UNIT I

INTRODUCTION

- Information security: a "well-informed sense of assurance that the information risks and controls are in balance." — Jim Anderson, Inovant (2002)
- Security professionals must review the origins of this field to understand its impact on our understanding of information security today

HISTORY OF INFORMATION SECURITY

- Computer security began immediately after the first mainframes were developed
  - Groups developing code-breaking computations during World War II created the first modern computers
  - Multiple levels of security were implemented
- Physical controls to limit access to sensitive military locations to authorized personnel
- The primary threats to security were physical theft of equipment, espionage against the products of the systems, and sabotage.

THE 1960'S

- Advanced Research Project Agency (ARPA) began to examine feasibility of redundant networked communications
- Larry Roberts developed ARPANET from its inception. ARPANET is the predecessor to the Internet.

THE 1970 AND 80'S

- ARPANET grew in popularity as did its potential for misuse
- Fundamental problems with ARPANET security were identified
  - No safety procedures for dial-up connections to ARPANET
  - Non-existent user identification and authorization to system
- Late 1970s: microprocessor expanded computing capabilities and security threats
- Information security began with Rand Report R-609 (paper that started the study of computer security)

- Scope of computer security grew from physical security to include:
  - Safety of data
  - Limiting unauthorized access to data
  - Involvement of personnel from multiple levels of an organization

MULTICS

Much of the early research on computer security is centered on a system called Multiplexed Information and Computing Service (MULTICS). It was the first operating system to integrate security into its core functions. It was a mainframe, time-sharing operating system developed in the mid- 1960s by a consortium of General Electric (GE), Bell Labs, and the Massachusetts Institute of Technology (MIT). In mid-1969 several developers (Ken Thompson, Dennis Ritchie, Rudd Canaday, and Doug McIlro) created a new operating system called UNIX. While the MULTICS system implemented multiple security levels and passwords, the UNIX system did not. Its primary function was text processing. The password function was introduced in UNIX in 1970's In the late 1970s, the microprocessor brought the personal computer and a new age of computing.

THE 1990'S

- Networks of computers became more common; so too did the need to interconnect networks
- Internet became first manifestation of a global network of networks
- Initially based on de facto standards
- In early Internet deployments, security was treated as a low priority

2000 TO PRESENT

- The Internet brings millions of computer networks into communication with each other—many of them unsecured
- Ability to secure a computer's data influenced by the security of every computer to which it is connected
- Growing threat of cyber-attacks has increased the need for improved security

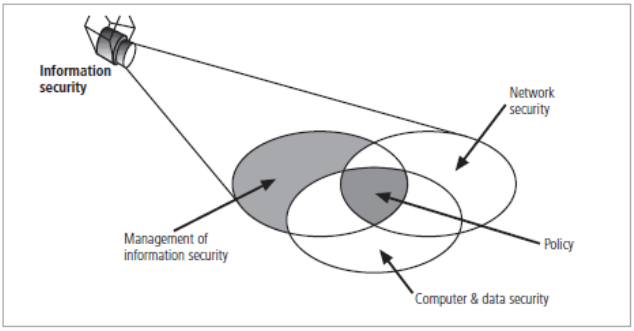
SECURITY

"The quality or state of being secure—to be free from danger"

A successful organization should have multiple layers of security in place:

- **Physical security**, to protect physical items, objects, or areas from unauthorized access and misuse
- **Personnel security**, to protect the individual or group of individuals who are authorized to access the organization and its operations
- **Operations security**, to protect the details of a particular operation or series of activities
- **Communications security**, to protect communications media, technology, and content
- **Network security**, to protect networking components, connections, and contents
- **Information security**, to protect the confidentiality, integrity and availability of information assets, whether in storage, processing, or transmission. It is achieved

The **Committee on National Security Systems** (CNSS) defines information security as the protection of information and its critical elements, including the systems and hardware that use, store, and transmit that information. The CNSS model of information security evolved from a concept developed by the computer security industry called the C.I.A. triangle.



C.I.A Triangle is based on the three characteristics of information: confidentiality, integrity, and availability. The threats to the confidentiality, integrity, and availability of information have evolved into a vast collection of events, including accidental or intentional damage, destruction, theft, unintended or unauthorized modification, or other misuse from human or nonhuman threats. This new environment of many constantly evolving threats has

prompted the development of a more robust model that addresses the complexities of the current information security environment. The expanded model consists of a list of critical characteristics of information.

Critical Characteristics of Information

The value of information comes from the characteristics it possesses. Each critical characteristic of information—that is, the expanded C.I.A. triangle—is defined below

Availability: Availability enables authorized users—persons or computer systems—to access information without interference or obstruction and to receive it in the required format.

Accuracy: Information has accuracy when it is free from mistakes or errors and it has the value that the end user expects. If information has been intentionally or unintentionally modified, it is no longer accurate.

Authenticity: Authenticity of information is the quality or state of being genuine or original, rather than a reproduction or fabrication. Information is authentic when it is in the same state in which it was created, placed, stored, or transferred. **E-mail spoofing,** the act of sending an e-mail message with a modified field, is a problem for many people today, because often the modified field is the address of the originator. Spoofing the sender's address can fool e-mail recipients into thinking that messages are legitimate traffic. Spoofing can also alter data being transmitted across a net- work, as in the case of user data protocol (UDP) packet spoofing, which can enable the attacker to get access to data stored on computing systems. Another variation on spoofing is **phishing,** when an attacker attempts to obtain personal or financial information using fraudulent means, most often by posing as another individual or organization. Pretending to be someone you are not is sometimes called pretexting. When used in a phishing attack, e-mail spoofing lures victims to a Web server that does not represent the organization it purports to, in an attempt to steal their private data such as account numbers and pass- words.

Confidentiality: Information has confidentiality when it is protected from disclosure or exposure to unauthorized individuals or systems. Confidentiality ensures that only those with the rights and privileges to access information are able to do so. When unauthorized individuals or systems can view information, confidentiality is breached. To protect the confidentiality of

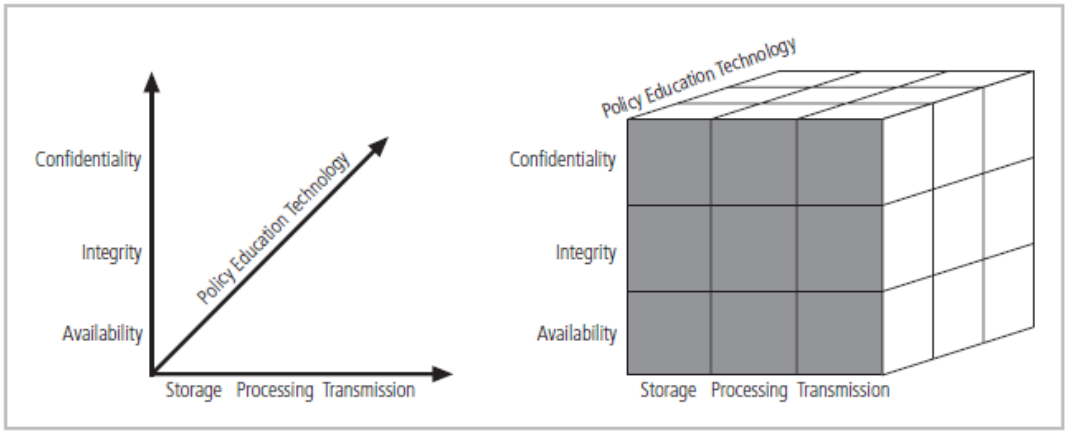
information, you can use a number of measures, including the following: Information classification secure document storage Application of general security policies Education of information custodians and end users.

Integrity: Information has integrity when it is whole, complete, and uncorrupted. Corruption can occur while information is being stored or transmitted. Many computer viruses and worms are designed with the explicit purpose of corrupting data. For this reason, a key method for detecting a virus or worm is to look for changes in file integrity as shown by the size of the file. Another key method of assuring information integrity is file hashing, in which a file is read by a special algorithm that uses the value of the bits in the file to compute a single large number called a hash value. The hash value for any combination of bits is unique. If a computer system performs the same hashing algorithm on a file and obtains a different number than the recorded hash value fort that file, the file has been compromised and the integrity of the information is lost. Noise in the transmission media can also cause data to lose its integrity.

Utility: The utility of information is the quality or state of having value for some purpose or end. Information has value when it can serve a purpose. If information is available, but is not in a format meaningful to the end user, it is not useful.

Possession: The possession of information is the quality or state of ownership or control. Information is said to be in one's possession if one obtains it, independent of format or other characteristics. While a breach of confidentiality always results in a breach of posses- sion, a breach of possession does not always result in a breach of confidentiality.

NSTISSC / CNSS SECURITY MODEL



National Training Standard for Information Systems Security Professionals (NSTISSC) was renamed to Committee on National Security Systems (CNSS). NSTISSC/CNSS security model is a comprehensive information security model and has become a widely accepted evaluation standard for the security of information systems. The model is created by John McCumber in 1991. It provides a graphical representation of the architectural approach widely used in computer and information security and it is now known as the McCumber Cube.

The McCumber Cube in the figure shows three dimensions. If extrapolated, the three dimensions of each axis become a 3 x 3 x 3 cube with 27 cells representing areas that must be addressed to secure today's information systems. To ensure system security, each of the 27 areas must be properly addressed during the security process. For example, the intersection between technology, integrity, and storage requires a control or safeguard that addresses the need to use technology to protect the integrity of information while in storage. One such control might be a system for detecting host intrusion that protects the integrity of information by alerting the security administrators to the potential modification of a critical file.

COMPONENTS OF AN INFORMATION SYSTEM

Information system (IS) is entire set of components necessary to use information as a resource in the organization

Software:

The software component of the IS comprises applications, operating systems, and assorted command utilities. The exploitation of errors in software programming accounts for a substantial portion of the attacks on information. Software carries the lifeblood of information through an organization. Unfortunately, software programs are often created under the constraints of project management, which limit time, cost, and manpower. Information security is all too often implemented as an afterthought, rather than developed as an integral component from the beginning. In this way, software programs become an easy target of accidental or intentional attacks

Hardware:

Hardware is the physical technology that houses and executes the software, stores and transports the data, and provides interfaces for the entry and removal of information from the

system. Physical security policies deal with hardware as a physical asset and with the protection of physical assets from harm or theft. Securing the physical location of computers and the computers themselves is important because a breach of physical security can result in a loss of information.

Data:

Data stored, processed, and transmitted by a computer system must be protected. Data is often the most valuable asset possessed by an organization and it is the main target of intentional attacks.

People:

People have always been a threat to information security. People can be the weakest link in an organization's information security program. And unless policy, education and training, awareness, and technology are properly employed to prevent people from accidentally or intentionally damaging or losing information, they will remain the weakest link. Social engineering is common place of human error. It can be used to manipulate the actions of people to obtain access information about a system.

Procedures:

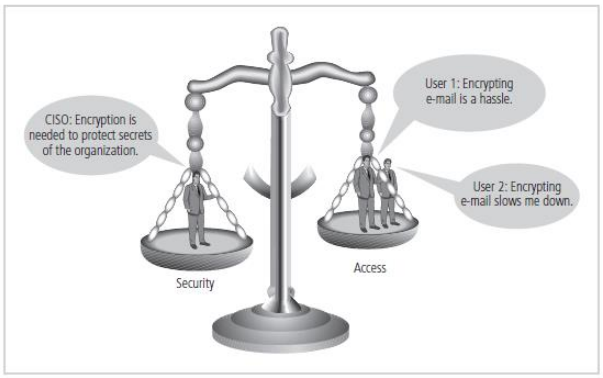
Procedures are written instructions for accomplishing a specific task. When an unauthorized user obtains an organization's procedures, this poses a threat to the integrity of the information. Most organizations distribute procedures to their legitimate employees so they can access the information system, but many of these companies often fail to provide proper education on the protection of the procedures. Educating employees about safeguarding procedures is as important as physically securing the information system.

Networks:

When information systems are connected to each other to form local area networks (LANs), and these LANs are connected to other networks such as the Internet, new security challenges rapidly emerge. The physical technology that enables network functions is becoming more and more accessible to organizations of every size. Applying the traditional tools of physical security, such as locks and keys, to restrict access to and interaction with the

hardware components of an information system are still important; but when computer systems are networked, this approach is no longer enough. Steps to provide network security are essential.

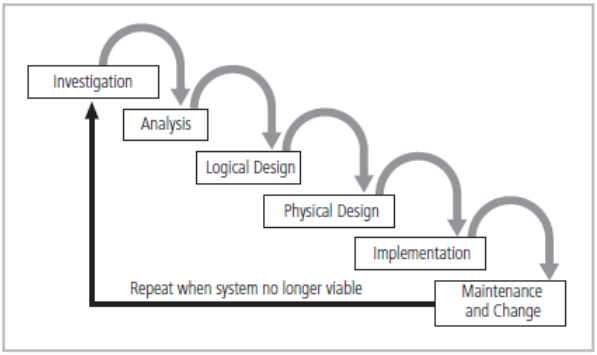
Balancing Information Security and Access



- Impossible to obtain perfect security—it is a process, not an absolute
- Security should be considered balance between protection and availability
- To achieve balance, level of security must allow reasonable access, yet protect against threats

The Systems Development Life Cycle

SDLC is a methodology for design and implementation of information system within an organization.



Methodology and Phases

A methodology is a formal approach to solving a problem by means of a structured sequence of procedures. Once a methodology has been adopted, the key milestones are

established and a team of individuals is selected and made accountable for accomplishing the project goals. The traditional SDLC consists of six general phases. The waterfall model pictured in Figure illustrates that each phase begins with the results and information gained from the previous phase. At the end of each phase comes a structured review or reality check, during which the team determines if the project should be continued, discontinued, outsourced, postponed, or returned to an earlier phase depending on whether the project is proceeding as expected and on the need for additional expertise, organizational knowledge, or other resources.

1. Investigation
    - What problem is the system being developed to solve?
    - Objectives, constraints, and scope of project are specified
    - Preliminary cost-benefit analysis is developed
    - At the end, feasibility analysis is performed to assess economic, technical, and behavioral feasibilities of the process
2. Analysis
    - Consists of assessments of:
        - The organization
        - Current systems
        - Capability to support proposed systems
    - Analysts determine what new system is expected to do and how it will interact with existing systems
    - Ends with documentation of findings and update of feasibility analysis
3. Logical Design
    - Main factor is business need
        - Applications capable of providing needed services are selected
    - Data support and structures capable of providing the needed inputs are identified
    - Technologies to implement physical solution are determined
    - Feasibility analysis performed at the end
4. Physical Design
    - Technologies to support the alternatives identified and evaluated in the logical design are selected
    - Components evaluated on make-or-buy decision

- Feasibility analysis performed
    - Entire solution presented to end-user representatives for approval

5. Implementation
    - Needed software created
    - Components ordered, received, and tested
    - Users trained and documentation created
    - Feasibility analysis prepared
        - Users presented with system for performance review and acceptance test
6. Maintenance and Change
    - Longest and most expensive phase
    - Consists of tasks necessary to support and modify system for remainder of its useful life
    - Life cycle continues until the process begins again from the investigation phase
    - When current system can no longer support the organization's mission, a new project is implemented

The Security Systems Development Life Cycle

- The same phases used in traditional SDLC may be adapted to support specialized implementation of an IS project
- Identification of specific threats and creating controls to counter them
- SecSDLC is a coherent program rather than a series of random, seemingly unconnected actions

1. Investigation
    - Identifies process, outcomes, goals, and constraints of the project
    - Begins with Enterprise Information Security Policy (EISP)
    - Organizational feasibility analysis is performed
2. Analysis
    - Documents from investigation phase are studied
    - Analysis of existing security policies or programs, along with documented current threats and associated controls
    - Includes analysis of relevant legal issues that could impact design of the security solution

- Risk management task begins

3. Logical Design
   - Creates and develops blueprints for information security
   - Incident response actions planned:
     - Continuity planning
     - Incident response
     - Disaster recovery
   - Feasibility analysis to determine whether project should be continued or outsourced
4. Physical Design
   - Needed security technology is evaluated, alternatives are generated, and final design is selected
   - At end of phase, feasibility study determines readiness of organization for project
5. Implementation
   - Needed security technology is evaluated, alternatives are generated, and final design is selected
   - At end of phase, feasibility study determines readiness of organization for project
6. Maintenance and Change
   - Needed security technology is evaluated, alternatives are generated, and final design is selected
   - At end of phase, feasibility study determines readiness of organization for project

Business Needs First

Information security performs four important functions for an organization:

1. Protecting the organization's ability to function
2. Enabling the safe operation of applications running on the organization's IT systems
3. Protecting the data the organization collects and uses
4. Safeguarding the organization's technology assets

Protecting the Functionality of an Organization

Both general management and IT management are responsible for implementing information security that protects the organization's ability to function. Each of an organization's

communities of interest must address information security in terms of business impact and the cost of business interruption, rather than isolating security as a technical problem.

Enabling the Safe Operation of Applications

A modern organization needs to create an environment that safeguards applications, particularly those that are important elements of the organization's infrastructure—operating system platforms, electronic mail (e-mail), and instant messaging (IM) applications. Organizations acquire these elements from a service provider or they build their own. Once an organization's infrastructure is in place, management must continue to oversee it, and not relegate its management to the IT department.

Protecting Data that Organizations Collect and Use

Without data, an organization loses its record of transactions and/or its ability to deliver value to its customers. Protecting data in motion and data at rest are both critical aspects of information security. The value of data motivates attackers to steal, sabotage, or corrupt it. An effective information security program implemented by management protects the integrity and value of the organization's data.

Safeguarding Technology Assets in Organizations

Organizations must have secure infrastructure services based on size and scope of enterprise. Additional security services may be needed as organization grows. More robust solutions may be needed to replace security programs the organization has outgrown

Threats

Threat: an object, person, or other entity that represents a constant danger to an asset

- Management must be informed of the different threats facing the organization
- Overall security is improving
- The 2009 CSI/FBI survey found
  - 64 percent of organizations had malware infections
  - 14 percent indicated system penetration by an outsider

1. **Compromises to Intellectual Property**

Intellectual property is defined as "the ownership of ideas and control over the tangible or virtual representation of those ideas. Intellectual property can be trade secrets, copyrights, trademarks, and patents. The unauthorized appropriation of IP constitutes a threat to information security. The most common IP breach is software piracy. Many individuals and organizations do not purchase software as mandated by the owner's license agreements. Software licenses are strictly enforced by a number of regulatory and private organizations, and software publishers use several control mechanisms to prevent copyright infringement. In addition to the laws against software piracy, two watchdog organizations investigate allegations of software abuse: the Software & Information Industry Association (SIIA) at www.siia.net, formerly known as the Software Publishers Association, and the Business Software Alliance (BSA) at www.bsa.org. A number of technical mechanisms—digital watermarks and embedded code, copyright codes, and even the intentional placement of bad sectors on software media have been used to enforce copyright laws. The most common tool, a license agreement window that usually pops up during the installation of new software, establishes that the user has read and agrees to the license agreement. Another effort to combat piracy is the online registration process.

2. **Deliberate Software Attacks**

Deliberate software attacks is designing and deploying software to attack a system. This software is referred to as malicious code or malicious software, or malware. These software components or programs are designed to damage, destroy, or deny service to the target systems.

a. Virus

A computer virus consists of segments of code that perform malicious actions. The code attaches itself to an existing program and takes control of that program's access to the targeted computer. The virus controlled program then replicates itself into additional targeted systems. One of the most common methods of virus transmission is via e-mail attachment files.

b. Worms

A worm is a malicious program that replicates itself constantly, without requiring another program environment. Worms can continue replicating themselves until they completely fill available resources, such as memory, hard drive space, and network bandwidth. Once the worm has infected a computer, it can redistribute itself to all e-mail addresses found on the infected system.

Furthermore, a worm can deposit copies of itself onto all Web servers that it connects.

c. Trojan horses

Trojan horses are software programs that hide their true nature and reveal their designed behaviour only when activated. Trojan horses are frequently disguised as helpful, interesting, or necessary pieces of software, such as readme.exe files often included with shareware or freeware packages

d. Back door or trap door

A virus or worm can have a payload that installs a back door or trap door component in a system, which allows the attacker to access the system at will with special privileges.

e. Polymorphic threats

A polymorphic threat is one that over time changes the way it appears to antivirus software programs, making it undetectable by techniques that look for preconfigured signatures. These viruses and worms actually evolve, changing their size and other external file characteristics to elude detection by antivirus software programs.

f. Virus and worm hoaxes

People can disrupt the harmony and flow of an organization when they send group e-mails warning of supposedly dangerous viruses that don't exist. When people fail to follow virus-reporting procedures, the network becomes overloaded, and much time and energy is wasted as users forward the warning message to everyone they know, post the message on bulletin boards, and try to update their antivirus protection software.

3. Deviation in quality of service

An organization's information system depends on the successful operation of many inter-dependent support systems. Deviations in quality of service can result from incidents such as a backhoe taking out a fiber-optic link for an ISP, disruption. Irregularities in Internet service, communications, and power supplies can dramatically affect the availability of information and systems.

a. Internet service issues

- Internet service provider (ISP) failures can considerably undermine availability of information

- Outsourced Web hosting provider assumes responsibility for all Internet services as well as hardware and Web site operating system software

b. Communications and other service provider issues
    - Other utility services affect organizations: telephone, water, wastewater, trash pickup, etc.
    - Loss of these services can affect organization's ability to function

c. Power irregularities
    - Organizations with inadequately conditioned power are susceptible
    - Controls can be applied to manage power quality
    - Fluctuations (short or prolonged)
    - Excesses (spikes or surges) – voltage increase
    - Shortages (sags or brownouts) – low voltage
    - Losses (faults or blackouts) – loss of power

4. Espionage or Trespass

When an unauthorized individual gains access to the information an organization is trying to protect, that act is categorized as espionage or trespass. Attackers can use many different methods to access the information stored in an information system. Some information gathering techniques are quite legal, for example, using a Web browser to perform market research. These legal techniques are called, collectively, **competitive intelligence.** When information gatherers employ techniques that cross the threshold of what is legal or ethical, they are conducting **industrial espionage.** One simple form of Espionage is **shoulder surfing**. This technique is used in public or semi-public settings when individuals gather information they are not authorized to have by looking over another individual's shoulder or viewing the information from a distance. Instances of shoulder surfing occur at computer terminals, desks, ATM machines, on the bus or subway where people use smartphones and tablet PCs, or other places where a person is accessing confidential information.

Acts of trespass can lead to unauthorized real or virtual actions that enable information gatherers to enter premises or systems they have not been authorized to enter. The classic perpetrator of espionage or trespass is the hacker. Hackers are "people who use and create computer software to gain access to information illegally. There are generally two skill levels among hackers.

- Expert hacker

- Develops software scripts and program exploits
- Usually a master of many skills
- Will often create attack software and share with others

• Unskilled hacker

- Many more unskilled hackers than expert hackers
- Use expertly written software to exploit a system
- Do not usually fully understand the systems they hack

Other terms for system rule breakers:

- Cracker: "cracks" or removes software protection designed to prevent unauthorized duplication
- Phreaker: hacks the public telephone network

5. **Forces of Nature**

   Forces of nature or acts of God can present some of the most dangerous threats, because they usually occur with very little warning and are beyond the control of people. These threats, which include events such as fires, floods, earthquakes, and lightning as well as volcanic eruptions and insect infestations, can disrupt not only the lives of individuals but also the storage, transmission, and use of information. Organizations must implement controls to limit damage and prepare contingency plans for continued operations.

6. **Human Error or Failure**

   This category includes acts performed without intent or malicious purpose by an authorized user. When people use information systems, mistakes happen. Inexperience, improper training, and the incorrect assumptions are few causes. Regardless of the cause, even innocuous mistakes can produce extensive damage.

   Employee mistakes can easily lead to:

   - Revelation of classified data
   - Entry of erroneous data
   - Accidental data deletion or modification
   - Data storage in unprotected areas
   - Failure to protect information

7. **Information Extortion**

Information extortion occurs when an attacker or trusted insider steals information from a computer system and demands compensation for its return or for an agreement not to disclose it. Extortion is common in credit card number theft.

8. **Missing, Inadequate, or Incomplete Organizational Policy or Planning**

   Missing, inadequate, or incomplete organizational policy or planning makes an organization vulnerable to loss, damage, or disclosure of information assets when other threats lead to attacks. Information security is, at its core, a management function.

9. **Sabotage or Vandalism**

   This category of threat involves the deliberate sabotage of a computer system or business, or acts of vandalism to either destroy an asset or damage the image of an organization. These acts can range from petty vandalism by employees to organized sabotage against an organization. Vandalism to a Web site can erode consumer confidence, thus diminishing an organization's sales and net worth, as well as its reputation.

10. **Theft**

    The threat of theft—the illegal taking of another's property, which can be physical, electronic, or intellectual—is a constant. The value of information is diminished when it is copied without the owner's knowledge. Physical theft can be controlled quite easily. Electronic theft, however, is a more complex problem to manage and control.

11. **Technical Hardware Failures or Errors**

    Technical hardware failures or errors occur when a manufacturer distributes equipment containing a known or unknown flaw. These defects can cause the system to perform out- side of expected parameters, resulting in unreliable service or lack of availability.

12. **Technical Software Failures or Errors**

    Large quantities of computer code are written, debugged, published, and sold before all their bugs are detected and resolved. Sometimes, combinations of certain software and hardware reveal new bugs. Software bugs are so commonplace that entire Web sites are dedicated to documenting them.

13. **Technological Obsolescence**

    Antiquated or outdated infrastructure can lead to unreliable and untrustworthy systems. Management's strategic planning should always include an analysis of the technology currently in use. Ideally, proper planning by management should prevent technology from becoming obsolete, but when obsolescence is manifest, management must take

immediate action. IT professionals play a large role in the identification of probable obsolescence.

Attacks

An attack is an act that takes advantage of a vulnerability to compromise a controlled system. It is accomplished by a threat agent that damages or steals an organization's information or physical asset. A vulnerability is an identified weakness in a controlled system, where controls are not present or are no longer effective. The fallowing are various types of attacks:

1. Malicious code: includes execution of viruses, worms, Trojan horses, and active Web scripts with intent to destroy or steal information
2. Hoaxes: transmission of a virus hoax with a real virus attached; more devious form of attack
3. Back door: gaining access to system or network using known or previously unknown/newly discovered access mechanism
4. Password crack: attempting to reverse calculate a password
5. Brute force: trying every possible combination of options of a password
6. Dictionary: selects specific accounts to attack and uses commonly used passwords (i.e., the dictionary) to guide guesses
7. Denial-of-service (DoS): attacker sends large number of connection or information requests to a target
    a. Target system cannot handle successfully along with other, legitimate service requests
    b. May result in system crash or inability to perform ordinary functions
8. Distributed denial-of-service (DDoS): coordinated stream of requests is launched against target from many locations simultaneously
9. Spoofing: technique used to gain unauthorized access; intruder assumes a trusted IP address
10. Man-in-the-middle: attacker monitors network packets, modifies them, and inserts them back into network
11. Spam: unsolicited commercial e-mail; more a nuisance than an attack, though is emerging as a vector for some attacks

12. Mail bombing: also a DoS; attacker routes large quantities of e-mail to target
13. Sniffers: program or device that monitors data traveling over network; can be used both for legitimate purposes and for stealing information from a network
14. Phishing: an attempt to gain personal/financial information from individual, usually by posing as legitimate entity
15. Pharming: redirection of legitimate Web traffic (e.g., browser requests) to illegitimate site for the purpose of obtaining private information
16. Social engineering: using social skills to convince people to reveal access credentials or other valuable information to attacker

    "People are the weakest link. You can have the best technology; firewalls, intrusion-detection systems, biometric devices and somebody can call an unsuspecting employee. That's all she wrote, baby. They got everything." — Kevin Mitnick
17. Timing attack: relatively new; works by exploring contents of a Web browser's cache to create malicious cookie

Secure Software Development

Systems consist of hardware, software, networks, data, procedures, and people using the system. Secure systems require secure, or at least securable, software. The development of systems and the software they use is often accomplished using a methodology known as the systems development life cycle (SDLC). Many organizations recognize the need to include planning for security objectives in the SDLC they use to create systems, and have put in place procedures to create software that is more able to be deployed in a secure fashion. This approach to software development is known as software assurance, or SA

Software Assurance and the SA Common Body of Knowledge

- National effort underway to create common body of knowledge focused on secure software development
- US Department of Defense and Department of Homeland Security supported Software Assurance Initiative, which resulted in publication of Secure Software Assurance (SwA) Common Body of Knowledge (CBK)
- SwA CBK serves as a strongly recommended guide to developing more secure applications

The following stages should be incorporated into the software SDLC.

1. **Software Design Principles**

Good software development should result in a finished product that meets all of its design specifications. Information security considerations are a critical component of those specifications, though that has not always been true.

Some commonplace security principles:

- Keep design simple and small
- Access decisions by permission not exclusion
- Every access to every object checked for authority
- Design depends on possession of keys/passwords
- Protection mechanisms require two keys to unlock
- Programs/users utilize only necessary privileges
- Some commonplace security principles:
- Keep design simple and small
- Access decisions by permission not exclusion
- Every access to every object checked for authority
- Design depends on possession of keys/passwords
- Protection mechanisms require two keys to unlock
- Programs/users utilize only necessary privileges

2. **Software Development Security problems**

- Buffer overruns

  Buffers are used to manage mismatches in the processing rates between two entities involved in a communication process. A buffer overrun (or buffer overflow) is an application error that occurs when more data is sent to a program buffer than it is designed to handle. During a buffer overrun, an attacker can make the target sys- tem execute instructions, or the attacker can take advantage of some other unintended consequence of the failure.
- Command injection

Command injection problems occur when user input is passed directly to a compiler or interpreter. The underlying issue is the developer's failure to ensure that command input is validated before it is used in the program.

- Cross-site scripting

  Cross site scripting (or XSS) occurs when an application running on a Web server gathers data from a user in order to steal it. An attacker can use weak- nesses in the Web server environment to insert commands into a user's browser session. This allows the attacker to acquire valuable information, such as account credentials, account numbers, or other critical data

- Failure to handle errors

  Failure to handle errors can cause a variety of unexpected system behaviors. Programmers are expected to anticipate problems and prepare their application code to handle them.

- Failure to protect network traffic

  With the growing popularity of wireless net- working comes a corresponding increase in the risk that wirelessly transmitted data will be intercepted. Without appropriate encryption (such as that afforded by WPA), attackers can intercept and view your data.

- Failure to store and protect data securely

  Programmers are responsible for integrating access controls into, and keeping secret information out of, programs. Failure to properly implement sufficiently strong access controls makes the data vulnerable.

- Failure to use cryptographically strong random numbers

  Most cryptographic procedures require strong random number generators to ensure higher security. Weak random number could lead to vulnerability in the system.

- Format string problems

  Programmers may use data from untrusted sources as a format string. An attacker may embed characters that are meaningful as formatting directives into malicious input so that the attacker may be able to access information or overwrite much targeted portions of the program's stack with attacker's data.

- Neglecting change control

  Once the system is in production, change control processes ensure that only authorized changes are introduced and that all changes are adequately tested before being released.

- Improper file access

If an attacker changes the expected location of a file by intercepting and modifying a program code call, the attacker can force a program to use files other than the ones the program is supposed to use.

- Improper use of SSL

  Failure to use Hypertext Transfer Protocol Secure (HTTPS), to validate the certificate authority and then validate the certificate itself, or to validate the information against a certificate revocation list (CRL), can compromise the security of SSL traffic.

- Information leakage

  One of the most common methods of obtaining inside and classified information is directly or indirectly from an individual. By warning employees against disclosing information, organizations can protect the secrecy of their operation.

- Integer bugs (overflows/underflows)

  Integer bugs fall into four broad classes: overflows, underflows, truncations, and signed errors. Integer bugs are usually exploited indirectly—that is, triggering an integer bug enables an attacker to corrupt other areas of memory, gaining control of an application

- Race conditions

  A race condition is a failure of a program that occurs when an un-expected ordering of events in the execution of the program results in a conflict over access to the same system resource.

- SQL injection

  SQL injection occurs when developers fail to properly validate user input before using it to query a relational database.

- Trusting network address resolution

  Attackers most commonly compromise segments of the DNS by either attacking the name of the name server and substituting their own DNS primary name server, by incorrectly updating an individual record, or by responding before an actual DNS can. if the attacker discovers a delay in a name server (or can introduce one, as in a denial of service attack) they can set up another server to respond as if it were the actual DNS server, before the real DNS server can. The client accepts the first set of information it receives and is directed to that IP address.

- Unauthenticated key exchange

  One of the biggest challenges in private key systems, which involve two users sharing the same key, is securely getting the key to the other party.

- Use of magic URLs and hidden forms

  Sensitive state information is simply included in a “magic” URL (for example, the authentication ID is passed as a parameter in the URL for the exchanges that will follow) or included in hidden form fields on the HTML page. If this information is stored as plain text, an attacker can harvest the information from a magic URL as it travels across the network, or use scripts on the client to modify information in hidden form fields.
- Use of weak password-based systems

  Failure to require sufficient password strength, and to control incorrect password entry, is a serious security issue. Password policy can specify the number and type of characters, the frequency of mandatory changes, and even the reusability of old passwords
- Poor usability

  Employees prefer doing things the easy way. When faced with an “official way” of performing a task and an “unofficial way”—which is easier—they prefer the easier method. The later may result in security problems and may lead to loss of data.