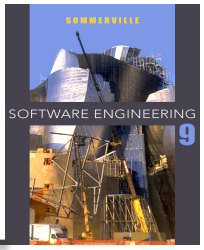
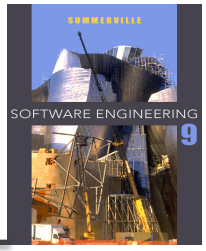


Chapter 11 – Security and Dependability

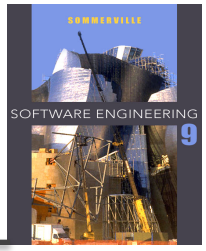
Lecture 1

Topics covered



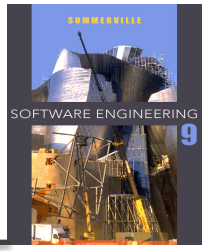
- ✧ Dependability properties
  - The system attributes that lead to dependability.
- ✧ Availability and reliability
  - Systems should be available to deliver service and perform as expected.
- ✧ Safety
  - Systems should not behave in an unsafe way.
- ✧ Security
  - Systems should protect themselves and their data from external interference.

System dependability



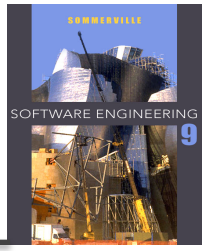
- For many computer-based systems, the most important system property is the dependability of the system.
- The dependability of a system reflects the user's degree of trust in that system. It reflects the extent of the user's confidence that it will operate as users expect and that it will not 'fail' in normal use.
- Dependability covers the related systems attributes of reliability, availability and security. These are all inter-dependent.

Importance of dependability



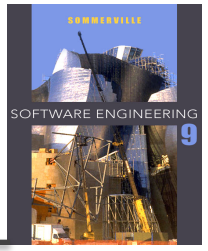
- System failures may have widespread effects with large numbers of people affected by the failure.
- Systems that are not dependable and are unreliable, unsafe or insecure may be rejected by their users.
- The costs of system failure may be very **high** if the failure leads to economic losses or physical damage.
- Undependable systems may cause information loss with a high consequent recovery cost.

Causes of failure

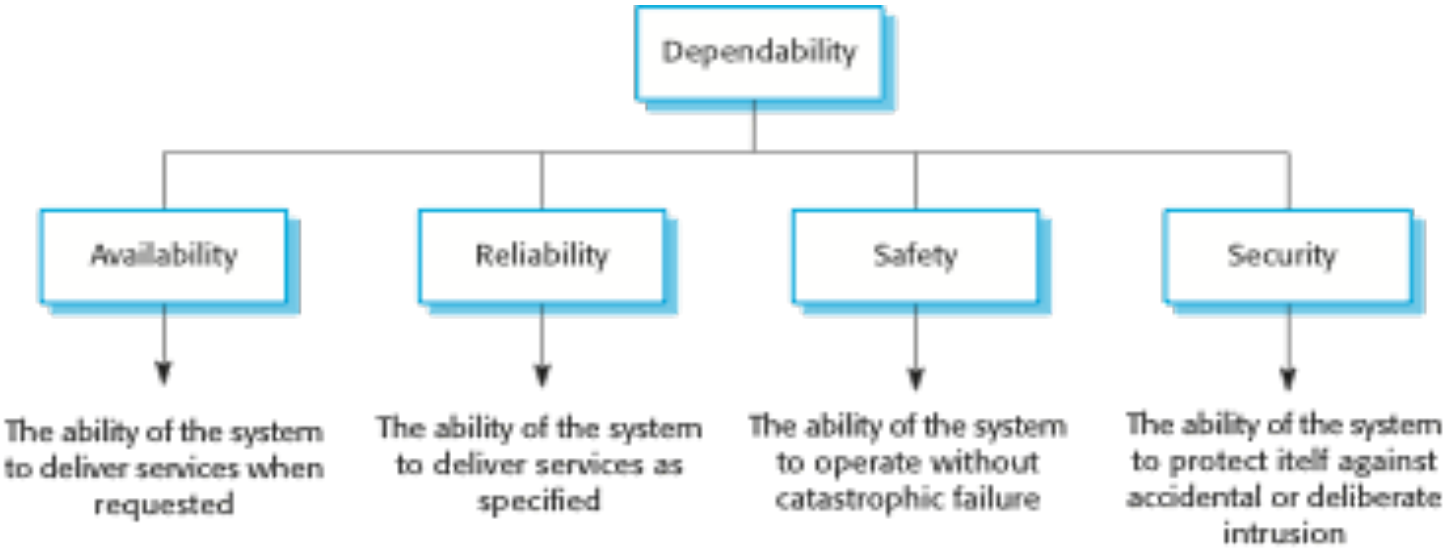


- Hardware failure
  - Hardware fails because of design and manufacturing errors or because components have reached the end of their natural life.
- Software failure
  - Software fails due to errors in its specification, design or implementation.
- Operational failure
  - Human operators make mistakes. Now perhaps the largest single cause of system failures in socio-technical systems.

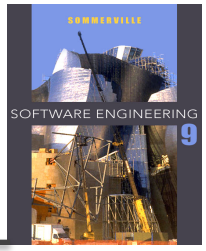
Principal dependability properties



- Availability: The probability that the system will be up and running and able to deliver **useful** services to users.
- Reliability: The probability that the system will correctly deliver services as expected by users.
- Safety: A judgment of how likely it is that the system will cause damage to people or its environment.
- Security: A judgment of how likely it is that the system can resist accidental or deliberate intrusions.

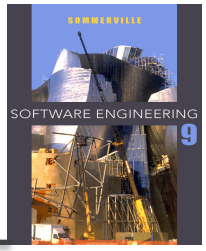


Other dependability properties



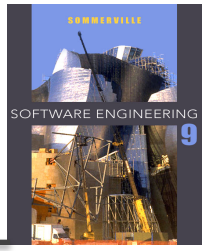
- Repairability
  - Reflects the extent to which the system can be repaired in the event of a failure
- Maintainability
  - Reflects the extent to which the system can be adapted to new requirements;
- Survivability
  - Reflects the extent to which the system can deliver services whilst under hostile attack;
- Error tolerance
  - Reflects the extent to which user input errors can be avoided and tolerated.

Dependability attribute dependencies



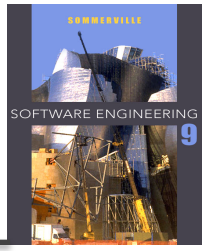
- Safe system operation depends on the system being available and operating reliably.
- A system may be unreliable because its data has been corrupted by an external attack.
- Denial of service attacks on a system are intended to make it unavailable.
- If a system is infected with a virus, you cannot be confident in its reliability or safety.

Dependability achievement



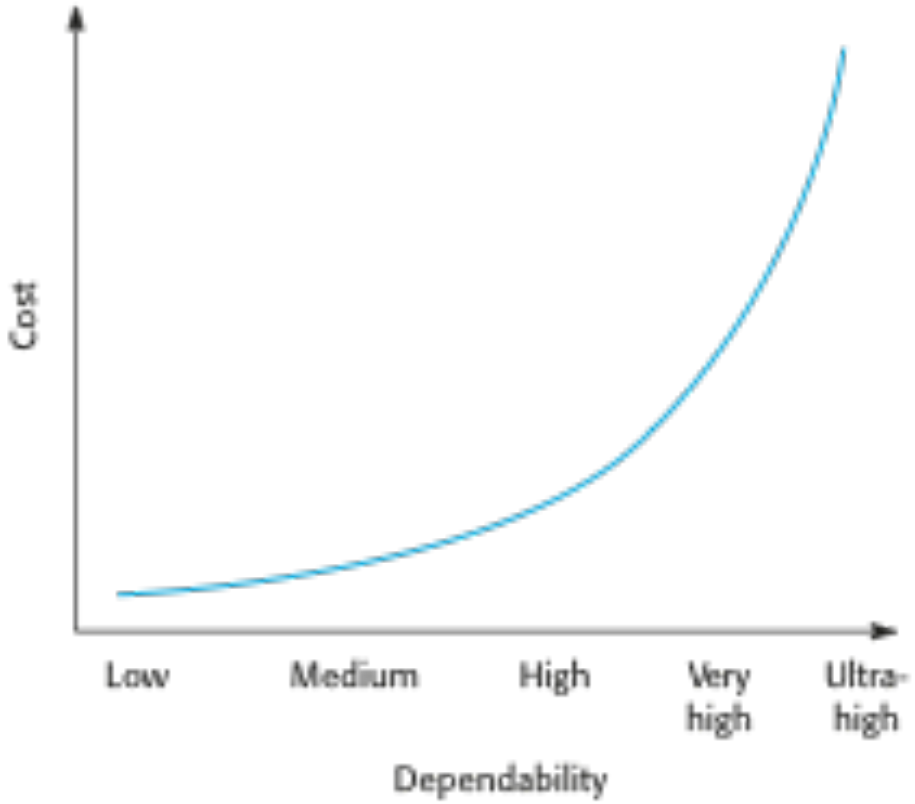
- Avoid the introduction of accidental errors when developing the system.
- Design V & V processes that are effective in discovering residual errors in the system.
- Design protection mechanisms that guard against external attacks.
- Configure the system correctly for its operating environment.
- Include recovery mechanisms to help restore normal system service after a failure.

Dependability costs

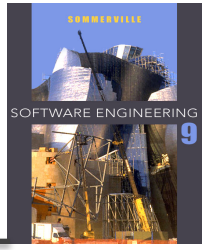


- Dependability costs tend to increase exponentially as increasing levels of dependability are required.
- There are two reasons for this
  - The use of more expensive development techniques and hardware that are required to achieve the higher levels of dependability.
  - The increased testing and system validation that is required to convince the system client and regulators that the required levels of dependability have been achieved.

Cost/dependability curve

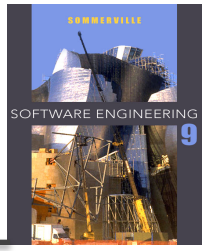


Availability and reliability



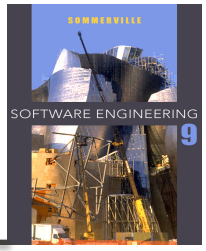
- Reliability
  - The probability of failure-free system operation over a specified time in a given environment for a given purpose
- Availability
  - The probability that a system, at a point in time, will be operational and able to deliver the requested services
- Both of these attributes can be expressed quantitatively e.g. availability of 0.999 means that the system is up and running for 99.9% of the time.

Availability and reliability



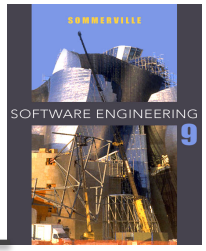
- It is sometimes possible to subsume system availability under system reliability
  - Obviously if a system is unavailable it is not delivering the specified system services.
- However, it is possible to have systems with low reliability that must be available.
  - So long as system failures can be repaired quickly and does not damage data, some system failures may not be a problem.
- Availability is therefore best considered as a separate attribute reflecting whether or not the system can deliver its services.
- Availability takes repair time into account, if the system has to be taken out of service to repair faults.

Perceptions of reliability



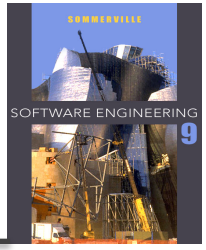
- The formal definition of reliability does not always reflect the user's perception of a system's reliability
  - The assumptions that are made about the environment where a system will be used may be incorrect
    - Usage of a system in an office environment is likely to be quite different from usage of the same system in a university environment
  - The consequences of system failures affects the perception of reliability
    - Unreliable windscreen wipers in a car may be irrelevant in a dry climate
    - Failures that have serious consequences (such as an engine breakdown in a car) are given greater weight by users than failures that are inconvenient

Reliability and specifications



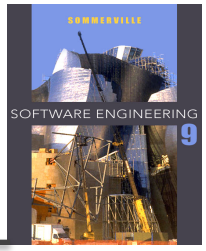
- Reliability can only be defined formally with respect to a system specification i.e. a failure is a deviation from a specification.
- However, many specifications are incomplete or incorrect – hence, a system that conforms to its specification may 'fail' from the perspective of system users.
- Furthermore, users don't read specifications so don't know how the system is supposed to behave.
- Therefore perceived reliability is more important in practice.

Availability perception



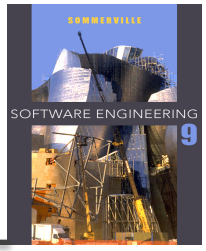
- Availability is usually expressed as a percentage of the time that the system is available to deliver services e.g. 99.95%.
- However, this does not take into account two factors:
  - The number of users affected by the service outage. Loss of service in the middle of the night is less important for many systems than loss of service during peak usage periods.
  - The length of the outage. The longer the outage, the more the disruption. Several short outages are less likely to be disruptive than 1 long outage. Long repair times are a particular problem.

Reliability terminology



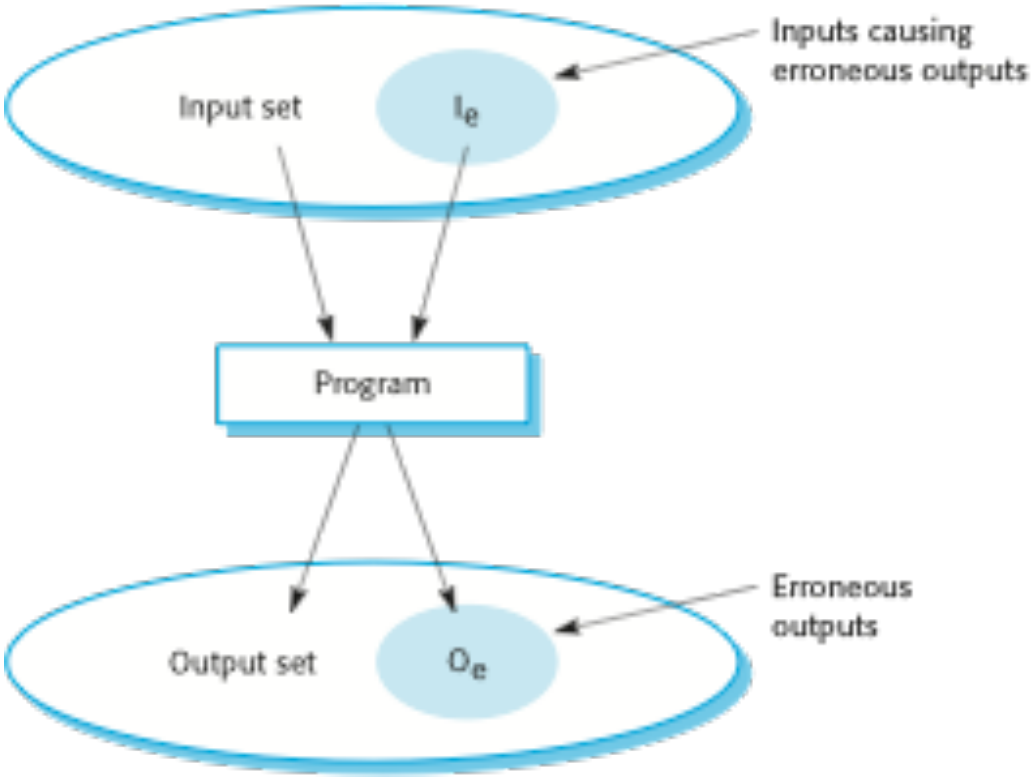
Term	Description
Human error or mistake	Human behavior that results in the introduction of faults into a system. For example, in the wilderness weather system, a programmer might decide that the way to compute the time for the next transmission is to add 1 hour to the current time. This works except when the transmission time is between 23.00 and midnight (midnight is 00.00 in the 24-hour clock).
System fault	A characteristic of a software system that can lead to a system error. The fault is the inclusion of the code to add 1 hour to the time of the last transmission, without a check if the time is greater than or equal to 23.00.
System error	An erroneous system state that can lead to system behavior that is unexpected by system users. The value of transmission time is set incorrectly (to 24.XX rather than 00.XX) when the faulty code is executed.
System failure	An event that occurs at some point in time when the system does not deliver a service as expected by its users. No weather data is transmitted because the time is invalid.

Faults and failures

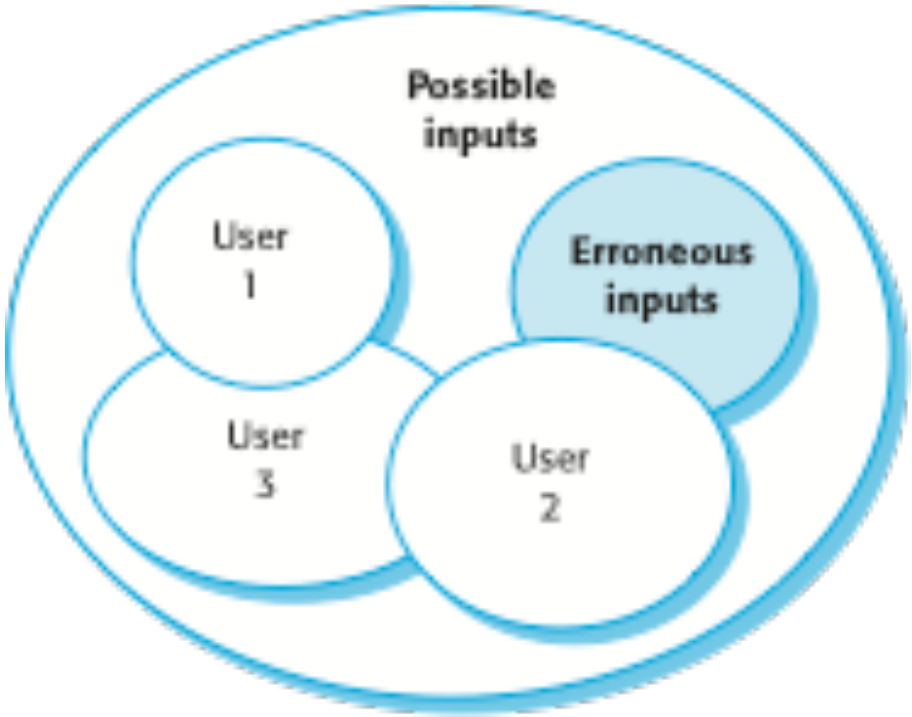


- Failures are a usually a result of system errors that are derived from faults in the system
- However, faults do not necessarily result in system errors
  - The erroneous system state resulting from the fault may be transient and 'corrected' before an error arises.
  - The faulty code may never be executed.
- Errors do not necessarily lead to system failures
  - The error can be corrected by built-in error detection and recovery
  - The failure can be protected against by built-in protection facilities. These may, for example, protect system resources from system errors

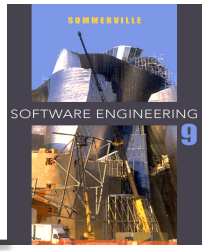
A system as an input/output mapping



Software usage patterns

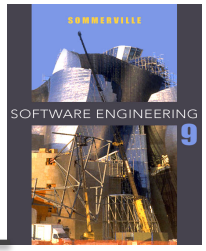


Reliability in use



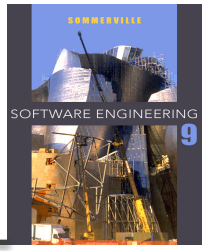
- Removing X% of the faults in a system will not necessarily improve the reliability by X%. A study at IBM showed that removing 60% of product defects resulted in a 3% improvement in reliability.
- Program defects may be in rarely executed sections of the code so may never be encountered by users. Removing these does not affect the perceived reliability.
- Users adapt their behaviour to avoid system features that may fail for them.
- A program with known faults may therefore still be perceived as reliable by its users.

Reliability achievement



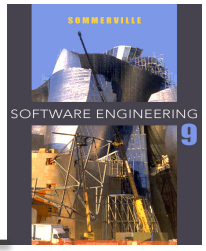
- Fault avoidance
  - Development technique are used that either minimise the possibility of mistakes or trap mistakes before they result in the introduction of system faults.
- Fault detection and removal
  - Verification and validation techniques that increase the probability of detecting and correcting errors before the system goes into service are used.
- Fault tolerance
  - Run-time techniques are used to ensure that system faults do not result in system errors and/or that system errors do not lead to system failures.

Safety



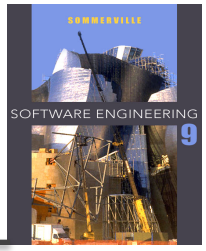
- Safety is a property of a system that reflects the system's ability to operate, normally or abnormally, without danger of causing human injury or death and without damage to the system's environment.
- It is important to consider software safety as most devices whose failure is critical now incorporate software-based control systems.
- Safety requirements are often exclusive requirements i.e. they exclude undesirable situations rather than specify required system services. These generate functional safety requirements.

Safety criticality



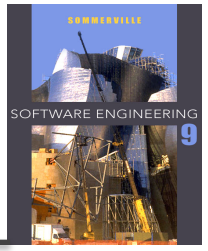
- Primary safety-critical systems
  - Embedded software systems whose failure can cause the associated hardware to fail and directly threaten people. Example is the insulin pump control system.
- Secondary safety-critical systems
  - Systems whose failure results in faults in other (socio-technical) systems, which can then have safety consequences. For example, the MHC-PMS is safety-critical as failure may lead to inappropriate treatment being prescribed.

Safety and reliability



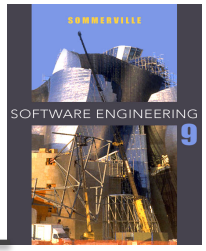
- Safety and reliability are related but distinct
  - In general, reliability and availability are necessary but not sufficient conditions for system safety
- Reliability is concerned with conformance to a given specification and delivery of service
- Safety is concerned with ensuring system cannot cause damage irrespective of whether or not it conforms to its specification

Unsafe reliable systems



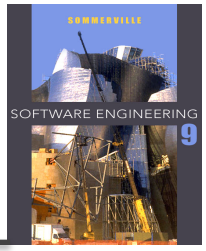
- There may be dormant faults in a system that are undetected for many years and only rarely arise.
- Specification errors
  - If the system specification is incorrect then the system can behave as specified but still cause an accident.
- Hardware failures generating spurious inputs
  - Hard to anticipate in the specification.
- Context-sensitive commands i.e. issuing the right command at the wrong time
  - Often the result of operator error.

Safety achievement



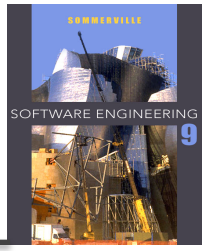
- Hazard avoidance
  - The system is designed so that some classes of hazard simply cannot arise.
- Hazard detection and removal
  - The system is designed so that hazards are detected and removed before they result in an accident.
- Damage limitation
  - The system includes protection features that minimise the damage that may result from an accident.

Normal accidents



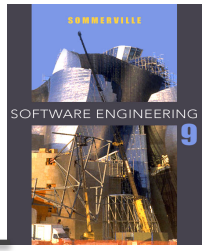
- Almost all accidents are a result of combinations of malfunctions rather than single failures.
- It is probably the case that anticipating all problem combinations, especially, in software controlled systems is impossible so achieving complete safety is impossible. Accidents are inevitable.

Software safety benefits



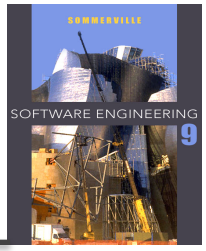
- Although software failures can be safety-critical, the use of software control systems contributes to increased system safety
  - Software monitoring and control allows a wider range of conditions to be monitored and controlled than is possible using electro-mechanical safety systems.
  - Software control allows safety strategies to be adopted that reduce the amount of time people spend in hazardous environments.
  - Software can detect and correct safety-critical operator errors.

Security



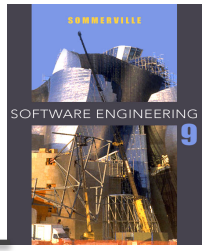
- The security of a system is a system property that reflects the system's ability to protect itself from accidental or deliberate external attack.
- Security is essential as most systems are networked so that external access to the system through the Internet is possible.
- Security is an essential pre-requisite for availability, reliability and safety.

Fundamental security



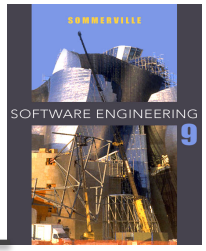
- If a system is a networked system and is insecure then statements about its reliability and its safety are unreliable.
- These statements depend on the executing system and the developed system being the same. However, intrusion can change the executing system and/or its data.
- Therefore, the reliability and safety assurance is no longer valid.

Security terminology



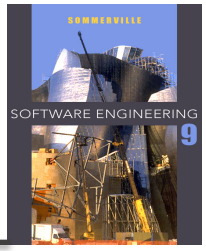
Term	Definition
Asset	Something of value which has to be protected. The asset may be the software system itself or data used by that system.
Exposure	Possible loss or harm to a computing system. This can be loss or damage to data, or can be a loss of time and effort if recovery is necessary after a security breach.
Vulnerability	A weakness in a computer-based system that may be exploited to cause loss or harm.
Attack	An exploitation of a system's vulnerability. Generally, this is from outside the system and is a deliberate attempt to cause some damage.
Threats	Circumstances that have potential to cause loss or harm. You can think of these as a system vulnerability that is subjected to an attack.
Control	A protective measure that reduces a system's vulnerability. Encryption is an example of a control that reduces a vulnerability of a weak access control system

Examples of security terminology (MHC-PMS)



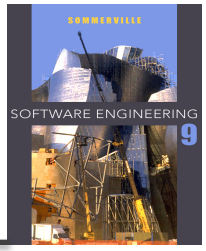
Term	Example
Asset	The records of each patient that is receiving or has received treatment.
Exposure	Potential financial loss from future patients who do not seek treatment because they do not trust the clinic to maintain their data. Financial loss from legal action by the sports star. Loss of reputation.
Vulnerability	A weak password system which makes it easy for users to set guessable passwords. User ids that are the same as names.
Attack	An impersonation of an authorized user.
Threat	An unauthorized user will gain access to the system by guessing the credentials (login name and password) of an authorized user.
Control	A password checking system that disallows user passwords that are proper names or words that are normally included in a dictionary.

Threat classes



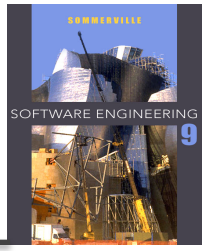
- Threats to the confidentiality of the system and its data
  - Can disclose information to people or programs that do not have authorization to access that information.
- Threats to the integrity of the system and its data
  - Can damage or corrupt the software or its data.
- Threats to the availability of the system and its data
  - Can restrict access to the system and data for authorized users.

Damage from insecurity



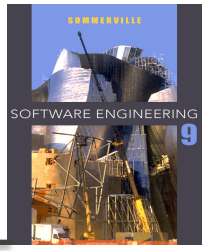
- Denial of service
  - The system is forced into a state where normal services are unavailable or where service provision is significantly degraded
- Corruption of programs or data
  - The programs or data in the system may be modified in an unauthorised way
- Disclosure of confidential information
  - Information that is managed by the system may be exposed to people who are not authorised to read or use that information

Security assurance



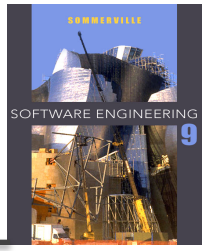
- Vulnerability avoidance
  - The system is designed so that vulnerabilities do not occur. For example, if there is no external network connection then external attack is impossible
- Attack detection and elimination
  - The system is designed so that attacks on vulnerabilities are detected and neutralised before they result in an exposure. For example, virus checkers find and remove viruses before they infect a system
- Exposure limitation and recovery
  - The system is designed so that the adverse consequences of a successful attack are minimised. For example, a backup policy allows damaged information to be restored

Key points



- The dependability in a system reflects the user's trust in that system.
- Dependability is a term used to describe a set of related 'non-functional' system attributes – availability, reliability, safety and security.
- The availability of a system is the probability that it will be available to deliver services when requested.
- The reliability of a system is the probability that system services will be delivered as specified.

Key points



- Reliability is related to the probability of an error occurring in operational use. A system with known faults may be reliable.
- Safety is a system attribute that reflects the system's ability to operate without threatening people or the environment.
- Security is a system attribute that reflects the system's ability to protect itself from external attack.
- Dependability is compromised if a system is insecure as the code or data may be corrupted.