



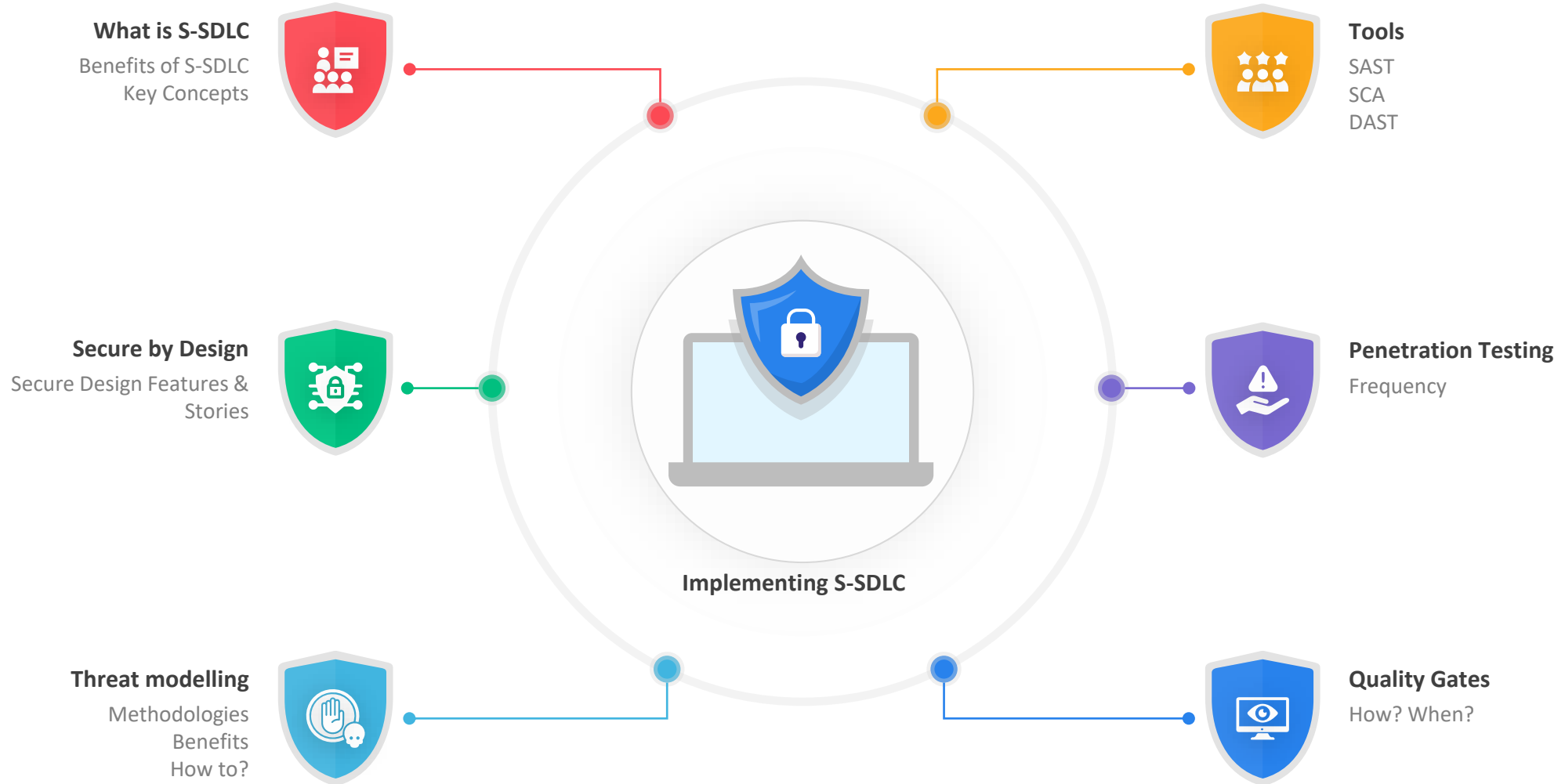
Secure-SDLC

Threat Modelling, Security by Design and
Quality Gates



Akif KAYAPINAR

Table of Content



Problems with SDLC

Traditional SDLC practices do not help with quick vulnerability detection and remediation.

Over the past decade, the number of vulnerabilities has increased, as has the cost of data breaches.

Security testing typically occurs only after the software is developed. This can lead to vulnerabilities being discovered late in the process, increasing the cost of development, also can lead to security breaches

Benefits

**Proactive
Application
Security**



**Reduces
Disruption to
Business**



Cost Efficient



**Helps in
Mitigating
Potential Risks**



Compliance



Key Concepts

Security by Design

Security requirements should be included during initial planning and design stages



Threat Modelling

Threat Modeling (TM) is software design analysis that looks for security weakness by juxtaposing software design views against a set of threat agents



SAST, SCA & DAST

Static Application Security Testing
Dynamic application Security Testing
Software composition analysis

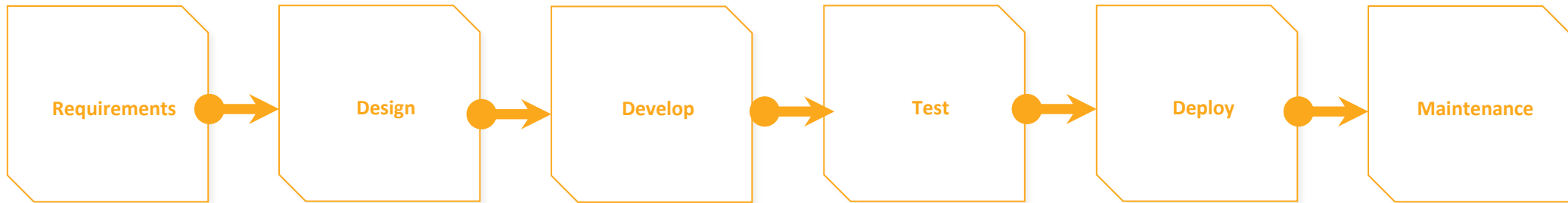


Penetration Testing

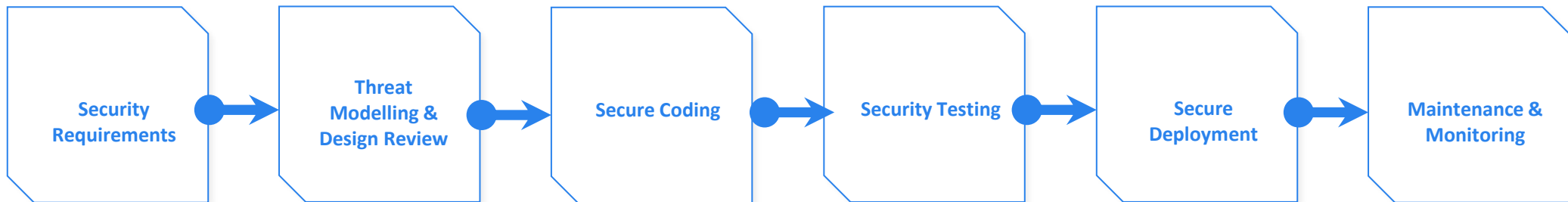
Making penetration testing part of your SDLC process to ensure the end product free from security vulnerabilities



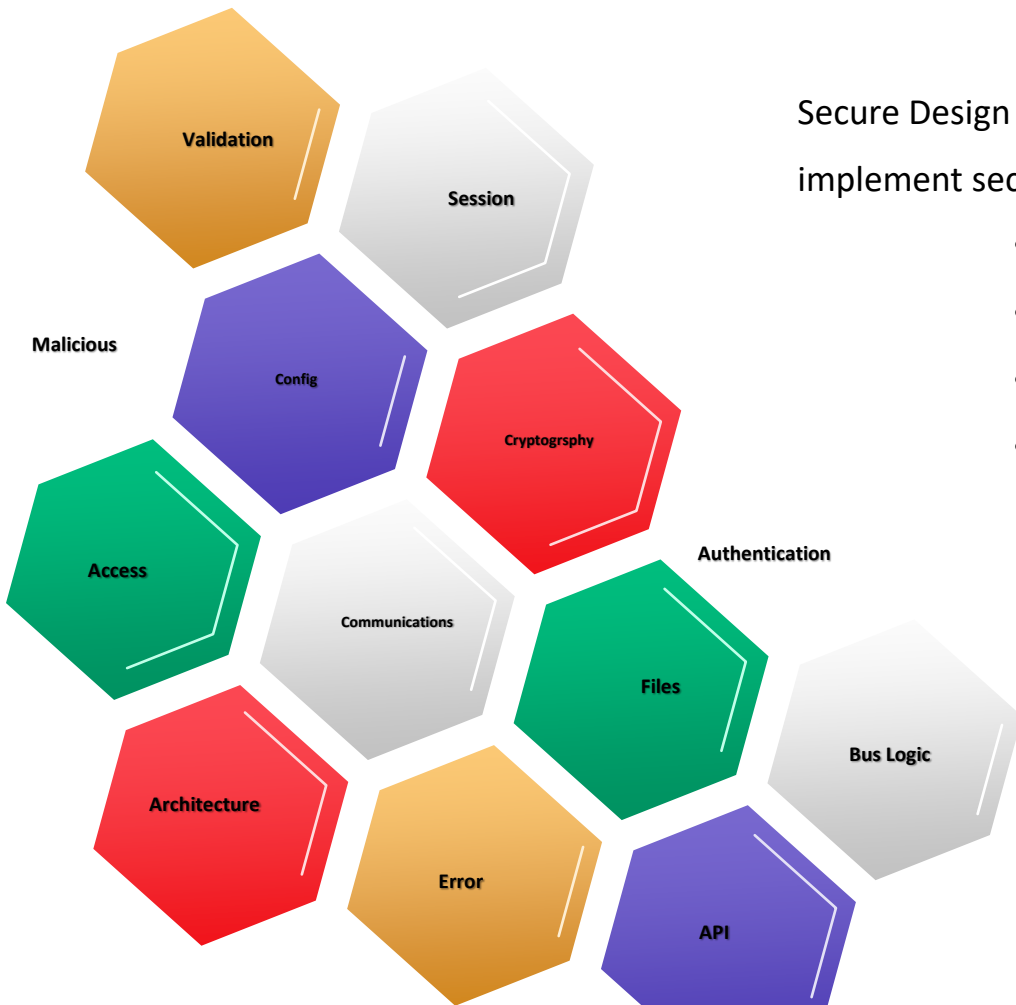
SDLC , S-SDLC



Secure SDLC enhances SDLC by embedding security into each development phase.



Secure by Design



Secure Design Features/Stories provide security requirements/patterns to enable applications to implement security consistently. These security requirements might cover:

- Compliance
- Company Policies & Standards
- OWASP ASVS
- NIST Cybersecurity Framework

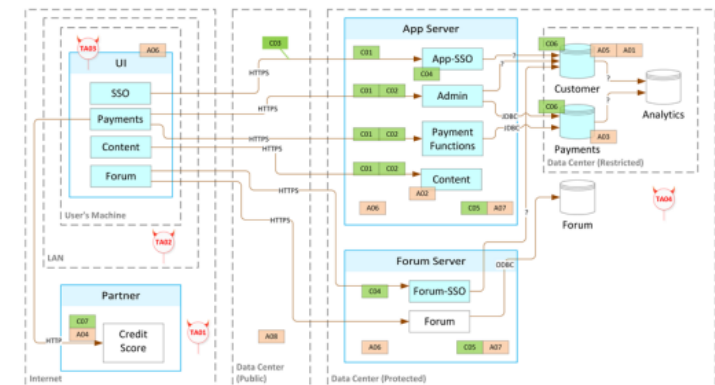
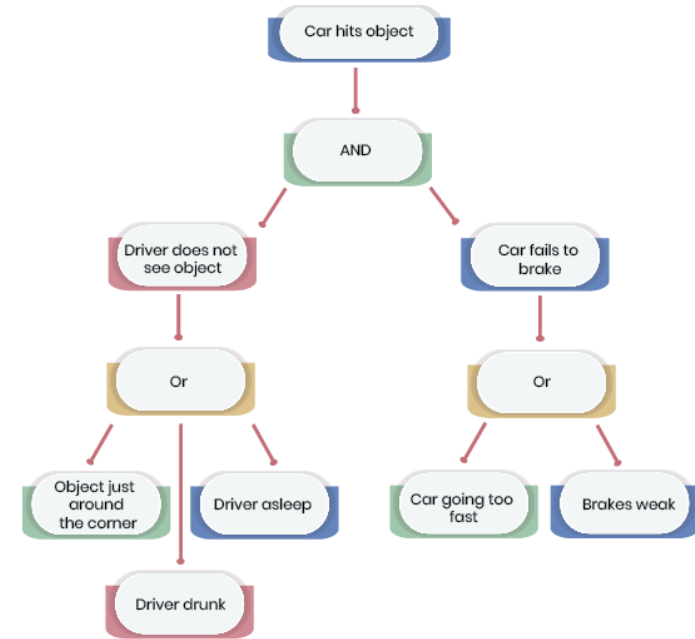
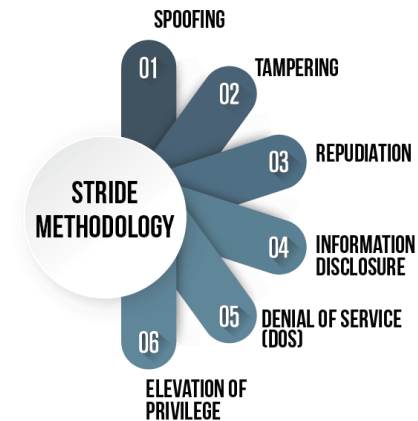
Threat Modelling

Threat Modeling (TM) is software design analysis that looks for security weakness by juxtaposing software design views against a set of threat agents.

- Identifies secure design weaknesses
 - Missing security controls
 - Weak or inappropriate security controls
 - Potential vulnerabilities
- Threat modeling finds weaknesses that cannot be found by other techniques
- It is not a replacement for pen-testing, secure code reviews or any other SSI capability

Threat Modelling Methodologies

- Attack Trees
- STRIDE
- Synopsys
- Pasta
- Trike
- DREAD



SAST, DAST & SCA

Static Application Security Testing

Analyze Source code
Identify potential security flaws early in development Process
Integrate into CI/CD pipeline to provide ongoing assessments
Be aware of False Positives

Software Composition Analysis

Analyze open source libraries
Vulnerability Detection and Monitoring
Ensure compliance for licensing obligations
Easy CI/CD pipeline integration for ongoing assessments

Dynamic Application Security Testing

Simulates attackers perspective
Catch vulnerabilities before its deployed

Penetration Testing

Helps an organization discover vulnerabilities and flaws in their systems that they might not have otherwise been able to find.

Penetration tests should be performed before every major release or significant changes within security architecture such as changes to cryptography, authentication, etc.

Identify policies within company for remediation timelines. *Example, All Critical and High findings must be remediated before release and medium findings needs to be remediated within 90 days.*

Quality Gates

Quality gates are checkpoints that require deliverables to meet specific, measurable success criteria before progressing. They help foster confidence and consistency throughout the entire software development lifecycle

Security must be part of the quality gates to make sure Secure SDLC is being followed in every step

Planning

Development

Integration

**Customer
Acceptance**

Production

Everything Altogether

1

During the initial Planning and design stage, Engage with Development team to perform initial risk assessment

2

After the initial discussions, Perform Threat modelling, or identify security design Features that will be applicable.

Threat modelling is an extensive process there fore not every development effort would require such extensive effort.



3

Provide identified gaps either form of Threat modelling output or secure design stories to the development teams so that capacity planning can be done by the team before development effort starts.



4

The first Quality gate is the planning gate and before development starts first gate needs to be closed. Within the first Gate inject Risk Assessment step as a security gate to make sure all the security requirements are identified and assigned to the development efforts.



5

During development, tools such as SAST, SCA needs to be incorporated. If no tooling is available manual reviews should be performed. Also all identified security controls must be implemented.



6


Another Security gate must be injected within the Development stage of Quality Gates. For this gate to be closed, All identified security controls must be implemented, and code analysis/ open-source library scans must be completed.

7

During the Integration part of SDLC, all the security testing needs to be performed. DAST and or Penetration testing performed at this stage.

8

Within Integration stage of Quality gates, another security quality gate should be injected. At this gate all security testing results are analyzed and quality gate can be signed off.

The image features a central text element 'Q&A?' surrounded by six vertical bars of different colors (red, orange, green, purple, blue, and teal) arranged in two columns of three on either side.

Q&A?