- OWASP Code Review

Some references to look at
https://developer.salesforce.com/docs/atlas.en-us.secure_coding_guide.meta/secure_coding_guide/secure_coding_guidelines.htm

Programs written in typed safe languages (such as C# or Java) are less vulnerable to certain security bugs such as buffer overflows than others like C and C++

Design Questions During Secure Code Review

Data flow
- Are user inputs used to directly reference business logic?
- Is there potential for data binding flaws?
- Is the execution flow correct in failure cases?

Authentication and access control
- Does the design implement access control for all resources?
- Are sessions handled correctly?
  What functionality can be accessed without authentication?

Existing security controls
- Are there any known weaknesses in third-part security controls
- Is the placements of security controls correct?

Architecture
- Are connections to external servers secure?
- Are inputs from external sources validated?

Configuration files and data stores
- Is there any sensitive data in configuration files?
- Who has access to configuration or data files?

Code Review Checklist

- Data Validation
- Authentication

- Session Management
- Authorization
- Cryptography
- Error Handling
- Logging
- Security Configuration
- Network Architecture

Advantages To Using Source Code Scanners

Reduction in manual efforts
Find all the instances of the vulnerabilities
Source to sink analysis

Disadvantages To Using Source Code Scanners

Business logic flaws remain untouched
Limited scope
Design flaws
False positives

Threat modeling process

1: Decompose the Application.
    External Dependencies
    Entry Points
    Assets
    Determining the Attack Surface
    Trust Levels
    Data flow analysis
    Transaction analysis
    Data Flow Diagrams
2: Determine and rank threats .
    STRIDE : Spoofing - Tampering - Repudiation - Information Disclosure - DOS - Priv Esc
    DREAD: Damage - Reproducibility - Exploitability - Affected users - Discovrability
3: Determine countermeasures and mitigation.

A1 Injection
A2 Broken Authentication And Session Management
A3 Cross-Site Scripting (XSS)
A4 Insecure Direct Object Reference
A5 Security Misconfiguration
A6 Sensitive Data Exposure
A7 Missing Function Level Access Control
A8 Cross-Site Request Forgery (CSRF)
A9 Using Components With Know Vulnerabilities
A10 Unvalidated Redirects And Forwards

HTML5
Same Origin Policy Reviewing Logging Code Error Handling Reviewing Security Alerts Review For Active Defence Race Conditions
Buffer Overruns
Client Side JavaScript
Appendix
Code Review Do's And Dont's Code Review Checklist Threat Modeling Example Code Crawling