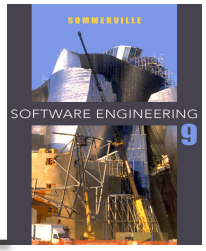
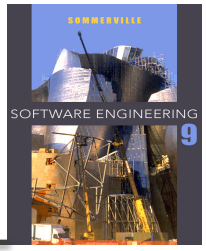


Chapter 14 – Security Engineering

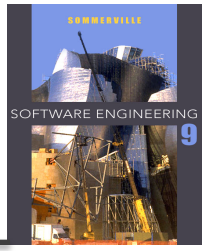
Lecture 1

Topics covered



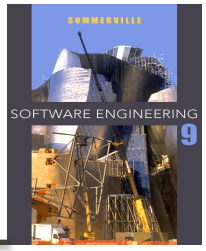
- Security engineering and security management
  - Security engineering concerned with applications; security management with infrastructure.
- Security risk assessment
  - Designing a system based on the assessment of security risks.
- Design for security
  - How system architectures have to be designed for security.

Security engineering



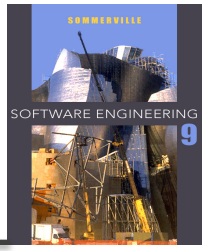
- Tools, techniques and methods to support the development and maintenance of systems that can resist malicious attacks that are intended to damage a computer-based system or its data.
- A sub-field of the broader field of computer security.
- Assumes background knowledge of dependability and security concepts (Chapter 10) and security requirements specification (Chapter 12)

Application/infrastructure security

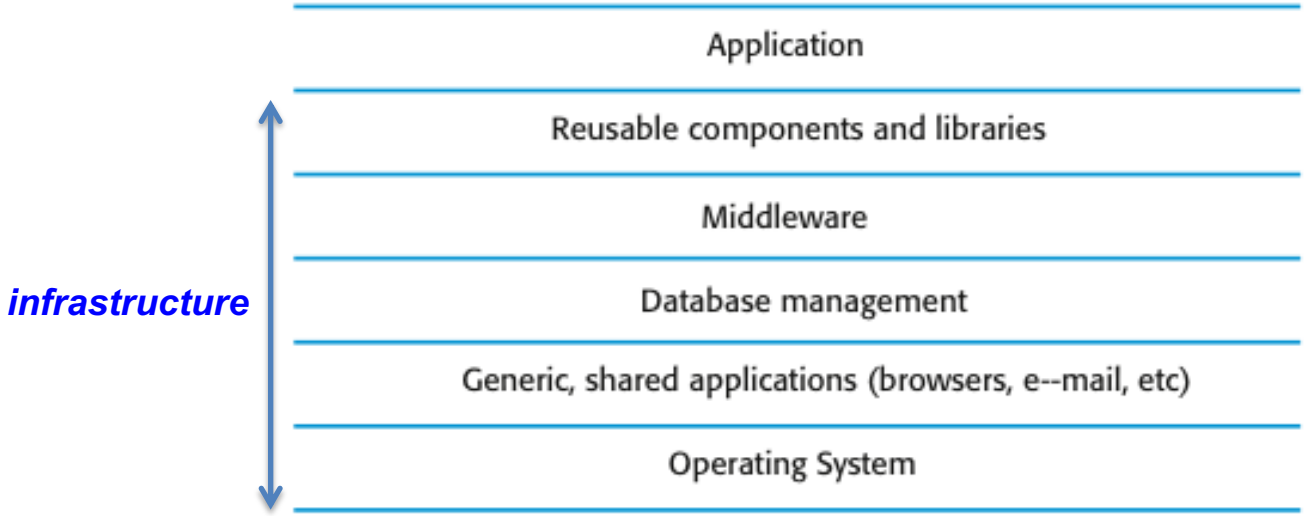


- Application security is a software engineering problem where the system is designed to resist attacks.
- Infrastructure security is a systems management problem where the infrastructure is configured to resist attacks.
- The focus of this chapter is application security.

System layers where security may be compromised

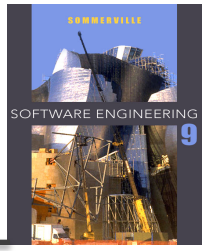


When you consider security issues, you have to consider both the application software and the infrastructure on which this system is built.



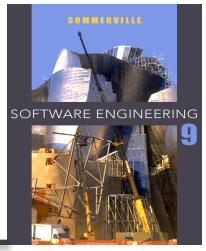
Middleware *supports distributed computing and database access.*

System security management



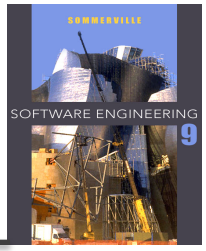
- User and permission management
  - Adding and removing users from the system and setting up appropriate permissions for users
- Software deployment and maintenance
  - Installing application software and middleware and configuring these systems so that vulnerabilities are avoided.
- Attack monitoring, detection and recovery
  - Monitoring the system for unauthorized access, design strategies for resisting attacks and develop backup and recovery strategies.

14.1 Security risk management



- Risk management is concerned with assessing the possible losses that might ensue from attacks on the system and balancing these losses against the costs of security procedures that may reduce these losses.
- Risk management should be driven by an organisational security policy.
- Risk management involves
  - Preliminary risk assessment
  - Life cycle risk assessment
  - Operational risk assessment

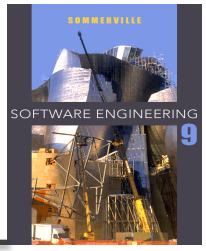
Preliminary risk assessment



Preliminary risk assessment focuses on deriving security requirements.

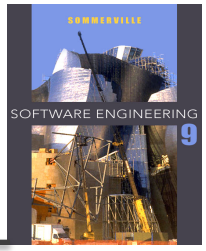
- At this stage, decisions on the detailed system requirements, the system design, or the implementation technology have not been made.
- The aim of this assessment process is to decide if an adequate level of security can be achieved at a reasonable cost. If this is the case, you can then derive specific security requirements for the system.
- You do not have information about potential vulnerabilities in the system or the controls that are included in reused system components or middleware.

Misuse cases



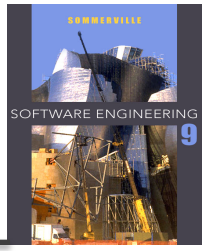
- Misuse cases are scenarios that represent malicious interactions with a system(instances of threats to a system). Pfleeger C. P. and Pfleeger S. L.(2007) characterize threats under four headings, which may be used as a starting point for identifying possible misuse cases. These headings are:
  - Interception threats
    - Attacker gains access to an asset
  - Interruption threats
    - Attacker makes part of a system unavailable
  - Modification threats
    - Allow an attacker to tamper with a system asset
  - Fabrication threats
    - False information is added to a system

Asset analysis



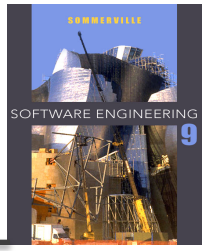
Asset	Value	Exposure
The information system	High. Required to support all clinical consultations. Potentially safety-critical.	High. Financial loss as clinics may have to be cancelled. Costs of restoring system. Possible patient harm if treatment cannot be prescribed.
The patient database	High. Required to support all clinical consultations. Potentially safety-critical.	High. Financial loss as clinics may have to be canceled. Costs of restoring system. Possible patient harm if treatment cannot be prescribed.
An individual patient record	Normally low although may be high for specific high-profile patients.	Low direct losses but possible loss of reputation.

Threat and control analysis



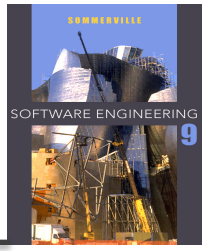
Threat	Probability	Control	Feasibility
Unauthorized user gains access as system manager and makes system unavailable	Low	Only allow system management from specific locations that are physically secure.	Low cost of implementation but care must be taken with key distribution and to ensure that keys are available in the event of an emergency.
Unauthorized user gains access as system user and accesses confidential information	High	Require all users to authenticate themselves using a biometric mechanism. Log all changes to patient information to track system usage.	Technically feasible but high-cost solution. Possible user resistance. Simple and transparent to implement and also supports recovery.

Security requirements



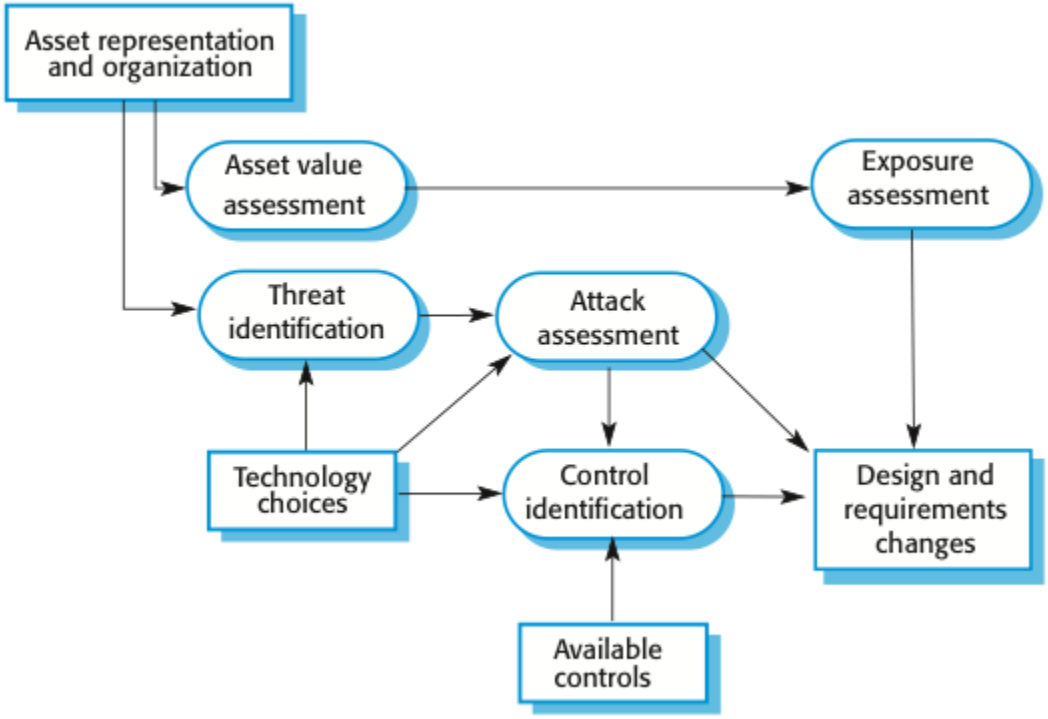
- Patient information must be downloaded at the start of a clinic session to a secure area on the system client that is used by clinical staff.
- Patient information must not be maintained on system clients after a clinic session has finished.
- A log on a separate computer from the database server must be maintained of all changes made to the system database.

14.1.1 Life cycle risk assessment

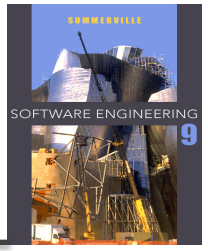


- Risk assessment while the system is being developed and after it has been deployed
- More information is available - system platform, middleware and the system architecture and data organisation.
- Vulnerabilities that arise from design choices may therefore be identified.

Life-cycle risk analysis



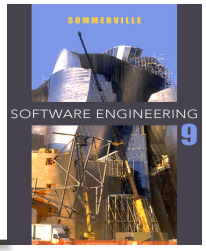
Design decisions from use of COTS



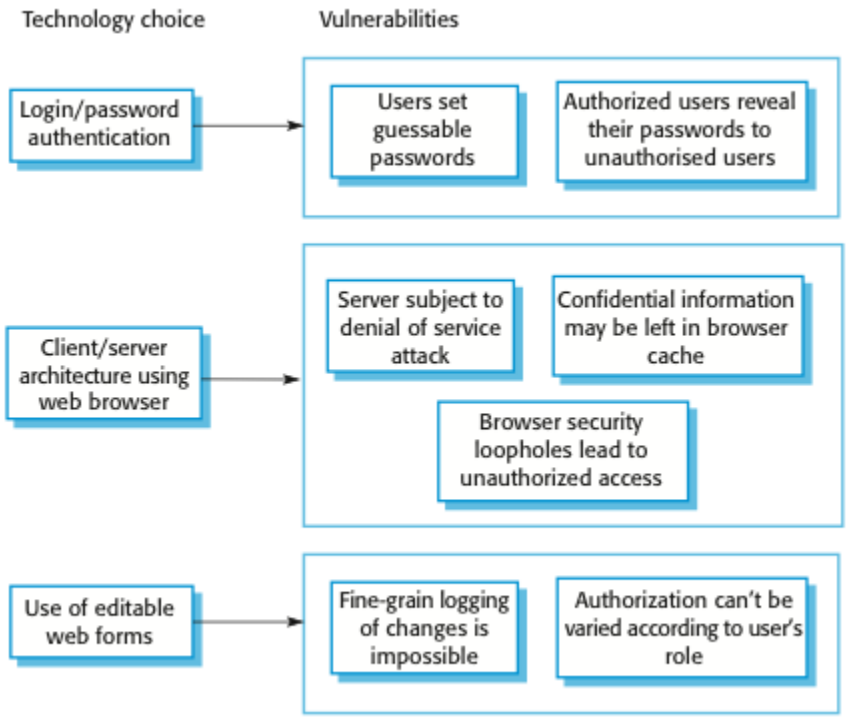
When you develop an application by reusing an existing system, you have to accept the design decisions made by the developers of that system.
Let us assume that some of these decisions are as follows:

- System users are authenticated using a name/password combination.
- The system architecture is client-server, with clients accessing data through a standard web browser.
- Information is presented to users as an editable web form. They can change information in place and upload the revised information to the server.

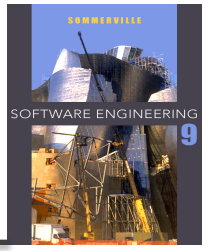
Vulnerabilities associated with technology choices



For a generic system, the above design decisions are perfectly acceptable, but a lifecycle risk analysis reveals that they have associated vulnerabilities. Examples of possible vulnerabilities are shown in the following figure:



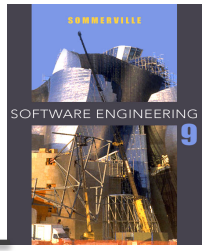
Security requirements



Some(not all) examples of requirements proposed to address the inherent vulnerabilities might be the following:

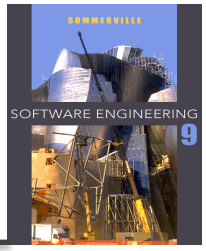
- A password checker program shall be made available and shall be run daily. Weak passwords shall be reported to system administrators.
- Access to the system shall only be allowed by approved client computers.
- All client computers shall have a single, approved web browser installed by system administrators.

14.1.2 Operational risk assessment



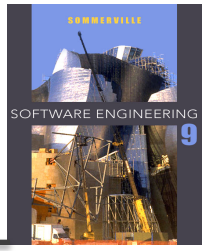
- Continuation of life cycle risk assessment but with additional information about the environment where the system is used.
- Environment characteristics can lead to new system risks
  - For example, say a system is being used in an environment in which users are frequently interrupted. Risk of interruption means that logged in computers are left unattended.
  - It may be possible for an unauthorized person to gain access to the information in the system.
  - This could then generate a requirement for a password-protected screen saver to be run after a short period of inactivity.

14.2 Design for security



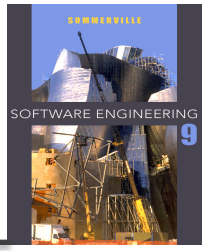
- Architectural design
  - how do architectural design decisions affect the security of a system?
- Good practice
  - what is accepted good practice when designing secure systems?
- Design for deployment
  - what support should be designed into a system to avoid the introduction of vulnerabilities when a system is deployed for use?

14.2.1 Architectural design



- Two fundamental issues have to be considered when designing an architecture for security.
  - Protection
    - How should the system be organised so that critical assets can be protected against external attack?
  - Distribution
    - How should system assets be distributed so that the effects of a successful attack are minimized?
- These are potentially conflicting
  - If assets are distributed, then they are more expensive to protect. If assets are protected, then usability and performance requirements may be compromised.

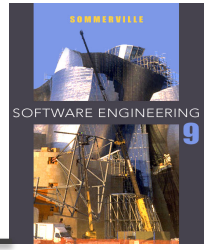
Protection

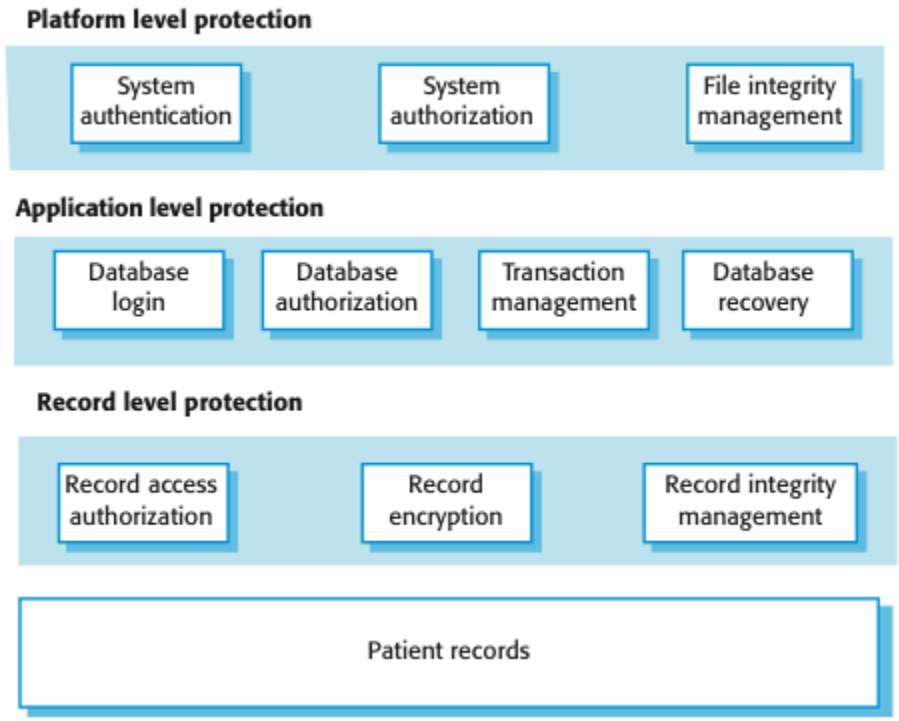


- Platform-level protection
  - Top-level controls on the platform on which a system runs.
- Application-level protection
  - Specific protection mechanisms built into the application itself e.g. additional password protection.
- Record-level protection
  - Protection that is invoked when access to specific information is requested

These lead to a layered protection architecture shown in next slide.

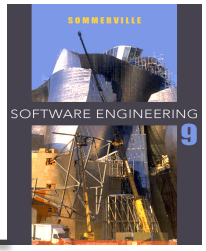
A layered protection architecture





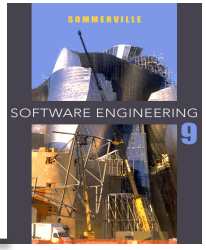
A layered protection architecture

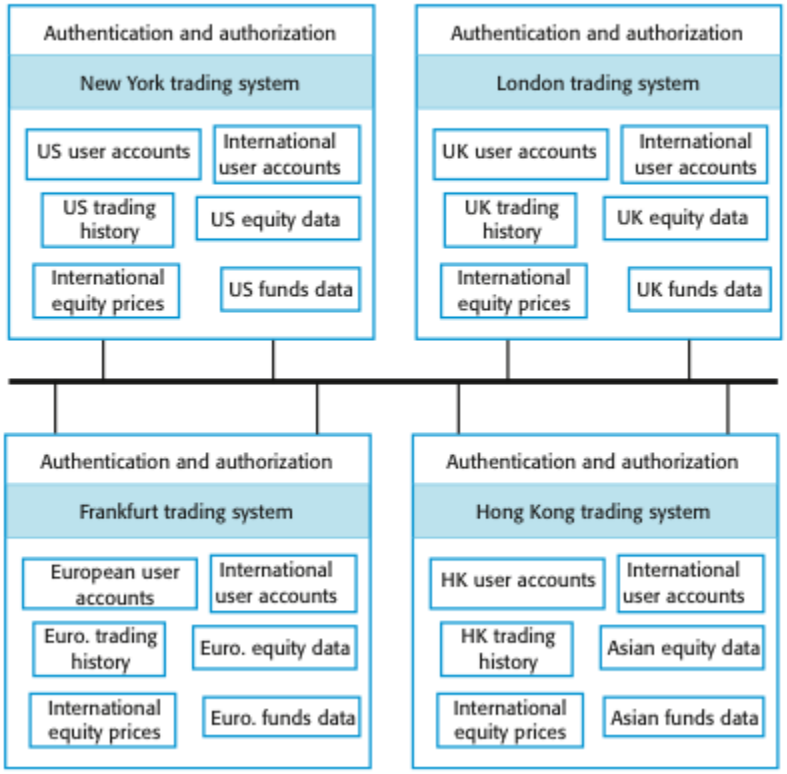
Distribution



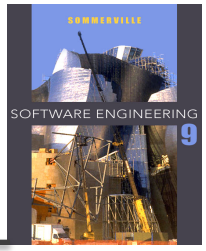
- Distributing assets means that attacks on one system do not necessarily lead to complete loss of system service
- Each platform has separate protection features and may be different from other platforms so that they do not share a common vulnerability
- Distribution is particularly important if the risk of denial of service attacks is high

Distributed assets in an equity trading system

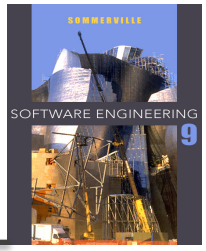




Key points



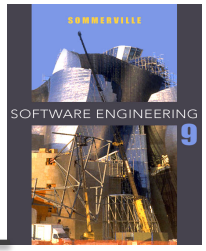
- Security engineering is concerned with how to develop systems that can resist malicious attacks
- Security threats can be threats to confidentiality, integrity or availability of a system or its data
- Security risk management is concerned with assessing possible losses from attacks and deriving security requirements to minimise losses
- Design for security involves architectural design, following good design practice and minimising the introduction of system vulnerabilities



Chapter 14 – Security Engineering

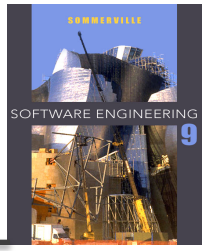
Lecture 2

Topics covered



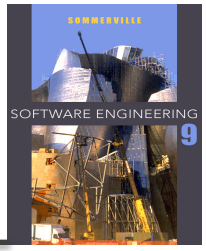
- Design guidelines for security
  - Guidelines that help you design a secure system
- Design for deployment
  - Design so that deployment problems that may introduce vulnerabilities are minimized
- System survivability
  - Allow the system to deliver essential services when under attack

14.2.2 Design guidelines for security engineering



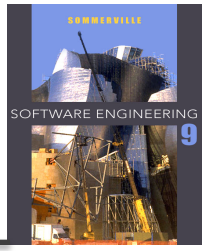
- Design guidelines encapsulate good practice in secure systems design
- Design guidelines serve two purposes:
  - They raise awareness of security issues in a software engineering team. Security is considered when design decisions are made.
  - They can be used as the basis of a review checklist that is applied during the system validation process.
- Design guidelines here are applicable during software specification and design

Design guidelines for secure systems engineering



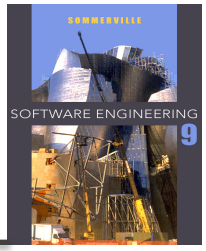
Security guidelines	
Base security decisions on an explicit security policy	
Avoid a single point of failure	
Fail securely	
Balance security and usability	
Log user actions	
Use redundancy and diversity to reduce risk	
Validate all inputs	
Compartmentalize your assets	
Design for deployment	
Design for recoverability	

Design guidelines 1-3



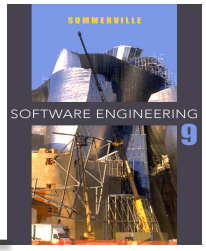
- Base decisions on an explicit security policy
  - Define a security policy for the organization that sets out the fundamental security requirements that should apply to all organizational systems.
- Avoid a single point of failure
  - Ensure that a security failure can only result when there is more than one failure in security procedures. For example, have password and question-based authentication.
- Fail securely
  - When systems fail, for whatever reason, ensure that sensitive information cannot be accessed by unauthorized users even although normal security procedures are unavailable.

Design guidelines 4-6



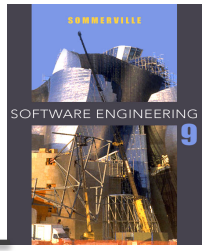
- Balance security and usability
  - Try to avoid security procedures that make the system difficult to use. Sometimes you have to accept weaker security to make the system more usable.
- Log user actions
  - Maintain a log of user actions that can be analyzed to discover who did what. If users know about such a log, they are less likely to behave in an irresponsible way.
- Use redundancy and diversity to reduce risk
  - Keep multiple copies of data and use diverse infrastructure so that an infrastructure vulnerability cannot be the single point of failure.

Design guidelines 7-10



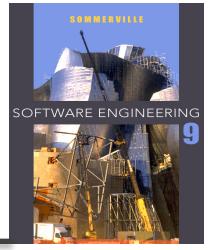
- Validate all inputs
  - Check that all inputs are within range so that unexpected inputs cannot cause problems.
- Compartmentalize your assets
  - Organize the system so that assets are in separate areas and users only have access to the information that they need rather than all system information.
- Design for deployment
  - Design the system to avoid deployment problems
- Design for recoverability
  - Design the system to simplify recoverability after a successful attack.

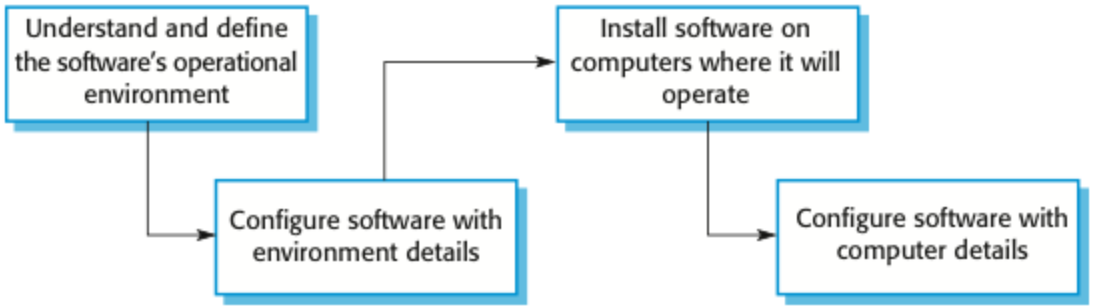
14.2.3 Design for deployment



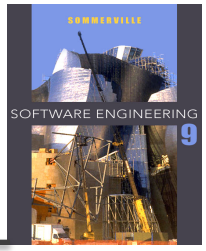
- Deployment involves configuring software to operate in its working environment, installing the system and configuring it for the operational platform.
- Vulnerabilities may be introduced at this stage as a result of configuration mistakes.
- Designing deployment support into the system can reduce the probability that vulnerabilities will be introduced.

Software deployment



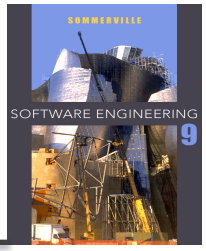


Configuration vulnerabilities



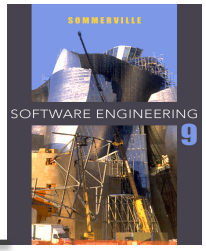
- Vulnerable default settings
  - Attackers can find out the default settings for software. If these are weak (often to increase usability) then they can be exploited by users when attacking a system.
- Development rather than deployment
  - Some configuration settings in systems are designed to support development and debugging. If these are not turned off, they can be a vulnerability that can be exploited by attackers.

Deployment support 1



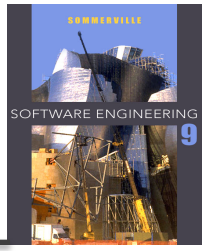
- Include support for viewing and analyzing configurations
  - Make sure that the system administrator responsible for deployment can easily view the entire configuration. This makes it easier to spot omissions and errors that have been made.
- Minimize default privileges and thus limit the damage that might be caused
  - Design the system so that the default privileges for an administrator are minimized. This means that if someone gains admin access, they do not have immediate access to the features of the system.

Deployment support 2



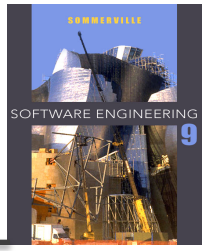
- Localize configuration settings
  - When setting up a system, all information that is relevant to the same part or component of a system should be localized so that it is all set up at once. Otherwise, it is easy to forget to set up related security features.
- Provide easy ways to fix security vulnerabilities
  - When problems are detected, provide easy ways, such as auto-updating, to repair security vulnerabilities in the deployed systems.

14.3 System survivability



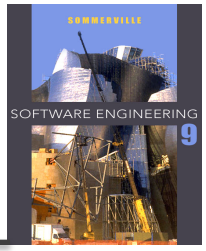
- Survivability is an emergent system property that reflects the systems ability to deliver essential services whilst it is under attack or after part of the system has been damaged
- Survivability analysis and design should be part of the security engineering process

Importance of survivability



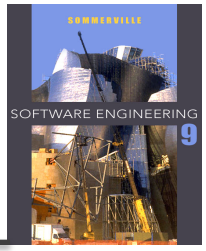
- Our economic and social lives are dependent on computer systems
  - Critical infrastructure – electricity, gas, telecommunications, transport
  - Healthcare
  - Government
- Loss of business systems for even a short time can have very severe economic effects
  - Airline reservation systems
  - E-commerce systems
  - Payment systems

Service availability



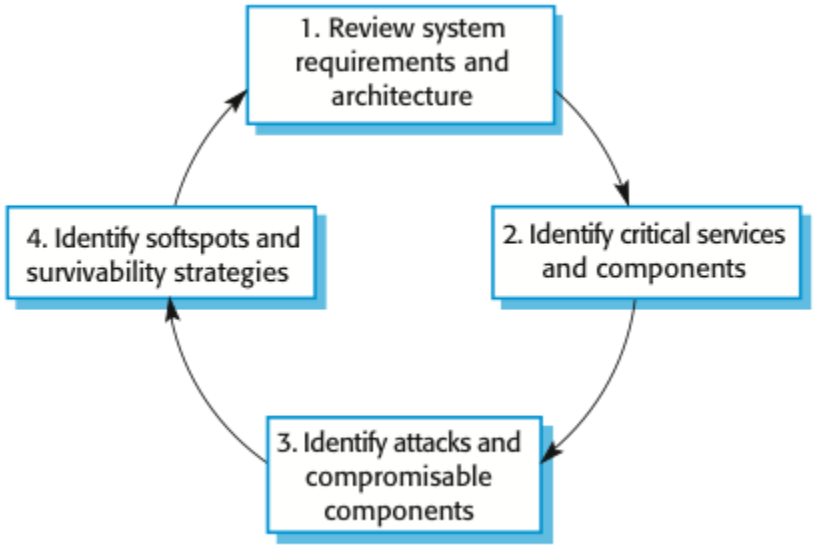
- Which system services are the most critical for a business?
- How might these services be compromised?
- What is the minimal quality of service that must be maintained?
- How can these services be protected?
- If a service becomes unavailable, how quickly can it be recovered?

Survivability strategies

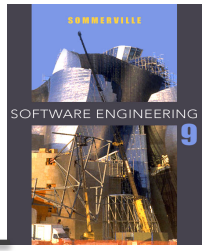


- Resistance
  - Avoiding problems by building capabilities into the system to resist attacks
- Recognition
  - Detecting problems by building capabilities into the system to detect attacks and failures and assess the resultant damage
- Recovery
  - Tolerating problems by building capabilities into the system to deliver services whilst under attack

Stages in survivability analysis

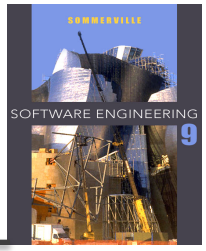


Key activities



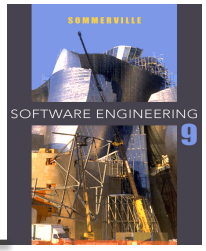
- System understanding
  - Review goals, requirements and architecture
- Critical service identification
  - Identify services that must be maintained
- Attack simulation
  - Devise attack scenarios and identify components affected
- Survivability analysis
  - Identify survivability strategies to be applied

Trading system survivability



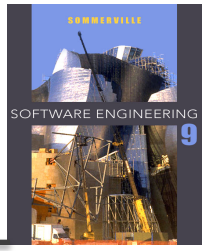
- User accounts and equity prices replicated across servers so some provision for survivability made
- Key capability to be maintained is the ability to place orders for stock
- Orders must be accurate and reflect the actual sales/purchases made by a trader

Survivable ordering service



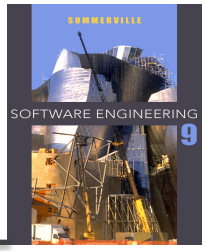
- The critical service that must survive is the ability for authorized users to place orders for stock
- This requires 3 components of the system to be available and operating reliability:
  - User authentication, allowing authorized users to log on to the system
  - Price quotation, allowing buying and selling prices to be quoted
  - Order placement, allowing buy and sell orders to be made

Possible attacks



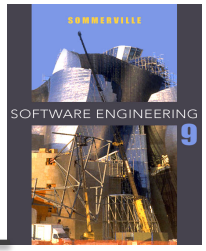
- Malicious user masquerades as a legitimate user and places malicious orders for stock, with the aim of causing problems for the legitimate user
- An unauthorized user corrupts the database of transactions thus making reconciliation of sales and purchases impossible

Survivability analysis in an equity trading system



Attack	Resistance	Recognition	Recovery
Unauthorized user places malicious orders	Require a dealing password that is different from the login password to place orders.	Send copy of order by e-mail to authorized user with contact phone number (so that they can detect malicious orders). Maintain user's order history and check for unusual trading patterns.	Provide mechanism to automatically 'undo' trades and restore user accounts. Refund users for losses that are due to malicious trading. Insure against consequential losses.
Corruption of transactions database	Require privileged users to be authorized using a stronger authentication mechanism, such as digital certificates.	Maintain read-only copies of transactions for an office on an international server. Periodically compare transactions to check for corruption. Maintain cryptographic checksum with all transaction records to detect corruption.	Recover database from backup copies. Provide a mechanism to replay trades from a specified time to re-create the transactions database.

Key points



- General security guidelines sensitize designers to security issues and serve as review checklists
- Configuration visualization, setting localization, and minimization of default privileges help reduce deployment errors
- System survivability reflects the ability of a system to deliver services whilst under attack or after part of the system has been damaged.