Protecting Information Assets
- Week 11 -

Application Development Security
MIS5206 Week 11

• Team Presentations
• Application Development Security
• Test Taking Tip
• Quiz
Application Development Security

As applications become more accessible through the web, cloud and mobile devices, organizations are being forced to abandon their reactive approach to security and, instead, to take a proactive approach by minimizing risk directly in the software they buy, create and use to serve themselves and their customers.
Usual trend

1. Buggy software is released to the market to beat the competition.

2. Hackers find new vulnerabilities and weaknesses in new software.

3. Websites post these vulnerabilities and how to exploit them.

4. Vendor develops and releases patch to fix vulnerabilities.

5. The new patch goes on the stack of software patches that all network administrators need to test and install.

Perimeter security solutions are often relied on as a solution to insecure application development practices

- Secure applications development practices not a consideration in the past
- AppDev projects’ scope/schedule/budget focuses on functionality not security
- Software developers are not security professionals and do not focus on security
- Software vendors skip proper security architecture, design and testing steps as they race to beat competitors to market with new features
- Security professionals typically not software developers – and lack complete insight in software vulnerability issues
- IT customers have come to expect to receive flawed software needing upgrades and patches
- IT customers unable to control the flaws in the software they purchase rely on perimeter protection
Best Practice: Build Security In

**Security Architecture**
Creation, use and enforcement of System Architecture standards provides the basic building blocks for developing, implementing and maintaining secure applications.

**Software Development Life Cycle**
Attention to security throughout the Software Development Life Cycle (SDLC) is the key to creating secure, manageable applications regardless of platform or technologies.

**Procurement Standards**
Describing the process and detailed criteria that will be used to assess the security level of third party software enables companies to make strategic, security-sensitive decisions about purchased software purchases.
Software Development Life Cycle

Requirements
- Why the software was created (i.e. goals)
- Who the software was created for
- What the software is intended to do

Design
- Specifications that identify how the software and data will be formed to accomplish the goals and be used to meet the requirements

Development
- Programming software code implemented and integrated with other systems and software libraries to meet specifications

Testing-Validation
- Assuring that the software and data works as planned to meet the goals

Release-Maintenance
- Deploying the software and data, and assuring it is properly configured, patched and monitored
Classic software development

Waterfall Model

Requirement Analysis
System Design
Implementation
Testing
Deployment
Maintenance

MIS 5206 Protecting Information Assets
Modified waterfall model...
Waterfall missed requirements identified too late
“V model” added testing throughout the development process

Like the waterfall model, V-model is considered very rigid and does not allow for flexible evolution of user requirements (and lacks formal handling of risk)
Rapid prototyping

Involves the users/stakeholders in an incremental, interactive, iterative process of validating and evolving requirements...

Iterative, incremental, spiral, RAD, agile application system development...
Incremental software design and development

They provide a more “agile” process...

The agile models do not use prototypes to represent the full product, but break the product application into individual features that are constantly being delivered.
Application development methodologies

- **Break and fix** – No real planning up front. Flaws reactively dealt with after release with the creation of updates and patches
- **Waterfall** – Very rigid, sequential approach that requires each phase to complete before the next one can begin. Inflexibility make it difficult to integrate changes
- **V-model** – Emphasizes verification and validation at each phase and testing to take place throughout the project, not just at the end
- **Prototyping** – Creating a model or sample of the application to elicit user feedback and for proof-of-concept purposes
- **Incremental** – Multiple development cycles are carried out on a piece of software throughout its development stages. Each phase provides a usable version of software
- **Spiral** – Iterative approach that emphasizes risk analysis per iteration. Allows for customer feedback to be integrated through a flexible evolutionary approach
- **Rapid Application Development (RAD)** – Combines prototyping and iterative development procedures with the goal of accelerating the software development process
- **Agile** – Iterative and incremental development processes that encourage team-based collaboration. Flexibility and adaptability are used instead of a strict process structure
Security strategy needs to be a consideration at each level of the architecture.

Security Architecture

- BUSINESS: Business Process and Operations
- APPLICATIONS: User Access - Multiple Technologies
- DATA: Databases and File Repositories
- INFRASTRUCTURE: Device and Configurations
SDLC Security Planning

- Functional and Technical Features/Requirements
- Staff background Checks
- Security plans, scenarios, and scripts
- Security Controls in Specifications
- Operational Practices
Development in-house concerns

- Security features
- Development process
- Changing requirements
- Threats
- Vulnerabilities
- Malicious insiders
SDLC and Security

Requirements analysis
– *Informational, functional, behavioral, and performance specifications...*
  + CIA risk assessment, + Risk-level acceptance,...

Design
– *Data models and data dictionary, work process and status transition models, input/output models, data flow models, flow of control models...*
  + Threat modeling, + Attack surface analysis,...

Develop ("*make*") / Implement ("*buy*")
– *Source code control system, code reviews, daily builds, automated CASE tools...*
  + Developer security training, + Static analysis, + Secure code repositories,...

Testing/Validation
– *Unit testing and integration testing (daily builds), manual and regression testing, user acceptance testing*
  + Dynamic analysis, + Fuzzing,...

Release/Maintenance
– *Release testing*
  + Separation of duties, +Change management,...
### SDLC and Security

#### Requirements analysis

- *Informational, functional, behavioral, and performance specifications...*
- + CIA risk assessment, + Risk-level acceptance,...

<table>
<thead>
<tr>
<th>Organisation &amp; relevant process</th>
<th>Name of Asset</th>
<th>Personal Identifying Information (PII) (Y/N)</th>
<th>Personal Health Information (PHI) (Y/N)</th>
<th>Critical Infrastructure Information (CII) (Y/N)</th>
<th>Customer Data (Y/N)</th>
<th>Organization Data (Y/N)</th>
<th>Confidentiality</th>
<th>Integrity</th>
<th>Availability</th>
<th>Categorization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thermal Distribution System</td>
<td>ChilledWater</td>
<td>TDS</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Thermal Distribution System</td>
<td>HeatedWater</td>
<td>TDS</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Thermal Distribution System</td>
<td>TDS</td>
<td>TDS</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Communication</td>
<td>Data</td>
<td>COM</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Communication</td>
<td>Voice</td>
<td>COM</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Communication</td>
<td>Security</td>
<td>COM</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Communication</td>
<td>COM</td>
<td>COM</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Public Works</td>
<td>Sewer</td>
<td>Utilities</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Public Works</td>
<td>Stormwater</td>
<td>Utilities</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Public Works</td>
<td>Water</td>
<td>Utilities</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Public Works</td>
<td>Utilities</td>
<td>Utilities</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>External</td>
<td>Parcels</td>
<td>Parcels</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>

*MIS 5206 Protecting Information Assets*
Software requirements often specified with

- **Informational model** – Dictates the type and content of information that will be processed and how it will be processed
- **Functional model** – Outlines the tasks and functions the application needs to carry out
- **Behavioral model** – Explains the states the application will be in during and after specific transitions take place
Validation versus Verification

• Validation
  – “Did they build the right application?”
  – In the development of large applications it is easy to lose sight of the main goal
  – Determines if the application/system provides the necessary solution for the intended real-world problem

• Verification
  – “Did they build the application right?”
  – Applications can be built that do not match the original specifications
    • *Often not designed/developed with security requirements in mind...*
  – Determines if the application accurately represent and meets the specifications
  – This step ensures that the specifications were met properly
Informational Model

Dock Administration Page

<table>
<thead>
<tr>
<th>Widget Type</th>
<th>Widget Name</th>
<th>Table</th>
<th>Column</th>
<th>Data Type</th>
<th>Source Table</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Grid</td>
<td>Owner/Operator/Both</td>
<td>OPERATOR_OWNER</td>
<td>OPERATOR_OWNER_BOTH</td>
<td>VARCHAR2</td>
<td>FILEMAKER</td>
<td>OPERATE</td>
</tr>
<tr>
<td>Data Grid</td>
<td>Organization</td>
<td>OPERATOR_OWNER</td>
<td>ORGANIZATION</td>
<td>VARCHAR2</td>
<td>FILEMAKER</td>
<td>ORGANIZ</td>
</tr>
<tr>
<td>Data Grid</td>
<td>Public or Private</td>
<td>OPERATOR_OWNER</td>
<td>PUBLIC_OR_PRIVATE</td>
<td>VARCHAR2</td>
<td>FILEMAKER</td>
<td>OWNERS</td>
</tr>
<tr>
<td>Data Grid</td>
<td>Current or Prior</td>
<td>OPERATOR_OWNER</td>
<td>CURRENT_OR_PRIOR</td>
<td>VARCHAR2</td>
<td>FILEMAKER</td>
<td>NAV</td>
</tr>
<tr>
<td>Data Grid</td>
<td>Facility Website</td>
<td>OPERATOR_OWNER</td>
<td>FACILITY_WEBSITE</td>
<td>VARCHAR2</td>
<td>FILEMAKER</td>
<td>URL</td>
</tr>
<tr>
<td>Data Grid</td>
<td>Official Title</td>
<td>CONTACT</td>
<td>OFFICIAL_TITLE</td>
<td>VARCHAR2</td>
<td>FILEMAKER</td>
<td>TITLE</td>
</tr>
<tr>
<td>Data Grid</td>
<td>Name Title</td>
<td>CONTACT</td>
<td>NAME_TITLE</td>
<td>VARCHAR2</td>
<td>FILEMAKER</td>
<td>NAME</td>
</tr>
<tr>
<td>Data Grid</td>
<td>First Name</td>
<td>CONTACT</td>
<td>FIRST_NAME</td>
<td>VARCHAR2</td>
<td>FILEMAKER</td>
<td>FIRST</td>
</tr>
<tr>
<td>Data Grid</td>
<td>Last Name</td>
<td>CONTACT</td>
<td>LAST_NAME</td>
<td>VARCHAR2</td>
<td>FILEMAKER</td>
<td>LAST</td>
</tr>
<tr>
<td>Data Grid</td>
<td>Middle Initial</td>
<td>CONTACT</td>
<td>MIDDLEInicial</td>
<td>VARCHAR2</td>
<td>FILEMAKER</td>
<td>MIDDLE</td>
</tr>
<tr>
<td>Data Grid</td>
<td>Phone</td>
<td>CONTACT</td>
<td>PHONE</td>
<td>VARCHAR2</td>
<td>FILEMAKER</td>
<td>PHONE</td>
</tr>
<tr>
<td>Data Grid</td>
<td>Fax</td>
<td>CONTACT</td>
<td>FAX</td>
<td>VARCHAR2</td>
<td>FILEMAKER</td>
<td>FAX</td>
</tr>
<tr>
<td>Data Grid</td>
<td>Department</td>
<td>CONTACT</td>
<td>DEPARTMENT</td>
<td>VARCHAR2</td>
<td>FILEMAKER</td>
<td>DEPARTMENT</td>
</tr>
<tr>
<td>Data Grid</td>
<td>Room Number</td>
<td>CONTACT</td>
<td>ROOM_NUMBER</td>
<td>VARCHAR2</td>
<td>FILEMAKER</td>
<td>ROOM_NO</td>
</tr>
<tr>
<td>Data Grid</td>
<td>Street Address</td>
<td>CONTACT</td>
<td>STREET_ADDRESS</td>
<td>VARCHAR2</td>
<td>FILEMAKER</td>
<td>STREET</td>
</tr>
<tr>
<td>Data Grid</td>
<td>PO Box</td>
<td>CONTACT</td>
<td>PO_BOX</td>
<td>VARCHAR2</td>
<td>FILEMAKER</td>
<td>PO_BBOX</td>
</tr>
<tr>
<td>Data Grid</td>
<td>City</td>
<td>CONTACT</td>
<td>CITY</td>
<td>VARCHAR2</td>
<td>FILEMAKER</td>
<td>CITY</td>
</tr>
<tr>
<td>Data Grid</td>
<td>State</td>
<td>CONTACT</td>
<td>STATE_CONTACT</td>
<td>VARCHAR2</td>
<td>FILEMAKER</td>
<td>STATE</td>
</tr>
<tr>
<td>Data Grid</td>
<td>Zip Code</td>
<td>CONTACT</td>
<td>ZIP_CODE</td>
<td>VARCHAR2</td>
<td>FILEMAKER</td>
<td>ZIP</td>
</tr>
<tr>
<td>Data Grid</td>
<td>Email</td>
<td>CONTACT</td>
<td>EMAIL</td>
<td>VARCHAR2</td>
<td>FILEMAKER</td>
<td>EMAIL</td>
</tr>
<tr>
<td>Data Grid</td>
<td>User ID</td>
<td>NAV_UNIT_UPDATE_HISTORY</td>
<td>USER_ID</td>
<td>NUMBER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Grid</td>
<td>Update Type</td>
<td>NAV_UNIT_UPDATE_HISTORY</td>
<td>UPDATE_TYPE</td>
<td>VARCHAR2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Grid</td>
<td>Comment</td>
<td>NAV_UNIT_UPDATE_HISTORY</td>
<td>COMMENT</td>
<td>VARCHAR2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Grid</td>
<td>Date</td>
<td>NAV_UNIT_UPDATE_HISTORY</td>
<td>DATE_TIME</td>
<td>DATE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dropdown</td>
<td>Migration Status</td>
<td>MIGRATION_STATUS</td>
<td>MIGRATION_STATUS</td>
<td>VARCHAR2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Grid</td>
<td>This Navigation Unit is a</td>
<td>PARENT CHILD</td>
<td>PARENT_OR_CHILD</td>
<td>VARCHAR2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Grid</td>
<td>Of Related GUID</td>
<td>PARENT CHILD</td>
<td>RELATED_NAV_UNIT</td>
<td>NUMBER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Grid</td>
<td>Relationship</td>
<td>PARENT CHILD</td>
<td>RELATIONSHIP</td>
<td>VARCHAR2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Data dictionary
Verification

Did the build the application right?

Does it match the data model?
Figure 2. Use Case Hierarchy Diagram

Validation

Did they build the right application?
Functional model

Figure 4. Review Pump Station Status Use Case Diagram

Validation

*Did they build the right application?*

MIS 5206 Protecting Information Assets
Behavioral model

Validation
“Did they build the right application?”

Verification
“Did they build the application right?”
Behavioral model

PIF/SIF

Draft → Created → Released → SM-Approved → Recommended → DD-Approved → Transmitted to Design

FM-Rejected → SM-Rejected → DD-Rejected → Delayed

PIF Status Summary

<table>
<thead>
<tr>
<th>Status</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Not Started</td>
<td>0</td>
</tr>
<tr>
<td>2 Draft</td>
<td>64</td>
</tr>
<tr>
<td>3 Created</td>
<td>10</td>
</tr>
<tr>
<td>4 Released</td>
<td>587</td>
</tr>
<tr>
<td>5 FM-Rejected</td>
<td>29</td>
</tr>
<tr>
<td>6 SM-Approved</td>
<td>0</td>
</tr>
<tr>
<td>7 SM-Rejected</td>
<td>0</td>
</tr>
<tr>
<td>8 Recommended</td>
<td>0</td>
</tr>
<tr>
<td>9 DD-Approved</td>
<td>6</td>
</tr>
<tr>
<td>10 DD-Rejected</td>
<td>0</td>
</tr>
<tr>
<td>11 Delayed</td>
<td>0</td>
</tr>
<tr>
<td>12 Transmitted to Design</td>
<td>2</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>698</strong></td>
</tr>
</tbody>
</table>
Behavioral model
SDLC and Security

Requirements analysis

– *Informational, functional, behavioral, and performance specifications*...

+ CIA risk assessment, + Risk-level acceptance,...

Design

– *Data models and data dictionary, work process and status transition models, input/output models, data flow models, flow of control models*...

+ Threat modeling, + Attack surface analysis,...
SDLC Design Security

Threat modeling is a systematic approach for understanding how different threats could be realized and a successful attack could take place.

Microsoft’s Threat Modeling Process

- Identify Security Objectives
- Application Overview
- Identify Vulnerabilities
- Identify Threats
- Decompose Application

- Attackers may be able to read other users' messages

- User may not have logged off on a shared computer
- Data validation may fail, allowing SQL injection
- Authorization may fail, allowing unauthorized access
- Browser cache may contain contents of message

- Implement data validation
- Implement authorization checks
- Implement anticaching HTTP headers
- If risk is high, use SSL
**SDLC Design Security**

*Attack surface* is what is available to be used by an attacker against the application itself.

Goal of attack surface analysis is to identify and reduce the amount of code and functionality accessible to untrusted users.

Development team should reduce the attack surface as much as possible to remove “resources” that can be used as avenues for the attacker to use.
MITRE’s Common Weakness Enumeration

CWE/SANS TOP 25 Most Dangerous Software Errors

What Errors Are Included in the Top 25 Software Errors?
Version 3.0 Updated June 27, 2011

The Top 25 Software Errors are listed below in three categories:

- Software Error Category: Insecure Interaction Between Components (6 errors)
- Software Error Category: Risky Resource Management (8 errors)
- Software Error Category: Porous Defenses (11 errors)
# CWE/SANS TOP 25 Most Dangerous Software Errors

## Insecure Interaction Between Components
These weaknesses are related to insecure ways in which data is sent and received between separate components, modules, programs, processes, threads, or systems.

<table>
<thead>
<tr>
<th>CWE ID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CWE-89</td>
<td>Improper Neutralization of Special Elements used in an SQL Command ('SQL injection')</td>
</tr>
<tr>
<td>CWE-78</td>
<td>Improper Neutralization of Special Elements used in an OS Command ('OS Command injection')</td>
</tr>
<tr>
<td>CWE-79</td>
<td>Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')</td>
</tr>
<tr>
<td>CWE-434</td>
<td>Unrestricted Upload of File with Dangerous Type</td>
</tr>
<tr>
<td>CWE-352</td>
<td>Cross-Site Request Forgery (CSRF)</td>
</tr>
<tr>
<td>CWE-661</td>
<td>URL Redirection to Untrusted Site ('Open Redirect')</td>
</tr>
</tbody>
</table>

## Porous Defenses
The weaknesses in this category are related to defensive techniques that are often misused, abused, or just plain ignored.

<table>
<thead>
<tr>
<th>CWE ID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CWE-306</td>
<td>Missing Authentication for Critical Function</td>
</tr>
<tr>
<td>CWE-862</td>
<td>Missing Authorization</td>
</tr>
<tr>
<td>CWE-798</td>
<td>Use of Hard-coded Credentials</td>
</tr>
<tr>
<td>CWE-311</td>
<td>Missing Encryption of Sensitive Data</td>
</tr>
<tr>
<td>CWE-807</td>
<td>Reliance on Untrusted Inputs in a Security Decision</td>
</tr>
<tr>
<td>CWE-730</td>
<td>Execution with Unnecessary Privileges</td>
</tr>
<tr>
<td>CWE-863</td>
<td>Incorrect Authorization</td>
</tr>
<tr>
<td>CWE-732</td>
<td>Incorrect Permission Assignment for Critical Resource</td>
</tr>
<tr>
<td>CWE-327</td>
<td>Use of a Broken or Risky Cryptographic Algorithm</td>
</tr>
<tr>
<td>CWE-307</td>
<td>Improper Restriction of Excessive Authentication Attempts</td>
</tr>
<tr>
<td>CWE-799</td>
<td>Use of a One-Way Hash without a Salt</td>
</tr>
</tbody>
</table>

## Risky Resource Management
The weaknesses in this category are related to ways in which software does not properly manage the creation, usage, transfer, or destruction of important system resources.

<table>
<thead>
<tr>
<th>CWE ID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CWE-120</td>
<td>Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')</td>
</tr>
<tr>
<td>CWE-22</td>
<td>Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')</td>
</tr>
<tr>
<td>CWE-494</td>
<td>Download of Code Without Integrity Check</td>
</tr>
<tr>
<td>CWE-829</td>
<td>Inclusion of Functionality from Untrusted Control Sphere</td>
</tr>
<tr>
<td>CWE-676</td>
<td>Use of Potentially Dangerous Function</td>
</tr>
<tr>
<td>CWE-131</td>
<td>Incorrect Calculation of Buffer Size</td>
</tr>
<tr>
<td>CWE-134</td>
<td>Uncontrolled Format String</td>
</tr>
<tr>
<td>CWE-190</td>
<td>Integer Overflow or Wraparound</td>
</tr>
</tbody>
</table>
The OWASP Top 10 - 2013 is as follows:

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

Requirements analysis
- *Informational, functional, behavioral, and performance specifications*...
  + CIA risk assessment, + Risk-level acceptance,...

Design
- *Data models and data dictionary, work process and status transition models, input/output models, data flow models, flow of control models*...
  + Threat modeling, + Attack surface analysis,...

Develop (*"make"*) / Implement (*"buy"*)
- *Source code control system, code reviews, daily builds, automated CASE tools*...
  + Developer security training, + Static analysis, + Secure code repositories,...
## Secure Software Development

### Secure Software Development Curriculum

<table>
<thead>
<tr>
<th>Level</th>
<th>Course</th>
<th>Certification</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Level 1</strong></td>
<td>DEV522: Defending Web Applications Security Essentials</td>
<td>GWEB</td>
</tr>
<tr>
<td></td>
<td>DEV531: Defending Mobile Applications Security Essentials</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DEV534: Secure DevOps: A Practical Introduction</td>
<td></td>
</tr>
<tr>
<td><strong>Level 2</strong></td>
<td>DEV541: Secure Coding in Java/JEE: Developing Defensible Applications</td>
<td>GSSP-JAVA</td>
</tr>
<tr>
<td></td>
<td>DEV543: Secure Coding in C &amp; C++</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DEV544: Secure Coding in .NET: Developing Defensible Applications</td>
<td>GSSP-.NET</td>
</tr>
<tr>
<td><strong>Specialty Courses</strong></td>
<td>SEC542: Web App Penetration Testing and Ethical Hacking</td>
<td>GWAPT</td>
</tr>
<tr>
<td></td>
<td>SEC642: Advanced Web App Penetration Testing, Ethical Hacking, and Exploitation Techniques</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HST: Certified Secure Software Lifecycle Professional (CSSLP®) CBK® Training Seminar</td>
<td></td>
</tr>
</tbody>
</table>
# List of tools for static code analysis

This article needs additional cita (Learn how and when to remove this)

This is a list of tools for static code analysis.

<table>
<thead>
<tr>
<th>Name</th>
<th>Manager</th>
<th>Established</th>
<th>Server side: all free software</th>
<th>Client side: all free JS code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allot</td>
<td>Debian Project</td>
<td>2003</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Assembla</td>
<td>Assembla, Inc</td>
<td>2005</td>
<td>No</td>
<td>Unknown</td>
</tr>
<tr>
<td>Betavine</td>
<td>Vodafone</td>
<td>2007</td>
<td>No</td>
<td>Unknown</td>
</tr>
<tr>
<td>Bitbucket</td>
<td>Allassian</td>
<td>2008</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>CloudForge</td>
<td>CollabNet</td>
<td>2000</td>
<td>No</td>
<td>Unknown</td>
</tr>
<tr>
<td>CodePlex</td>
<td>Microsoft</td>
<td>2006-05</td>
<td>No</td>
<td>Unknown</td>
</tr>
<tr>
<td>GitHub</td>
<td>GitHub, Inc</td>
<td>2008-04</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Gna!</td>
<td>Unknown</td>
<td>2004-01</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>GNU Savannah</td>
<td>Savannah Administration</td>
<td>2001-01</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Kallithea</td>
<td>SFC</td>
<td>2014[^9]</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Launchpad</td>
<td>Canonical</td>
<td>2004</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>OSDN</td>
<td>OSDN K.K.</td>
<td>2002-04</td>
<td>Unknown</td>
<td>Yes</td>
</tr>
<tr>
<td>Ourproject.org</td>
<td>Comunes Collective</td>
<td>2002</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>OW2 Consortium</td>
<td>OW2 Consortium</td>
<td>Unknown</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td>Rosetta Code</td>
<td>Unknown</td>
<td>2007</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td>SEUL</td>
<td>Unknown</td>
<td>1997-05</td>
<td>Unknown</td>
<td>NO</td>
</tr>
<tr>
<td>SourceForge</td>
<td>BizX LLC</td>
<td>1999-11</td>
<td>Yes[^12][^13]</td>
<td>Yes</td>
</tr>
<tr>
<td>Tigris.org</td>
<td>(community)</td>
<td>2000</td>
<td>Unknown</td>
<td>No</td>
</tr>
<tr>
<td>Team Foundation Server</td>
<td>Microsoft</td>
<td>2012-2005[^18]</td>
<td>Unknown</td>
<td>No</td>
</tr>
<tr>
<td>Visual Studio Team Services</td>
<td>Microsoft</td>
<td>2012[^17]</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

## Language

**Multi-language**
- Advion Bauhaus Suite – A tool for Ada, C, C++, C#, and Java code that ensures the maintainability and stability of a code base.
- BlueOptima – Coding Effort Analytics objectively measures the productivity of the product.
- CAST Application Intelligence Platform – Detailed, audience-specific major databases.
SDLC and Security

Requirements analysis

- *Informational, functional, behavioral, and performance specifications*...
  + CIA risk assessment, + Risk-level acceptance,...

Design

- *Data models and data dictionary, work process and status transition models, input/output models, data flow models, flow of control models*...
  + Threat modeling, + Attack surface analysis,...

Develop (“make”) / Implement (“buy”)

- *Source code control system, code reviews, daily builds, automated CASE tools*...
  + Developer security training, + Static analysis, + Secure code repositories,...

Testing/Validation

- *Unit testing and integration testing (daily builds), manual and regression testing, user acceptance testing*
  + Dynamic analysis, + Fuzzing,...
Testing/validation
Testing/validation

MIS 5206 Protecting Information Assets
SDLC and Security

Requirements analysis
- *Informational, functional, behavioral, and performance specifications*...
  + CIA risk assessment, + Risk-level acceptance,...

Design
- *Data models and data dictionary, work process and status transition models, input/output models, data flow models, flow of control models*...
  + Threat modeling, + Attack surface analysis,...

Develop ("make") / Implement ("buy")
- *Source code control system, code reviews, daily builds, automated CASE tools*...
  + Developer security training, + Static analysis, + Secure code repositories,...

Testing/Validation
- *Unit testing and integration testing (daily builds), manual and regression testing, user acceptance testing*
  + Dynamic analysis, + Fuzzing,...

Release/Maintenance
- *Release testing*
  + Separation of duties, +Change management, +Operational practices...
Separation of Duties

Different environments (development, testing, and production) should be separated, without overlapping operations or functionality

- Developers should not have access to modify code used in production
- Code should be tested, submitted to a library, and then sent to the production environment
Releases/Maintenance

• Disaster Recovery Plan (DRP)
• COTS products (security patches?)
• Remove installation programs
• Machine content/intent
• File and program overlay settings and privileges
Operational Practices

• Commercial Off The Shelf (COTS) capabilities
• Open source libraries
• Operational Practices
  – System Security Plan (SSP)
  – Contingency Plan (BCP/DRP)
  – Awareness
  – Training
  – Documentation
Operational Practices

• Testing and Accreditation
  – Test Data
  – Test unit, subsystem, and entire system
  – Technical evaluation

• Security Management
  – Administrative controls
  – Safeguards
  – Physical security
  – Personnel, responsibilities, job functions, and interfaces
  – Commercial or in-house services
Operational Practices

• Support training classes
• User administration and access privileges
• Backup and restoration
  – Data, applications, configurations, restart instructions and procedures
  – Performing backups: How often? In which ways?
  – Performing backups
  – Offsite storage
  – Testing restoration
• Ensure implementation of only approved and accredited systems
• Cryptography keys
  – Generation and Use
  – Protection and storage
• Audit logs
  – How collected?
  – Where stored?
  – How protected?
  – How analyzed?
Operational Assurance Activities

• Review
  – Interdependencies among applications and systems
  – Runtime operation
  – Technical controls

• Verify documentation
  – Of access permissions
  – Is current and accurate

• Verify proper deregistration
  – i.e. removal of users and privileges

• Is availability and distribution of output products secure?
• Are software & hardware licenses fulfilled and warranties in place?
Other topics: Disposal

- Storage and protection of cryptographic keys
- Legal requirements of records retention
- Archiving federal information
- Sanitize media
Other topics: Procurement Security

Differ based on type of procurement

– Software purchase
  • Commercial Off-The-Shelf (COTS)
  • Custom development

– Outsourcing of services
  • Not just software

– Software as a service
  • e.g. Online Tax Services
Other topics: COTS Software

• Clout is key to gain and keep attention of COTS vendors
  – Big markets get attention: U.S. Government?
  – Major vendors starting to “see the light”

• Security requirements definition in RFP is important
  – Possible product differentiator

• Contract security language
  – Growing importance and emphasis
Other topics: Outsourcing

• Define security goals and policies for software as well as services and hosting
  – Ensure outsourcing maintains the same level of compliance
  – Beware of sub-outsourcing

• Software security and vendor requirements need to be specific and detailed
  – Education of vendors may be necessary
  – Security requirements definition in RFP is important
    • Possible product differentiator
  – Ongoing patching and support is important
Other topics: Software as a service

• Who controls the data?
  – Is security adequate for all types of data?
    • Evaluate controls by mapping to data classification
  – Ensure service maintains compliance with policies and security goals
  – Don't forget e-Discovery
Team Presentations

• Team 1
  – Silas, Vu, James, Blake, Mushima, Sean

• Team 2
  – Stella, Brock, Jose, Anthony, Amanda, Ruslan
Test Taking Tip

Focus on addressing each question individually

• As you take the test, if you don't know an answer, don't obsess over it

• Answer the best way you can or skip over the question and come back to it after you've answered other questions
Quiz