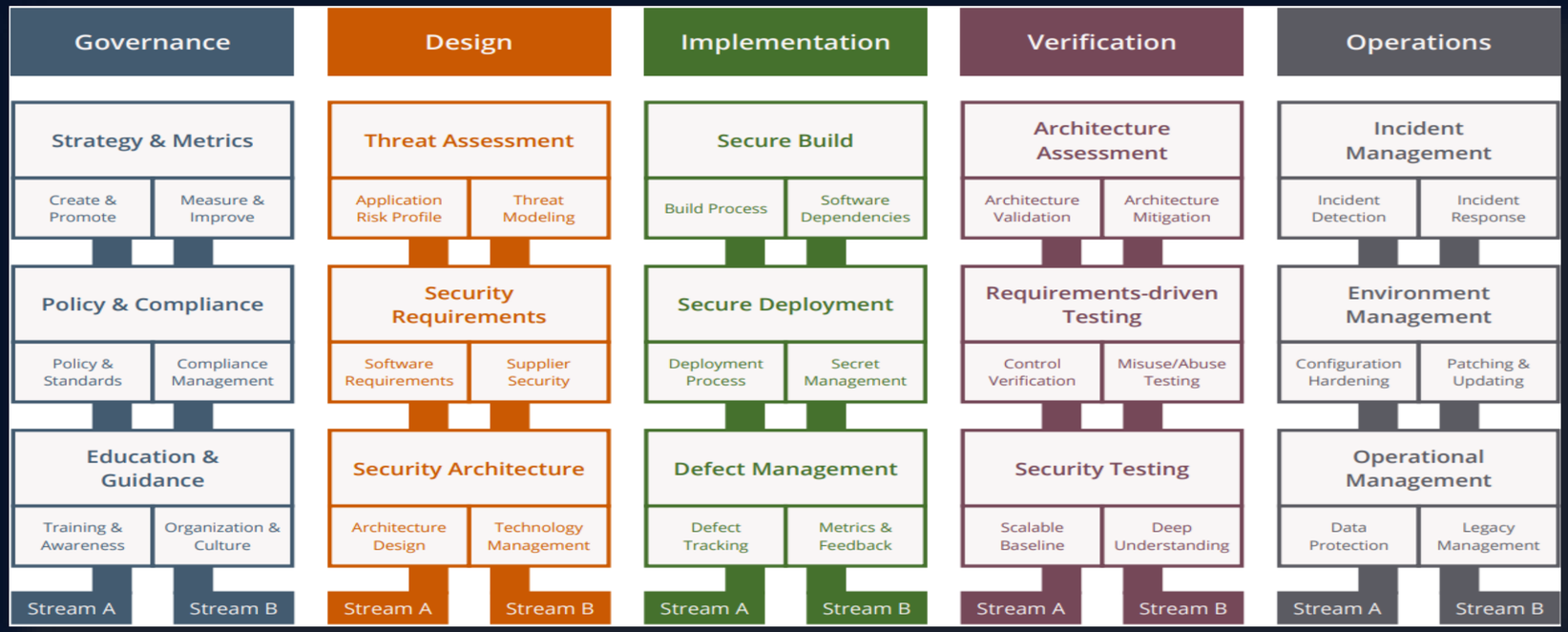
OWASP ASVS – Verification Levels

Level	Target applications	Testing methodology	Detection Tools
Level 1	Entry Level	Black box is enough	DAST, IAST
Level 2	Applications that contains sensitive data	Black box + White Box	SAST, IAST
Level 3	Most Critical Applications (high value transactions, medical data, etc)	Black box + White Box	SAST, IAST

OWASP (SAMM)

OWASP Software Assurance Maturity Model provides an effective and measurable way for all types of organizations to analyze and improve their software security posture.

OWASP SAMM supports the complete software lifecycle, resulting in more secure software and reduced risk of security breaches and vulnerabilities.



Resources

Useful Links and Tools

- **OWASP in SDLC**
https://owasp.org/www-project-integration-standards/writeups/owasp_in_sdlc/

- **OWASP Application Testing Guide**
https://owasp.org/www-project-web-security-testing-guide/assets/archive/OWASP_Testing_Guide_v4.pdf

- **OWASP ZAP**
https://www.zaproxy.org/

- **Burp Suite**
https://portswigger.net/burp

- **CIS Benchmark**
https://www.cisecurity.org/cis-benchmarks

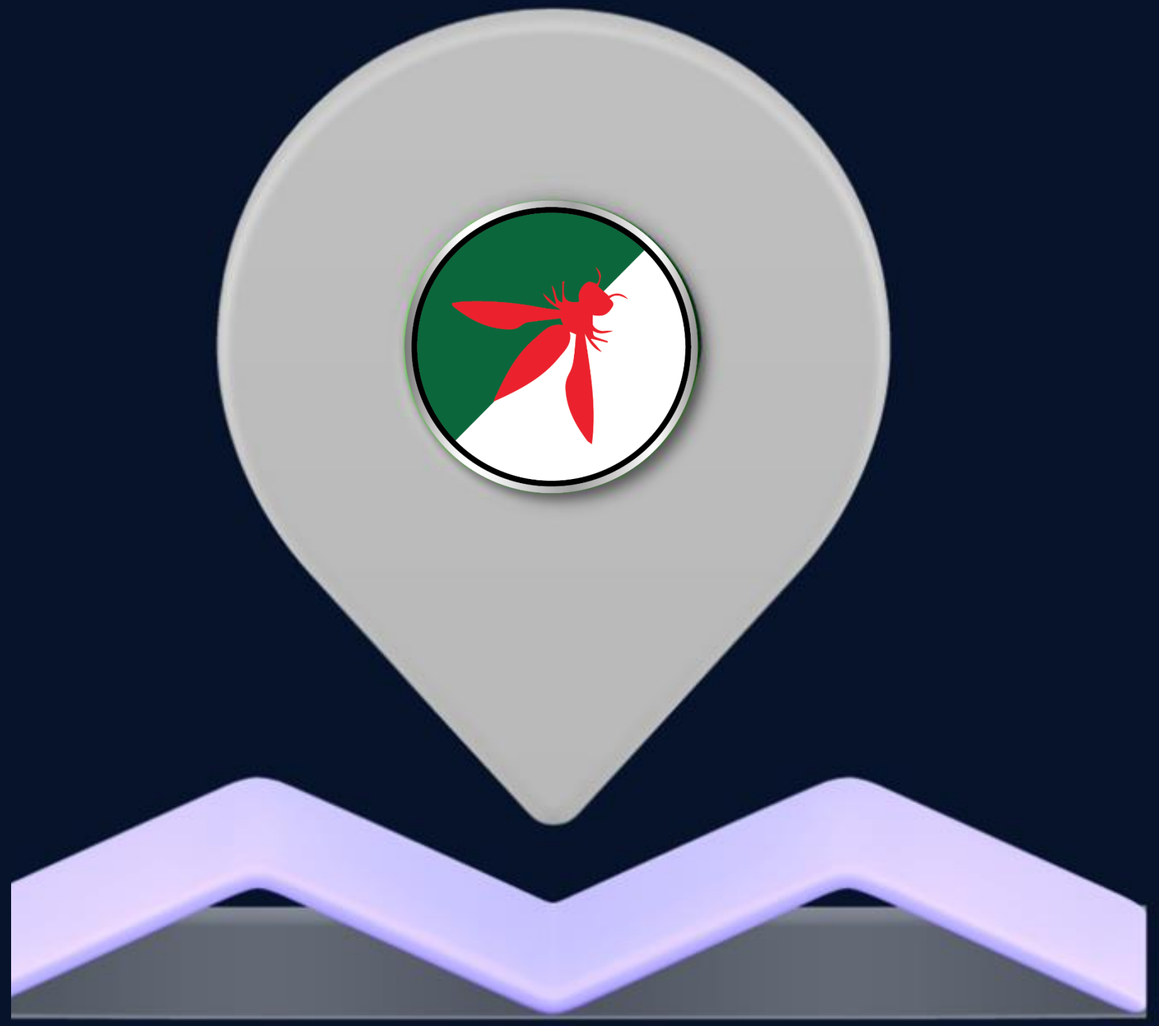
- **SAST & DAST**
https://www.gartner.com/reviews/market/application-security-testing

- **OWASP ASVS**
https://owasp.org/www-project-application-security-verification-standard/

- **OWASP SAMM**
https://owaspsamm.org/

OWASP ALGIERS

Contact us



ALGIERS-LEADERS@OWASP.ORG

https://owasp.org/www-chapter-algiers/

owasp-algiers

OWASP Algiers Chapter