Contents

[Part I Secure Software Concepts 2](#_Toc19890298)

[Ch1. General Security Concepts 2](#_Toc19890299)

[1.1 General Security Concepts 2](#_Toc19890300)

[1.1.1 Security Basics 2](#_Toc19890301)

[1.1.2 System Tenets 2](#_Toc19890302)

[1.1.3 Secure Design Tenets 2](#_Toc19890303)

[Layered Security 2](#_Toc19890304)

[1.2 Security Models 2](#_Toc19890305)

[1.2.1 Access Control Models 2](#_Toc19890306)

[FIGURE 1-2 Using roles to mediate permission assignments 2](#_Toc19890307)

[1.2.2 Multilevel Security Model 2](#_Toc19890308)

[1.2.3 Integrity Models 2](#_Toc19890309)

[1.2.4 Information Flow Models 2](#_Toc19890310)

[FIGURE 1-3 Operational model of security 2](#_Toc19890311)

[1.3 Adversaries 2](#_Toc19890312)

[1.3.1 Adversary Type 2](#_Toc19890313)

[1.3.2 Adversary Groups 2](#_Toc19890314)

[1.3.3 Threat Landscape Shift 2](#_Toc19890315)

[Ch2 Risk Management 2](#_Toc19890316)

[2.1 Definitions and Terminology 2](#_Toc19890317)

[2.1.1 General Terms 2](#_Toc19890318)

[2.1.2 Quantitative Terms 2](#_Toc19890319)

[2.1.3 Risk Management Statements 3](#_Toc19890320)

[FIGURE 2-1 Well-formed risk statement 3](#_Toc19890321)

[2.2 Types of Risk 3](#_Toc19890322)

[2.2.1 Business Risk 3](#_Toc19890323)

[2.2.2 Technology Risk 3](#_Toc19890324)

[2.2.3 Risk Controls 3](#_Toc19890325)

[FIGURE 2-2 Controls framework 3](#_Toc19890326)

[2.2.4 Qualitative Risk Management 3](#_Toc19890327)

[2.2.5 Qualitative Matrix 3](#_Toc19890328)

[Table 2-1 Sample Qualitative Matrix 3](#_Toc19890329)

[2.2.6 Quantitative Risk Management 3](#_Toc19890330)

[FIGURE 2-3 Sample residual risk calculation 3](#_Toc19890331)

[2.2.7 Comparison of Qualitative & Quantitative Methods 3](#_Toc19890332)

[2.3 Governance, Risk, and Compliance 3](#_Toc19890333)

[2.3.1 Regulations and Compliance 3](#_Toc19890334)

[2.3.2 Legal 3](#_Toc19890335)

[2.3.3 Standards 3](#_Toc19890336)

[2.4 Risk Management Models 3](#_Toc19890337)

[2.4.1 General Risk Management Model 3](#_Toc19890338)

[2.4.2 Software Engineering Institute Model 4](#_Toc19890339)

[2.4.3 Model Application 4](#_Toc19890340)

[FIGURE 2-4 Risk complexity versus project size 4](#_Toc19890341)

[2.5 Risk Options 4](#_Toc19890342)

[Ch3 Security Policies & Regulations 4](#_Toc19890343)

[3.1 Regulations and Compliance 4](#_Toc19890344)

[3.1.1 FISMA 4](#_Toc19890345)

[3.1.2 Sarbanes-Oxley 4](#_Toc19890346)

[3.1.3 Gramm-Leach-Bliley 4](#_Toc19890347)

[3.1.4 HIPAA and HITECH 4](#_Toc19890348)

[3.1.5 Payment Card Industry Data Security Standard (PCI DSS) 4](#_Toc19890349)

[3.1.6 Other Regulations 4](#_Toc19890350)

[3.2 Legal Issues 4](#_Toc19890351)

[3.2.1 Intellectual Property 4](#_Toc19890352)

[3.3 Privacy 4](#_Toc19890353)

[3.3.1 Privacy Policy 4](#_Toc19890354)

[3.3.2 Personally Identifiable Information (PII) 4](#_Toc19890355)

[3.3.3 Personal Health Information (PHI) 4](#_Toc19890356)

[3.3.4 Breach Notifications 4](#_Toc19890357)

[3.3.5 Data Protection Principles 4](#_Toc19890358)

[3.3.6 California Consumer Privacy Act 2018 (AB 375) 4](#_Toc19890359)

[3.4 Security Standards 4](#_Toc19890360)

[3.5 Secure Software Architecture 4](#_Toc19890361)

[3.6 Trusted Computing 4](#_Toc19890362)

[3.7 Acquisition 4](#_Toc19890363)

[Ch4 Software Development Methodologies 4](#_Toc19890364)

[4.1 Secure Development Lifecycle 4](#_Toc19890365)

[4.2 Security Features != Secure Software 4](#_Toc19890366)

[4.3 Software Development Models 4](#_Toc19890367)

[4.4 Microsoft Security Development Lifecycle 5](#_Toc19890368)

[Part II Secure Software Requirements 5](#_Toc19890369)

[Ch5 Policy Decomposition 5](#_Toc19890370)

[5.1 Confidentiality, Integrity, Availability Requirements 5](#_Toc19890371)

[5.2 Authentication, Authorization, Auditing Requirements 5](#_Toc19890372)

[5.3 Internal & External Requirements 5](#_Toc19890373)

[Ch6 Data Classification and Categorization 5](#_Toc19890374)

[6.1 Data Classification 5](#_Toc19890375)

[6.2 Data Ownership 5](#_Toc19890376)

[6.3 Labeling 5](#_Toc19890377)

[6.4 Types of Data 5](#_Toc19890378)

[6.5 Data Lifecycle 5](#_Toc19890379)

[Ch7 Requirements 5](#_Toc19890380)

[7.1 Functional Requirements 5](#_Toc19890381)

[7.2 Operational Requirements 5](#_Toc19890382)

[7.3 Requirements Traceability Matrix 5](#_Toc19890383)

[7.4 Connecting the Dots 5](#_Toc19890384)

[Part III Secure Software Design 5](#_Toc19890385)

[Ch8 Design Processes 5](#_Toc19890386)

[8.1 Attack Surface Evaluation 5](#_Toc19890387)

[8.2 Threat Modeling 5](#_Toc19890388)

[8.3 Control Identification & Prioritization 5](#_Toc19890389)

[8.4 Risk Assessment for Code Reuse 5](#_Toc19890390)

[8.5 Documentation 5](#_Toc19890391)

[8.6 Design & Architecture Technical Review 5](#_Toc19890392)

[Ch9 Design Considerations 5](#_Toc19890393)

[9.1 Application of Methods to Address Core Security Concepts 5](#_Toc19890394)

[9.2 Interfaces 6](#_Toc19890395)

[Ch10 Securing Commonly Used Architecture 6](#_Toc19890396)

[10.1 Distributed Computing 6](#_Toc19890397)

[10.2 Service-Oriented Architecture 6](#_Toc19890398)

[10.3 Rich Internet Applications 6](#_Toc19890399)

[10.4 Pervasive/Ubiquitous Computing 6](#_Toc19890400)

[10.5 Mobile Applications 6](#_Toc19890401)

[10.6 Integration with Existing Architectures 6](#_Toc19890402)

[10.7 Cloud Architectures 6](#_Toc19890403)

[Ch11 Technologies 6](#_Toc19890404)

[11.1 Authentication and Identity Management 6](#_Toc19890405)

[11.2 Credential Management 6](#_Toc19890406)

[11.3 Flow Control (Proxies, Firewalls, Middleware) 6](#_Toc19890407)

[11.4 Logging 6](#_Toc19890408)

[11.5 Data Loss Prevention 6](#_Toc19890409)

[11.6 Virtualization 6](#_Toc19890410)

[11.7 Digital Rights Management 6](#_Toc19890411)

[11.8 Trusted Computing 6](#_Toc19890412)

[11.9 Database Security 6](#_Toc19890413)

[11.10 Programming Language Environment 6](#_Toc19890414)

[11.11 Operating Systems 6](#_Toc19890415)

[11.12 Embedded Systems 6](#_Toc19890416)

[Part IV Secure Software Implementation/ Programming 7](#_Toc19890417)

[Ch12 Common Software Vulnerabilities & Countermeasures 7](#_Toc19890418)

[12.1 CWE/SANS Top 25 Vulnerability Categories 7](#_Toc19890419)

[12.2 OWASP Vulnerability Categories 7](#_Toc19890420)

[12.3 Common Vulnerabilities & Countermeasures 7](#_Toc19890421)

[12.4 Input Validation Failures 7](#_Toc19890422)

[12.5 Common Enumerations 7](#_Toc19890423)

[12.6 Virtualization 7](#_Toc19890424)

[12.7 Embedded Systems 7](#_Toc19890425)

[12.8 Side Channel 7](#_Toc19890426)

[12.9 Social Engineering Attacks 7](#_Toc19890427)

[Ch13 Defensive Coding Practices 7](#_Toc19890428)

[13.1 Declarative vs. Programmatic Security 7](#_Toc19890429)

[13.2 Memory Management 7](#_Toc19890430)

[13.3 Error Handling 7](#_Toc19890431)

[13.4 Interface Coding 7](#_Toc19890432)

[13.5 Primary Mitigations 7](#_Toc19890433)

[13.6 Learning from Past Mistakes 7](#_Toc19890434)

[Ch14 Secure Software Coding Operations 7](#_Toc19890435)

[14.1 Code Analysis (Static & Dynamic) 7](#_Toc19890436)

[14.2 Code/Peer Review 7](#_Toc19890437)

[14.3 Build Environment 7](#_Toc19890438)

[14.4 Antitampering Techniques 7](#_Toc19890439)

[14.5 Configuration Management: Source Code & Versioning 7](#_Toc19890440)

[Part V Secure Software Testing 7](#_Toc19890441)

[Ch15 Security Quality Assurance Testing 7](#_Toc19890442)

[15.1 Standards for Software Quality Assurance 7](#_Toc19890443)

[15.2 Testing Methodology 7](#_Toc19890444)

[15.3 Functional Testing 7](#_Toc19890445)

[15.4 Security Testing 7](#_Toc19890446)

[15.5 Environment 7](#_Toc19890447)

[15.6 Attack Surface Validation 8](#_Toc19890448)

[15.7 Testing Artifacts 8](#_Toc19890449)

[15.8 Test Data Lifecycle Management 8](#_Toc19890450)

[Ch16 Security Testing 8](#_Toc19890451)

[16.1 Scanning 8](#_Toc19890452)

[16.2 Penetration Testing 8](#_Toc19890453)

[16.3 Fuzzing 8](#_Toc19890454)

[16.4 Simulation Testing 8](#_Toc19890455)

[16.5 Testing for Failure 8](#_Toc19890456)

[16.6 Cryptographic Validation 8](#_Toc19890457)

[16.7 Regression Testing 8](#_Toc19890458)

[16.8 Impact Assessment and Corrective Action 8](#_Toc19890459)

[Part VI Secure Lifecycle Management 8](#_Toc19890460)

[Ch17 Secure Lifecycle Management 8](#_Toc19890461)

[17.1 Introduction to Acceptance 8](#_Toc19890462)

[17.2 Pre-release Activities 8](#_Toc19890463)

[17.3 Post-release Activities 8](#_Toc19890464)

[Part VII Software Deployment, Operations, Maintenance 8](#_Toc19890465)

[Ch18 Secure Software Installation & Deployment 8](#_Toc19890466)

[18.1 Secure Software Installation & Deployment 8](#_Toc19890467)

[18.2 Configuration Management 8](#_Toc19890468)

[Ch19 Secure Software Operations & Maintenance 8](#_Toc19890469)

[19.1 Secure Software Operations 8](#_Toc19890470)

[19.2 The Software Maintenance Process 8](#_Toc19890471)

[19.3 Secure DevOps 8](#_Toc19890472)

[19.4 Secure Software Disposal 8](#_Toc19890473)

[Part VIII Supply Chain & Software Acquisition 8](#_Toc19890474)

[Ch20 Supply Chain & Software Acquisition 8](#_Toc19890475)

[20.1 Supplier Risk Assessment 8](#_Toc19890476)

[20.2 Supplier Sourcing 8](#_Toc19890477)

[20.3 Software Development & Testing 9](#_Toc19890478)

[20.4 Software Delivery, Operations, Maintenance 9](#_Toc19890479)

[20.5 Supplier Transitioning 9](#_Toc19890480)

# Part I Secure Software Concepts

# Ch1. General Security Concepts

## 1.1 General Security Concepts

### 1.1.1 Security Basics

Attributes (CIA): confidentiality, integrity, availability

Action-Oriented: authentication, authorization, auditing (accounting)

Non-repudiation

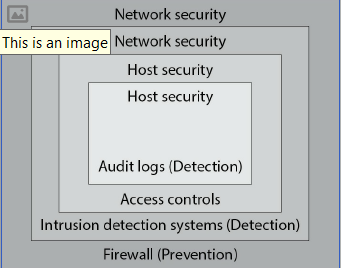
### 1.1.2 System Tenets

Communication between components requires management of a communication session, aka **session management**. Securely managing error conditions is referred to as **exception management**. Software systems require configuration in production: **configuration management** is key in the creation of secure systems.

### 1.1.3 Secure Design Tenets

* Good Enough Security
* Least Privilege
* Separation of Duties
* Defense in Depth: layered security + diversity defense
* Fail-safe
* Economy of Mechanism
* Complete Mediation
* Open Design
* Least Common Mechanism
* Psychological Acceptability
* Weakest Link
* Leverage Existing Components
* Single Point of Failure

#### Layered Security

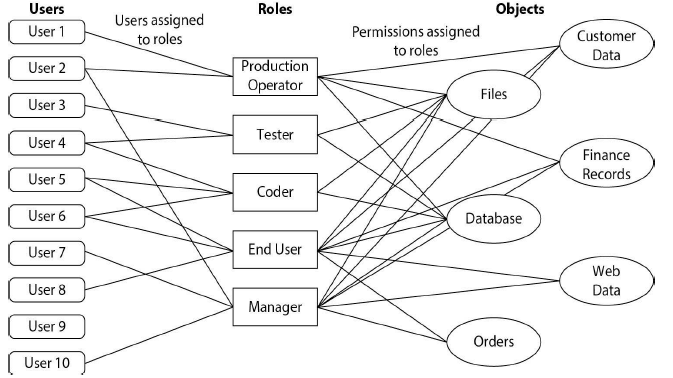


## 1.2 Security Models

### 1.2.1 Access Control Models

* Access Control List (ACL)
* Discretionary Access Control (DAC)
* Mandatory Access Control (MAC)
* Role-based Access Control (RBAC)

#### FIGURE 1-2 Using roles to mediate permission assignments



* Rule-based Access Control (RBAC)
* Attribute-Based Access Control (ABAC)
* Bell-LaPadula Confidentiality Model
* Take-Grant Model

### 1.2.2 Multilevel Security Model

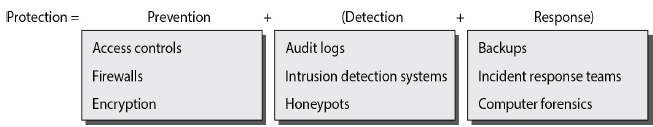
### 1.2.3 Integrity Models

* Biba Integrity Model
* Clark-Wilson Model

### 1.2.4 Information Flow Models

* Brewer-Nash Model (Chinese Wall)
* Data Flow Diagrams
* Use Case Models
* Assurance Models
* NIST CSF Model
* Operational Model of Security

#### FIGURE 1-3 Operational model of security



## 1.3 Adversaries

### 1.3.1 Adversary Type

* Script Kiddie
* Hacker
* Elite

### 1.3.2 Adversary Groups

* Unstructured Threat
* Structured Threat
* Highly Structured Threat
* Nation-state Threat
* Insider vs. Outsider Threat

### 1.3.3 Threat Landscape Shift

# Ch2 Risk Management

## 2.1 Definitions and Terminology

### 2.1.1 General Terms

**Risk** = possibility of suffering harm or loss.

**Inherent risk** = expressed as a product of likelihood and impact, before any controls are introduced to reduce the risk.

**Residual risk** = remains after a control is utilized & reduces the specific risk associated with a vulnerability. Level of risk that must be borne by the entity.

**Total risk** = sum of all risks associated with an asset, a process, or business.

**Risk management** = overall decision-making process of identifying threats and vulnerabilities and their potential impacts, determining the costs to mitigate such events, and deciding what actions are cost effective for controlling these risks.

**Risk assessment** = process of analyzing an environment to identify the risks (threats & vulnerabilities) and mitigating actions to determine (quantitatively/ qualitatively) the impact of an event that would affect a project, program, or business. It is also sometimes referred to as risk analysis.

**Asset** = resource or information an organization needs to conduct business.

**Vulnerability** = characteristic of an asset that can be exploited by a threat to cause harm (failed to instal patches to fix a cross-site scripting (XSS) error)

**Attack** = instance of attempting to perform undesired or unauthorized activities

via a vulnerability.

**Impact** = loss resulting when a threat exploits a vulnerability. A malicious hacker (the threat) uses an XSS tool to hack your unpatched website (the vulnerability), stealing credit card information that is used fraudulently. The credit card company pursues legal recourse against your company to recover the losses from the credit card fraud (the impact).

**Threat** = any circumstance/ event with the potential to cause harm to asset

**Mitigate** = action to reduce likelihood of threat occurring.

**Control** = measure to detect, prevent, mitigate risk associated with threat. The term control is also called countermeasure or safeguard.

**Qualitative risk assessment** = process of subjectively determining the impact of an event that affects a project, program, or business. Completing the qualitative risk assessment usually involves the use of expert judgment, experience, or group consensus to complete the assessment.

### 2.1.2 Quantitative Terms

**Quantitative risk assessment** = process of objectively determining impact of an event that affects a project, program, or business. Use of metrics & models.

**Single loss expectancy (SLE)** = monetary loss/impact of occurrence of a threat.

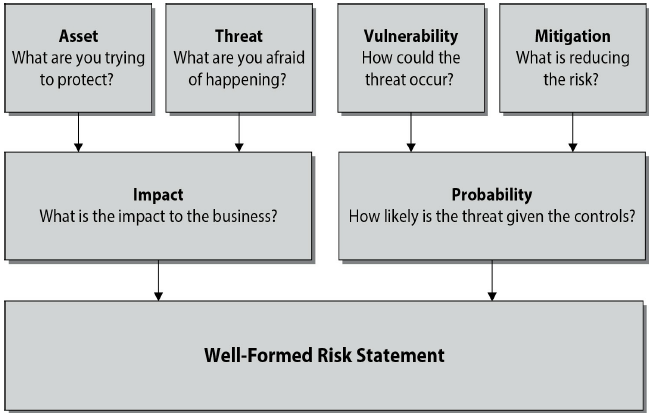
**Exposure factor** = measure of magnitude of loss of an asset. Used in calculation of single loss expectancy.

**Annualized rate of occurrence (ARO)** = frequency with which an event is expected to occur on an annualized basis.

**Annualized loss expectancy (ALE)** = how much event is expected to cost per year

### 2.1.3 Risk Management Statements

#### FIGURE 2-1 Well-formed risk statement



## 2.2 Types of Risk

### 2.2.1 Business Risk

**Treasury management** = Businesses operate as financial enterprises. The

management of company holdings in bonds, futures, currencies, and other financial instruments is a source of financial risk to the business.

**Revenue management** = actions associated with customer behavior and the generation of revenue. As revenue is the lifeblood of business, revenue management is an important area where business risks can affect the enterprise.

**Contract management** = refers to managing contracts with customers, vendors, and partners. Contract management can affect both costs & revenues, and is an important aspect of the financial operation of a business.

**Fraud** = deliberate deception made for personal gain to obtain property

or services, and is a form of business risk.

**Regulatory** Security, privacy, other operation regulations can be sources of business risk. When the regulation effect is related to the technology being employed, it can also be seen as a technological risk.

**Business continuity** Management of risks associated with recovering & restoring business functions after a disaster or major disruption. Software enterprises tend to be highly dependent upon personnel, so issues that impact personnel involved in software development can be viewed as a business continuity risk.

**Technology** creates opportunities for risk, and the employment of technology can be a business risk.

### 2.2.2 Technology Risk

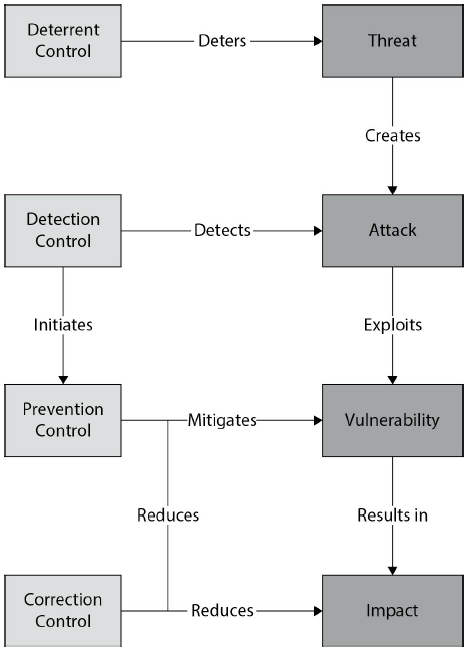
**Security - Privacy** - **Project risk management - Change management**

### 2.2.3 Risk Controls

**Classes of controls**: • Administrative • Technical • Physical

**Types of controls**: • Preventative (deterrent) • Detective • Corrective (recovery) • Compensating

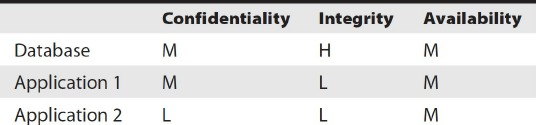
#### FIGURE 2-2 Controls framework



### 2.2.4 Qualitative Risk Management

### 2.2.5 Qualitative Matrix

#### Table 2-1 Sample Qualitative Matrix



* Failure mode effects analysis (FMEA)

### 2.2.6 Quantitative Risk Management

* Annualized Loss Expectancy Model

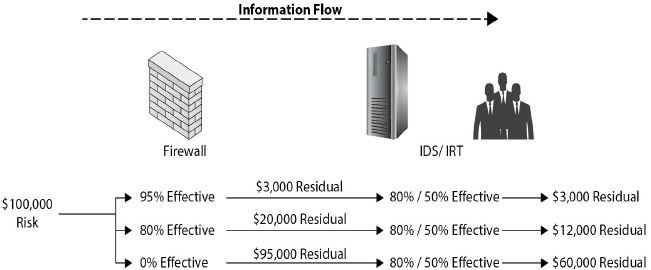
Single Loss Expectancy SLE = asset value \* exposure factor

Annualized Rate of occurrence (ARO) = number of events / number of years

ALE = SLE \* ARO

* Residual Risk Model

#### FIGURE 2-3 Sample residual risk calculation



* Return on Investment (ROI)

ROI (%) = (Avoided Loss – Control Cost) / (Control Cost) \* 100

ROI (Time) = (Avoided Annual Loss) / (Annual Control Cost)

**Example:**

A company owns five web servers, each of which is valued at $100,000 and contributes equally to the company’s capacity. The web servers are geographically spaced at the different regional offices. Each web server provides internal web services to the regional office. The daily value of the content server is calculated at $10,000 to support workers in the office.

Mountain West

SLE = loss \* duration = $10,000 \* 2 = $20,000

ARO = 1/5 = 0.2

ALE = SLE \* ARO = $20,000 \* 0.2 = $4,000

Southeast

SLE = loss \* duration = $10,000 \* 2 = $20,000

ARO = 1/5 = 0.2

ALE = SLE \* ARO = $20,000 \* 0.2 = $4,000

A backup generator costs $40,000, annual maintenance costs $2,000

🡺 Annual cost = $8,000 ($40,000 / 5) + $2,000 annual maintenance cost

= $10,000 total annual cost

For the Mountain West office, ROI has no meaningful value, as it costs more

for the control than the loss it would prevent. For the Southeast Region office,

ROI % = 100% with a payback period of six months.

### 2.2.7 Comparison of Qualitative & Quantitative Methods

## 2.3 Governance, Risk, and Compliance

### 2.3.1 Regulations and Compliance

### 2.3.2 Legal

### 2.3.3 Standards

## 2.4 Risk Management Models

### 2.4.1 General Risk Management Model

Step 1: Asset Identification

Step 2: Threat Assessment: common weaknesses (CWE from mitre.org), SANS Top 25 list, OWASP Top 10 list

Step 3: Impact Determination & Quantification

Step 4: Control Design and Evaluation

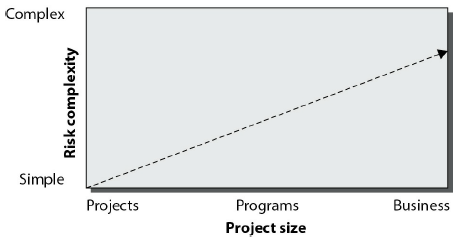
Step 5: Residual Risk Management

### 2.4.2 Software Engineering Institute Model

**1. Identify**: Examine system, enumerating potential risks. **2. Analyze**: Convert risk data into information to make decisions. Evaluate impact, probability, time frame of risks. Classify and prioritize each of the risks. **3. Plan**: Review & evaluate the risks and decide what actions to mitigate. Implement the plan. **4. Track**: Monitor risks & mitigation plans. Trends provide info to activate plans & contingencies. Review periodically to measure progress & identify new risks. **5. Control**: Make corrections for deviations from risk mitigation plans. Correct products & processes as required. Changes in business procedures may require adjustments in plans or actions, as do faulty plans and risks.

### 2.4.3 Model Application

#### FIGURE 2-4 Risk complexity versus project size



## 2.5 Risk Options

# Ch3 Security Policies & Regulations

## 3.1 Regulations and Compliance

### 3.1.1 FISMA

Federal Information Security Management Act of 2002 (FISMA)

Guidelined developed by National Institute of Standards and Technology (NIST): risk management framework (RMF), Information Security Automation Program and the Security Content Automation Protocol (SCAP)

### 3.1.2 Sarbanes-Oxley

Section 404 mandates a specific level of internal control measures

### 3.1.3 Gramm-Leach-Bliley

The Financial Modernization Act of 1999, aka Gramm-Leach-Bliley Act (GLBA): designed to protect consumers’ personal financial information (PFI).

### 3.1.4 HIPAA and HITECH

Healthcare Insurance Portability and Accountability Act (HIPAA) deals with personal health information (PHI). The Health Information Technology for

Economic and Clinical Health Act (HITECH Act) is part of the American Recovery

and Reinvestment Act of 2009 (ARRA),

### 3.1.5 Payment Card Industry Data Security Standard (PCI DSS)

Payment Card Industry PCI standards: Data Security Standard (PCI DSS), Payment Application Data Security Standard (PA DSS), PIN Transaction Security (PTS)

### 3.1.6 Other Regulations

Federal Financial Institutions Examination Council (FFIEC) rules

## 3.2 Legal Issues

### 3.2.1 Intellectual Property

* Patents
* Copyrights
* Trademarks
* Trade Secrets
* Warranty

## 3.3 Privacy

### 3.3.1 Privacy Policy

### 3.3.2 Personally Identifiable Information (PII)

### 3.3.3 Personal Health Information (PHI)

### 3.3.4 Breach Notifications

### 3.3.5 Data Protection Principles

* **Safe Harbor Principles**: allowed non-EU firms to deal with the EUDPD by following 7 elements: •**Notice**: Customers be informed that data is being collected & how it will be used •**Choice**: Customers can opt out of collection & forward transfer of data to 3rd parties •**Onward Transfer**: Transfers of data to 3rd parties may only occur to other organizations that follow adequate data protection principles •**Security**: Reasonable efforts be made to prevent loss of collected information •**Data Integrity**: Data be relevant & reliable for purpose it was collected for •**Access**: Customers be able to access information held about them and correct/delete if inaccurate •**Enforcement**: Effective means of enforcing these rules
* **GDPR Personal Data Elements**: personal data = information relating to identified/ identifiable natural person, including • online identifiers • IP addresses • and cookies

### 3.3.6 California Consumer Privacy Act 2018 (AB 375)

## 3.4 Security Standards

3.4.1 ISO

3.4.2 NIST

## 3.5 Secure Software Architecture

3.5.1 Security Frameworks

## 3.6 Trusted Computing

3.6.1 Principles

3.6.2 Trusted Computing Base

3.6.3 Trusted Platform Module

3.6.4 Microsoft Trustworthy Computing Initiative

## 3.7 Acquisition

3.7.1 Definitions and Terminology

3.7.2 Build vs. Buy Decision

3.7.3 Outsourcing

3.7.4 Contractual Terms and Service Level Agreements

# Ch4 Software Development Methodologies

## 4.1 Secure Development Lifecycle

4.1.1 Principles

4.1.2 Security vs. Quality

## 4.2 Security Features != Secure Software

4.2.1 Secure Development Lifecycle Components

4.2.2 Software Team Awareness and Education

4.2.3 Gates and Security Requirements

4.2.4 Bug Tracking

4.2.5 Threat Modeling

4.2.6 Fuzzing

4.2.7 Security Reviews

4.2.8 Mitigations

## 4.3 Software Development Models

4.3.1 Waterfall

4.3.2 Spiral

4.3.3 Prototype

4.3.4 Agile Methods

4.3.5 Open Source

## 4.4 Microsoft Security Development Lifecycle

4.4.1 History

4.4.2 SDL Foundation

4.4.3 SDL Components

# Part II Secure Software Requirements

# Ch5 Policy Decomposition

## 5.1 Confidentiality, Integrity, Availability Requirements

5.1.1 Confidentiality

5.1.2 Integrity

5.1.3 Availability

## 5.2 Authentication, Authorization, Auditing Requirements

5.2.1 Identification and Authentication

5.2.2 Authorization

5.2.3 Access Control Mechanisms

5.2.4 Auditing

## 5.3 Internal & External Requirements

5.3.1 Internal

5.3.2 External

# Ch6 Data Classification and Categorization

## 6.1 Data Classification

6.1.1 Data States

6.1.2 Data Usage

6.1.3 Data Risk Impact

## 6.2 Data Ownership

6.2.1 Data Owner

6.2.2 Data Custodian

## 6.3 Labeling

6.3.1 Sensitivity

6.3.2 Impact

## 6.4 Types of Data

6.4.1 Structured

6.4.2 Unstructured

## 6.5 Data Lifecycle

6.5.1 Generation

6.5.2 Retention

6.5.3 Disposal

# Ch7 Requirements

## 7.1 Functional Requirements

7.1.1 Role and User Definitions

7.1.2 Objects

7.1.3 Activities/Actions

7.1.4 Subject-Object-Activity Matrix

7.1.5 Use Cases

7.1.6 Abuse Cases (Inside/Outside Adversaries)

7.1.7 Sequencing and Timing

7.1.8 Secure Coding Standards

## 7.2 Operational Requirements

7.2.1 Deployment Environment

## 7.3 Requirements Traceability Matrix

## 7.4 Connecting the Dots

# Part III Secure Software Design

# Ch8 Design Processes

## 8.1 Attack Surface Evaluation

8.1.1 Attack Surface Measurement

8.1.2 Attack Surface Minimization

## 8.2 Threat Modeling

8.2.1 Threat Model Development

## 8.3 Control Identification & Prioritization

## 8.4 Risk Assessment for Code Reuse

## 8.5 Documentation

## 8.6 Design & Architecture Technical Review

# Ch9 Design Considerations

## 9.1 Application of Methods to Address Core Security Concepts

9.1.1 Confidentiality, Integrity, and Availability

9.1.2 Authentication, Authorization, and Auditing

9.1.3 Secure Design Principles

9.1.4 Interconnectivity

## 9.2 Interfaces

# Ch10 Securing Commonly Used Architecture

## 10.1 Distributed Computing

10.1.1 Client Server

10.1.2 Peer-to-Peer

10.1.3 Message Queuing

## 10.2 Service-Oriented Architecture

10.2.1 Enterprise Service Bus

10.2.2 Web Services

## 10.3 Rich Internet Applications

10.3.1 Client-Side Exploits or Threats

10.3.2 Remote Code Execution

## 10.4 Pervasive/Ubiquitous Computing

10.4.1 Wireless

10.4.2 Location-Based

10.4.3 Constant Connectivity

10.4.4 Radio Frequency Identification

10.4.5 Near-Field Communication

10.4.6 Sensor Networks

## 10.5 Mobile Applications

## 10.6 Integration with Existing Architectures

## 10.7 Cloud Architectures

10.7.1 Software as a Service

10.7.2 Platform as a Service

10.7.3 Infrastructure as a Service

# Ch11 Technologies

## 11.1 Authentication and Identity Management

11.1.1 Identity Management

11.1.2 Authentication

## 11.2 Credential Management

11.2.1 X.509 Credentials

11.2.2 Single Sign-On

## 11.3 Flow Control (Proxies, Firewalls, Middleware)

11.3.1 Firewalls

11.3.2 Proxies

11.3.3 Application Firewalls

11.3.4 Queuing Technology

## 11.4 Logging

11.4.1 Syslog

## 11.5 Data Loss Prevention

## 11.6 Virtualization

## 11.7 Digital Rights Management

## 11.8 Trusted Computing

11.8.1 TCB

11.8.2 TPM

11.8.3 Malware

11.8.4 Code Signing

## 11.9 Database Security

11.9.1 Encryption

11.9.2 Triggers

11.9.3 Views

11.9.4 Privilege Management

## 11.10 Programming Language Environment

11.10.1 CLR

11.10.2 JVM

11.10.3 Compiler Switches

11.10.4 Sandboxing

11.10.5 Managed vs. Unmanaged Code

## 11.11 Operating Systems

## 11.12 Embedded Systems

11.12.1 Control Systems

11.12.2 Firmware

# Part IV Secure Software Implementation/ Programming

# Ch12 Common Software Vulnerabilities & Countermeasures

## 12.1 CWE/SANS Top 25 Vulnerability Categories

## 12.2 OWASP Vulnerability Categories

## 12.3 Common Vulnerabilities & Countermeasures

12.3.1 Injection Attacks

12.3.2 Cryptographic Failures

## 12.4 Input Validation Failures

12.4.1 Buffer Overflow

12.4.2 Canonical Form

12.4.3 Missing Defense Functions

12.4.4 General Programming Failures

## 12.5 Common Enumerations

12.5.1 Common Weakness Enumerations (CWE)

12.5.2 Common Vulnerabilities and Exposures (CVE)

## 12.6 Virtualization

## 12.7 Embedded Systems

## 12.8 Side Channel

## 12.9 Social Engineering Attacks

12.9.1 Phishing

# Ch13 Defensive Coding Practices

## 13.1 Declarative vs. Programmatic Security

13.1.1 Bootstrapping

13.1.2 Cryptographic Agility

13.1.3 Handling Configuration Parameters

## 13.2 Memory Management

13.2.1 Type-Safe Practice

13.2.2 Locality

## 13.3 Error Handling

13.3.1 Exception Management

## 13.4 Interface Coding

## 13.5 Primary Mitigations

## 13.6 Learning from Past Mistakes

# Ch14 Secure Software Coding Operations

## 14.1 Code Analysis (Static & Dynamic)

14.1.1 Static

14.1.2 Dynamic

## 14.2 Code/Peer Review

## 14.3 Build Environment

14.3.1 Integrated Development Environment (IDE)

## 14.4 Antitampering Techniques

## 14.5 Configuration Management: Source Code & Versioning

# Part V Secure Software Testing

# Ch15 Security Quality Assurance Testing

## 15.1 Standards for Software Quality Assurance

15.1.1 ISO 9216

15.1.2 SSE-CMM

15.1.3 OSSTMM

## 15.2 Testing Methodology

## 15.3 Functional Testing

15.3.1 Unit Testing

15.3.2 Integration or Systems Testing

15.3.3 Performance Testing

15.3.4 Regression Testing

## 15.4 Security Testing

15.4.1 White-Box Testing

15.4.2 Black-Box Testing

15.4.3 Grey-Box Testing

## 15.5 Environment

15.5.1 Bug Tracking

15.5.2 Defects

15.5.3 Errors

15.5.4 Vulnerabilities

15.5.5 Bug Bar

## 15.6 Attack Surface Validation

## 15.7 Testing Artifacts

## 15.8 Test Data Lifecycle Management

# Ch16 Security Testing

## 16.1 Scanning

16.1.1 Attack Surface Analyzer

## 16.2 Penetration Testing

## 16.3 Fuzzing

## 16.4 Simulation Testing

## 16.5 Testing for Failure

## 16.6 Cryptographic Validation

16.6.1 FIPS 140-2

## 16.7 Regression Testing

## 16.8 Impact Assessment and Corrective Action

# Part VI Secure Lifecycle Management

# Ch17 Secure Lifecycle Management

## 17.1 Introduction to Acceptance

17.1.1 Software Qualification Testing

17.1.2 Qualification Testing Plan

17.1.3 Qualification Testing Hierarchy

## 17.2 Pre-release Activities

17.2.1 Implementing the Pre-release Testing Process

17.2.2 Completion Criteria

17.2.3 Risk Acceptance

## 17.3 Post-release Activities

17.3.1 Validation and Verification

17.3.2 Independent Testing

# Part VII Software Deployment, Operations, Maintenance

# Ch18 Secure Software Installation & Deployment

## 18.1 Secure Software Installation & Deployment

18.1.1 Installation Validation and Verification

18.1.2 Planning for Operational Use

## 18.2 Configuration Management

18.2.1 Organizing the Configuration Management Process

18.2.2 Configuration Management Roles

18.2.3 The Configuration Management Plan

18.2.4 The Configuration Management Process

# Ch19 Secure Software Operations & Maintenance

## 19.1 Secure Software Operations

19.1.1 Operations Process Implementation

## 19.2 The Software Maintenance Process

19.2.1 Monitoring

19.2.3 Incident Management

19.2.4 Problem Management

19.2.5 Change Management

19.2.6 Backup, Recovery, Archiving

## 19.3 Secure DevOps

## 19.4 Secure Software Disposal

19.4.1 Software Disposal Planning

19.4.2 Software Disposal Execution

# Part VIII Supply Chain & Software Acquisition

# Ch20 Supply Chain & Software Acquisition

## 20.1 Supplier Risk Assessment

20.1.1 What Is Supplier Risk Assessment?

20.1.2 Risk Assessment for Code Reuse

20.1.3 Intellectual Property

20.1.4 Legal Compliance

20.1.5 Supplier Prequalification

## 20.2 Supplier Sourcing

20.2.1 Contractual Integrity Controls

20.2.2 Vendor Technical Integrity Controls for Suppliers

20.2.3 Managed Services

20.2.4 Service Level Agreements

## 20.3 Software Development & Testing

20.3.1 Code Testing

20.3.2 Security Testing Controls

20.3.3 Software Requirements Testing/ Validation

20.3.4 Software Requirements Testing/ Validation for

Subcontractors

## 20.4 Software Delivery, Operations, Maintenance

20.4.1 Chain of Custody

20.4.2 Publishing and Dissemination Controls

20.4.3 System-of-Systems Integration

20.4.4 Software Authenticity & Integrity

20.4.5 Product Deployment & Sustainment Controls

20.4.6 Monitoring and Incident Management

20.4.7 Vulnerability Management, Tracking, Resolution

## 20.5 Supplier Transitioning