**Security of Information Systems (computers and networks)**

Security Threats

To understand the types of threats to security that exist, we need to have a definition of security requirements. Computer and network security addresses four requirements:

1. Confidentiality (secrecy): Requires that the information in a computer system only be accessible for reading by authorized parties. This type of access includes printing, displaying, and other forms of disclosure, including simply revealing the existence of an object.
2. Integrity (consistency): Requires that computer system assets can be modified only by authorized parties. Modification includes writing, changing, changing status, deleting, and creating. Before modification and after it, system should be in the consistent state, specified by some constraints.
3. Availability: Requires that computer system assets are available to authorized parties.
4. Authenticity: Requires that a computer system be able to verify the identity of a user.

Types of Threats

**Threat** is a potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm. That is, a threat is a possible danger that might exploit a **vulnerability**.



[6]

*[6] is a reference from the list of references in CMSE512 outline*

**Attack** is an assault on system security that derives from an intelligent threat: that is, an intelligent act that is a deliberate attempt to evade security services and violate the security policy of a system

The types of attacks on the security of a computer system are best characterized by viewing the function of the computer system as providing information. The normal flow of information may be affected by

1. Interruption: An asset is destroyed or becomes unavailable or unusable. This is an attack on **availability**. Examples include destruction of a piece of hardware, such as a hard disk, the cutting of a communication line, or the disabling of the file management system.
2. Interception: An unauthorized party gains access to an asset. This is an attack on **confidentiality**. The unauthorized party could be a person, a program, or a computer. Examples include wiretapping to capture data in a network and the illicit copying of files or programs.
3. Modification: An unauthorized party not only gains access to but tampers with an asset. This is an attack on **integrity, confidentiality, and authenticity**. Examples include changing values in a data file, altering program so that it performs differently, and modifying the content of messages being transmitted in a network.
4. Fabrication: An unauthorized party inserts counterfeit objects into the system. This is an attack on **integrity and authenticity**. Examples include the insertion of spurious messages in a network or the additions of records to a file.

Computer System Assets

|  |  |  |  |
| --- | --- | --- | --- |
|  | Availability | Secrecy | Integrity/Authenticity |
| Hardware | Equipment is stolen or disabled, thus denying service |  |  |
| Software | Programs are deleted, denying access to users | An unauthorized copy of software is made | A working program is modified, either to cause it to fail during execution or to cause it to do some unintended task |
| Data | Files are deleted, denying access to users | An unauthorized read of data is performed. An analysis of statistical data reveals underlying data. | Existing files are modified or new files are fabricated. |
| Communication lines | Messages are destroyed or deleted. Communication lines or networks are rendered unavailable. | Messages are read. The traffic pattern of messages is observed. | Messages are modified, delayed, reordered, or duplicated. False messages are fabricated. |

Hardware

The main threat to computer system hardware is in the area of availability. Threats include accidental and deliberate damage to equipment as well as theft. Physical and administrative security measures are needed to deal with these threats.

Software

The operating system, utilities, and application programs are what make computer system hardware useful to business and individuals.

A key threat to software is an attack on availability. Software is easy to delete. Careful software configuration management, which includes making backups of the most recent version of software, can maintain high availability. A more difficult problem to deal with is software modification that results in a program that still functions but that behaves differently than before. Computer viruses and related attacks fall into this category and are treated later. And software secrecy – this problem of unauthorized copying of software is solved using special measures (e.g., use of a key disk, registration numbers, etc.) but completely it is not solved.

Data

Data security is more widespread than hardware or software security problems, which involves files and other forms of data controlled by individuals, groups, and business organizations. Security concerns with respect to data are broad, encompassing availability, secrecy, and integrity. The obvious concern with secrecy, of course, is the unauthorized reading of data files or databases, and this area has been the subject of perhaps more research and effort than any other area of computer security. A less obvious secrecy threat involves the analysis of data and manifests itself in the use of so-called statistical databases, which provide summary or aggregate information. They may be used to disclose personal information. For example, if one table records the aggregate of the incomes of A, B, C, and D, and another one records the aggregate of the incomes of A, B, C, D, and E, the difference between the two aggregates would be the income of E. This problem is exacerbated by the increasing desire to combine data sets. Finally, data integrity is a major concern in most installations. Modifications to data files can have consequences from minor to disastrous.

Communication Lines and Networks

There are two classes of attacks: passive and active.

Passive attacks are: release of message contents and traffic analysis.

The release of message contents is easily understood.

The traffic analysis is more subtle. Suppose that we had a way of masking the contents of messages or other information traffic so that opponents, even if they captured the massage, could not extract the information from the message. The common technique for masking is encryption. If we had encryption protection in place, an opponent might still be able to observe the pattern of these messages. The opponent could determine the location and identity of communicating hosts and could observe the frequency and length of messages being exchanged. This information might be useful in guessing the nature of the communication that was taking place.

Passive attacks are very difficult to detect because they do not involve any alteration of the data. However, it is feasible to prevent the success of these attacks. Thus, the emphasis in dealing with passive attacks is on prevention rather than detection.

Active attacks involve some modification of the data stream or the creation of the false stream and can be subdivided into four categories: masquerade, replay, modification of messages, and denial of service.

A masquerade takes place when one entity pretends to be a different entity. A masquerade attack usually includes one of the other forms of active attack. For example, authentication sequences can be captured and replayed after a valid authentication sequence has taken place, thus enabling an authorized entity with few privileges to obtain extra privileges by impersonating an entity that has those privileges.

Replay involves the passive capture of a data unit and its subsequent retransmission to produce unauthorized effect.

Modification of messages simply means that some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect. For

Communication Lines and Networks (Cont)

example, “Allow John Smith to read confidential file accounts” is modified to “Allow Fred Brown to read confidential file accounts”.

The denial of service prevents or inhibits the normal use or management of communication facilities. This attack may have a specific target; for example, an entity may suppress all messages directed to a particular destination (e.g., the security audit service). Another form of a service denial is the disruption of an entire network, either by disabling the network or by overloading it with messages so as to degrade performance. Active attack is difficult to prevent, that’s why the main goal is to detect attack and to recover from any disruption caused by it.



[6]

In addition to CIA concepts, there are also AAA concepts (assurance, authenticity, anonymity)

Assurance: how trust is provided and managed in computer systems. Trust: degree to which we have confidence that people or systems are behaving in the way we expect

Trust involves the interplay inter-play of the following

* Policies specifying behavioral expectations that people or systems have for themselves and others. For example, the designers of an online music system may specify policies how users can access and copy songs
* Permissions describing the behaviors that are allowed by the agents that interact with a person or a system. For instance, an online music store may provide permissions for limited access and copying for people who have purchased certain songs
* Protections describing mechanisms put in place to enforce permissions and policies. Using our running example of an online music store, we could imagine that such a system would build in protections to prevent people from unauthorized access and copying of its sources (trust directed from system to users)

Assurance involves the management of trust in two directions – from users to systems (e.g., valid use of credit card numbers for songs purchasing), and from system to users

Designers of computer systems want the people using the resources of their systems doing so in line with their policies. Also designers may want restricting time of use of their copy, or number of watches, or number of back-up copies of their music.

Thus, trust management deals with the design of effective, enforceable policies, methods for granting permissions to trusted users, and the components that can enforce those policies and permissions for protecting and managing the resources in the system.

Another important part of system assurance involves software engineering. The designers of a system need to know that the software that implements their system is coded so that it conforms to their design. For example, the designer of a system might specify that a Pseudo Random Number Generator (PRNG) is used for encryption with random numbers. If implementer, always uses the same seed value for PRNG then the numbers generated will be again repeated.

Thus, it is necessary having good specification and complying it implementation.

Also, for trust, it is necessary supporting some evidence of the operations conducted to be able using them in the case of going to court (say, in the case being cheated by a web site).

Authenticity

It is the ability to determine the statements, policies, and permissions issued by persons or systems are genuine. If such things can be faked, there is no way to enforce the implied contracts that people and systems engage in when buying or selling items online.

Nonrepudiation is the property that authentic statements issued by some person or system cannot be denied. Nonrepudiation is achieved by the use of digital signature.

Anonymity

There is an unfortunate side effect from using personal identities in electronic transactions (medical history, purchase history, legal records, e-mail communications, employment records). Therefore, anonymity is needed, which is the property that certain records or transactions not to be attributable to any individual.

If organizations need publishing data about their members or clients, we should expect that they do so in a privacy preserving fashion, using some of the following tools.

* Aggregation (sum, average, min, max, count)
* Mixing: the intertwining of transactions, information, or communications in a way that cannot be traced to any individual
* Proxies: trusted agents that are willing to engage in actions for an individual in a way that cannot be traced back to that person. Internet searching proxies are web sites that themselves provide an Internet browser interface, so that individuals can visit web sites that they might be blocked from, for instance, because of the country they are located in.
* Pseudonyms: fictional identities that can fill in for real identities in communications and transactions, but are otherwise known to a trusted party only. For example, many social networks allow users interacting user pseudonyms without revealing their actual identity.

PROTECTION

The introduction of multiprogramming leads to resources sharing by many users. This sharing involves not just the processor but also the following:

1. Memory
2. I/O devices, such as disks and printers
3. Programs
4. Data

The ability to share these resources introduced the need for protection. Operating system may offer protection along the following spectrum:

* + No protection: This is appropriate when sensitive procedures are being run at separate times.
  + Isolation: This approach implies that each process operates separately from other processes, with no sharing or communication. Each process has its own address space, files, and other objects.
  + Share all or share nothing: The owner of an object (e.g., a file or a memory segment) declares it to be public or private. In the former case, any process may access the object; in the latter, only the owner’s processes may access the object.
  + Share via access limitations: The operating system checks the permissibility of each access by a specific user to a specific object. The operating system acts as a guard between users and objects, ensuring that only authorized accesses occur.
  + Share via dynamic capabilities: This extends the concept of access control to allow dynamic creation of sharing rights for objects.
  + Limit use of an object: This form of protection limits not just access but the use to which the object may be put. For example, a user may be allowed to view a sensitive document, but not print it. Another example is that a user may be allowed access to a database to derive statistical summaries but not to determine specific data values.

Protection of Memory

In a multiprogramming environment, memory protection is essential. The concern here is not just security, but the correct functioning of the various processes.

The separation of the memory space of various processes is easily accomplished with a virtual memory scheme (segmentation, or paging, or combination of the two). There may be provided also sharing by allowing the same segment or page to appear in more than one table. There may be used hardware support for checking validity of access of the process to the object by comparison of the control key associated with the object, with the access requiring key associated with the process (e.g., RPL and DPL for Intel processors using four priority rings).

User-Oriented Access Control

The measures taken to control access in a data processing system fall into two categories: those associated with the user and those associated with the data.

The most common technique for user access control on a shared system or server is the user log on, which requires both a user identifier (ID) and a password.

Data-Oriented Access Control

Following successful logon, the user has been granted access to one or a set of hosts and applications. This is generally not sufficient for a system that includes sensitive data in its database. Through the user access procedure, a user can be identified to the system. Associated with each user, there can be a profile that specifies permissible operations and file accesses. The database management system, however, must control access to specific records or even portions of records. For example, it may be permissible for anyone in administration to obtain a list of personal, but only selected individuals may have access to salary information. Operating system may grant a user access to a file or use an application, following which there are no more security checks, while the database management system must make a decision on each individual access attempt. That decision will depend not only on the user’s identity but also on the specific parts of data being accessed and even on the information already divulged to the user.

Data-Oriented Access Control (Cont)

A general model of access control as exercised by a file or database management system is that of an access matrix:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | File 1 | File 2 | File 3 | File 4 | Account 1 | Account 2 |
| User A | Own  R  W |  | Own  R  W |  | Inquiry credit |  |
| User B | R | Own  R  W | W | R | Inquiry debit | Inquiry credit |
| User C | R  W | R |  | Own  R  W |  | Inquiry debit |

Access control lists for files are obtained from access matrix by taking content of its columns, and Capability lists (tickets) for users are obtained from the matrix by taking its rows. For example, Access control list for File 1 is:

File 1 => <A|Own,R,W> => <B|R> => <C|R,W>

Capability list for User B is:

User B => <File 1|R> => <File 2|Own, R,W> => <File 3|W> => <File 4|R> => <Account 1|Inquiry debit> => <Account 2|Inquiry credit>

Such lists allow keeping sparse access matrix effectively.

The basic elements of this model are as follows:

* Subject: An entity capable of accessing objects. Generally, the concept of subject equates with that of process. Any user or application actually gains access to an object by means of a process that represents that user or application
* Object: Anything to which access is controlled. Examples include files, portions of files, programs, and segments of memory.
* Access right: The way in which an object is accessed by a subject. Examples are read, write, and execute.

Capability lists (tickets) specify authorized object and operations for user. User may be authorized to loan tickets or give them to others. Because tickets may be dispersed around the system, they present a greater security problem than access control lists. In particular, tickets must be un-forgeable. One way to accomplish this is to have the operating system hold all tickets on behalf of users. These tickets would have to be held in a region inaccessible to users.

INTRUDERS

One of the most publicized threat to security is the intruder (the other is viruses), generally referred to as hacker or cracker. There can be identified three classes of intruders:

* Masquerader: An individual who is not authorized to use the computer and who penetrates a system’s access controls to exploit a legitimate user’s account
* Misfeasor: A legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access but misuses his or her privileges
* Clandestine user: An individual who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection.

The masquerader is likely to be an outsider; the misfeasor generally is an insider; and the clandestine user can be either an insider or an outsider.

Intrusion Techniques

The objective of the intruder is to gain access to a system or to increase the range of privileges accessible on a system. This requires the intruder to acquire information that should have been protected. In most cases, this information is in the form of a user password. With knowledge of some other user’s password, an intruder can log in to a system and exercise all the privilege accorded to the legitimate user.

Typically, a system must maintain a file that associates a password with each authorized user. If such a file is stored with no protection, then it is an easy matter to gain access to it and learn passwords. There is no "password file" under [Microsoft Windows NT](http://www.tech-faq.com/windows-password-file.shtml)/2000/XP. Microsoft Windows NT/2000/XP stores encrypted password hashes in the Windows Registry (<https://en.wikipedia.org/wiki/Security_Account_Manager> ). The password file for Unix is located in /etc and is a text file called passwd (<https://en.wikipedia.org/wiki/Passwd>). The password file can be protected in one of two ways:

* One-way encryption: The system stores only an encrypted form of the user’s password. When the user presents a password, the system encrypts that password and compares it with the stored value. In practice, the system usually performs a one-way transformation (not reversible) in which the password is used to generate a key for the encryption function and in which a fixed-length output is produced.
* Access control: Access to the password file is limited to one or very few accounts.

Intrusion Techniques (Cont)

If one or both of these countermeasures are in place, some effort is needed for a potential intruder to learn passwords. The following techniques for learning passwords are known:

* 1. Try default passwords used with standard accounts that are shipped with the system. Many administrators do not bother to change them.
  2. Exhaustively try all short passwords (those of one to three characters)
  3. Try words in the system’s online dictionary or a list of likely passwords. Examples of the latter are readily available on hacker bulletin boards (e.g., <http://www.openwall.com/passwords/wordlists/>).
  4. Collect information about users, such as their full names, the names of their spouse and children, pictures in their office, and books in their office that are related to hobbies.
  5. Try users’ phone numbers, social security numbers, and room numbers.
  6. Try all legitimate license plate numbers for this state.
  7. Use Trojan horse (described later) to bypass restrictions on access.
  8. Tap the line between a remote user and the host system.

The first six methods are various ways of guessing a password. If an intruder has to verify the guess by attempting to log in, it is tedious and countered means of attack. For example, a system can simply reject any login after three password attempts, thus requiring the intruder to reconnect to the host to try again. Under these circumstances, it is not practical to try more than handful of passwords. However, intruder is unlikely to try such crude methods. For example, if an intruder can gain access with a low level of privileges to an encrypted password file, then the strategy would be to capture that file and then use the encryption mechanism of that particular system at leisure until a valid password that provided greater privileges was discovered.

Guessing attacks are feasible, and indeed highly effective, when large number of guesses can be attempted automatically and each guess verified, without the guessing process being detectable.

The seventh method of attack listed earlier, the Trojan horse, can be particularly difficult to counter. For example, a low-privileged user produced a game program and invited the system operator to use it in his or her spare time. The program did indeed play a game, but in the background it also contained code to copy password file, which was unencrypted but access protected, into the user’s file. Because the game was running under the operator’s high-privilege mode, it was able to gain access to the password file.

The eighth attack listed, line tapping, is a matter of physical security. It can be countered with link encryption technique.

We turn now to a discussion of the two principal countermeasures: prevention and detection. Prevention is a challenging security goal and an uphill battle at all times. The difficulty stems from the fact that the defender must attempt to thwart all possible attacks, whereas the attacker is free to try to find the weakest link in the defense chain and attack at that point. Detection is concerned with learning of an attack after its success.