Unit 12 Security and Protection

Structure:

12.1 Introduction

In the previous unit, we have discussed the role of operating systems in distributed environment. Unlike computers in distributed environment, personal computers are designed and intended for individual use. Hence security and protection features were minimal. No two users could simultaneously use the same machine. Locking the room physically which housed the computer and its accessories could easily protect data and stored information. But today hardware costs have reduced and people have access to a wide variety of computing equipment. With a trend towards networking, users have access to data and code present locally as well as at remote locations. The main advantages of networking like data sharing and remote data access have increased the requirements of security and protection. Security and protection are the two main features that motivated development of a network operating system (example Novell NetWare). In this unit, let's discuss the methods of data security and protection.

Objectives:

After studying this unit, you should be able to:

- explain various attacks on computer security
- discuss different types of computer viruses and worms

- describe various security design principles
- explain protection mechanisms and security in distributed environment

12.2 Attacks on Security

Security is an important aspect of any operating system. Open Systems Interconnection (OSI) defines the elements of security in the following terms:

- *Confidentiality:* Information is not accessed in an unauthorized manner (controlled read)
- *Integrity:* Information is not modified or deleted in an unauthorized manner (controlled write)
- *Availability:* Information is available to authorized users when needed (controlled read / write / fault recovery)

Security is concerned with the ability of the operating system to enforce control over storage and movement of data in and between the objects that the operating system supports.

Major threats to security can be categorized as:

- Tapping
- Disclosure
- Amendment
- Fabrication
- Denial

Unauthorized use of service (tapping) and unauthorized disclosure of information (disclosure) are passive threats whereas unauthorized alteration or deletion of information (amendment), unauthorized generation of information (fabrication) and denial of service to authorized users (denial) are active threats. In either tapping or disclosure, information goes to a third party. In the former, information is accessed by the third party without the knowledge of the other two parties and in the latter the source willingly / knowingly discloses it to the third party.

A security system can be attacked in many ways. Some of them are discussed below:

Authentication

Authentication is verification of access to system resources. Penetration is by an intruder who may:

- Guess / steal somebody’s password and use it
- Use vendor supplied password usually used by system administrator for purposes of system maintenance
- Find a password by trial and error
- Use a terminal to access information that has been logged on by another user and just left like that.
- Use a dummy login program to fool a user

Browsing

Browsing through system files could get intruder information necessary to access files with access controls which are very permissive thus giving the intruder access to unprotected files / databases.

Invalid parameters

Passing of invalid parameters of failure to validate them properly can lead to serious security violations.

Line tapping

A communication line is tapped and confidential data is accessed or even modified. Threat could be in the form of tapping, amendment or fabrication.

Improper access controls

If the system administrator has not planned access controls properly, then some users may have too many privileges and others very few. This amounts to unauthorized disclosure of information or denial of service.

Rogue software

A variety of software programs exist under this title. Computer virus is very well known among others. This is a deliberately written program or part of it intended to create mischief. Such programs vary in terms of complexity or damage they cause. Creators of this software have a deep knowledge of the operating system and the underlying hardware. Other rogue software includes Trojan horse, Chameleon, Software bomb, Worm, etc.

The above mentioned were some common ways in which a security system could be attacked. Other ways in which a security system can be attacked may be through Trap doors, Electronic data capture, Lost line, Waste recovery and Covert channels.

Self Assessment Questions

1. Security is concerned with the ability of the operating system to enforce control over storage and movement of data in and between the objects that the operating system supports. (True / False)
2. _______________ refers to the Information that is not modified or deleted in an unauthorized manner.
3. _______________ is verification of access to system resources. (Pick the right option)
   a) Authentication
   b) Line Trapping
   c) Browsing
   d) Fabrication

12.3 Computer Worms

A computer worm is a full program by itself. It spreads to other computers over a network and while doing so consumes network resources to a very large extent. It can potentially bring the entire network to a halt.

The invention of computer worms was for a good purpose. Research scientists at XEROX PARC research center wanted to carry out large computations. They designed small programs (worms) containing some identified piece of computations that could be carried out independently and which could spread to other computers. The worm would then execute on a machine if idle resources were available or else it would hunt the network for machines with idle resources.

A computer worm does not harm any other program or data but spreads, thereby consuming large resources like disk storage, transmission capacity, etc. thus denying them to legal users. A worm usually operates on a network. A node in a network maintains a list of all other nodes on the network and also a list of machine addresses on the network. A worm program accesses this list and using it copies itself to all those address and spreads. This large continuous transfer across the network eats up network resources like line capacity, disk space, network buffers, tables, etc.

Two major safeguards against worms are:

- *Prevent its creation:* through strong security and protection policies

- *Prevent its spreading:* by introducing checkpoints in the communication system and disallowing transfer of executable files over a network unless until they are permitted by some authorized person.

12.4 Computer Virus

A computer virus is written with an intention of infecting other programs. It is a part of a program that piggybacks on to a valid program. It differs from the worm in the following ways:

- Worm is a complete program by itself and can execute independently whereas virus does not operate independently.
- Worm consumes only system resources but virus causes direct harm to the system by corrupting code as well as data.

Types of viruses

There are several types of computer viruses. New types get added every now and then. Some of the common varieties are:

- Boot sector infectors
- Memory resident infectors
- File specific infectors
- Command processor infectors
- General purpose infectors

Infection methods

Viruses infect other programs in the following ways:

- *Append:* virus code appends itself to a valid unaffected program
- *Replace:* virus code replaces the original executable program either completely or partially
- *Insert:* virus code gets inserted into the body of the executable code to carry out some undesirable actions
- *Delete:* Virus code deletes some part of the executable program
- *Redirect:* The normal flow of a program is changed to execute a virus code that could exist as an appended portion of an otherwise normal program.

Mode of operation

A virus works in a number of ways. The developer of a virus (a very intelligent person) writes an interesting program such as a game or a utility knowing well the operating system details on which it is supposed to execute. This program has some embedded virus code in it. The program is then distributed to users for use through enticing advertisements and at a low price. Having bought the program at a throwaway price, the user copies it into his / her machine not aware of the devil which will show up soon. The virus is now said to be in a nascent state. Curious about the output of the program bought, the user executes it. Because the virus is embedded in the host program being run, it also executes and spreads thus causing havoc.

Virus detection

Virus detection programs check for the integrity of binary files by maintaining a checksum and recalculating it at regular intervals. A mismatch indicates a change in the executable file, which may be caused due to tampering. Some programs are also available that are resident in memory and continuously monitor memory and I/O operations.

Virus removal

A generalized virus removal program is very difficult. Anti-virus codes for removal of viruses are available. Bit patterns in some virus code are predictable. The anti-virus programs scan the disk files for such patterns of the known virus and remove them. But with a number of viruses cropping up every now and then, development and availability of anti-virus for a particular type is delayed and harm done.

Virus prevention

‘Prevention is better than cure’. As the saying goes, there is no good cure available after infection. One of the safest ways to prevent virus attacks is to use legal copies of software. Also system needs to be protected against use of unauthorized / unchecked floppy disks. Frequent backups and running of monitoring programs help detection and subsequent prevention.

Self Assessment Questions

4. Computer worm harms other programs and data. (True / False)
5. A computer _______________ is written with an intention of infecting other programs.
6. _______________ software is used to remove viruses from programs.

12.5 Security Design Principles

General design principles for protection put forward by Saltzer and Schroeder can be outlined as under:

- *Public design:* a security system should not be a secret, an assumption that the penetrator will know about it is a better assumption.
- *Least privileges:* every process must be given the least possible privileges necessary for execution. This assures that domains to be protected are normally small. But an associated overhead is frequent switching between domains when privileges are updated.
- *Explicit demand:* access rights to processes should not be granted as default. Access rights should be explicitly demanded. But this may result in denial of access on some ground to a legal user.
- *Continuous verification:* access rights should be verified frequently. Checking only at the beginning may not be sufficient because the intruder may change access rights after initial check.
- *Simple design:* a simple uniform security system built in layers, as an integral part of the system is preferred.
- *User acceptance:* Users should not have to spend a lot of effort to learn how to protect their files.
- *Multiple conditions:* wherever possible, the system must be designed to depend on more than one condition, for example, two passwords / two keys.

12.6 Authentication

Authentication is a process of verifying whether a person is a legal user or not. This can be by either verification of users logging into a centralized system or authentication of computers that are to work in a network or a distributed environment.

Password is the most commonly used scheme. It is easy to implement. User name is associated with a password. This is stored in encrypted form by the system. When the user logs onto the system, the user has to enter his user name and password against a prompt. The entered password is then encrypted and matched with the one that is stored in the file system. A tally will allow the user to login. No external hardware is needed. But limited protection is provided.

The password is generally not echoed on the screen while being keyed in. Also it is stored in encrypted form. It cannot be deciphered easily because knowing the algorithm for deciphering will not suffice as the key is ought to be known for deciphering it.

Choosing a password can be done by the system or by the system administrator or by the users themselves. A system-selected password is not a good choice as it is difficult to remember. If the system administrator gives a user a password then more than one person knows about it. User chosen passwords is practical and popular. Users should choose passwords that are not easy to guess. Choosing user names, family names, names of cities, etc are easy to guess.

Length of a password plays an important role in the effectiveness of the password. If it is short it is easy to remember and use but easy to decipher too. Longer the password it is difficult to break and also to remember and key in. Trade off results in a password of length 6-8 characters.

Salting is a technique to make it difficult to break a password. Salting technique appends a random number ‘n’ to the password before encryption is done. Just knowing the password is not enough. The system itself calculates, stores and compares these random numbers each time a password is used.

Multiple passwords at different levels could provide additional security. Change of password at regular intervals is a good practice. Many operating systems allow a user to try only a few guesses for a login after which the user is logged off the system.

12.7 Protection Mechanism

System resources need to be protected. Resources include both hardware and software. Different mechanisms for protection are as follows:

Files need to be protected from unauthorized users. The problem of protecting files is more acute in multi-user systems. Some files may have only read access for some users, read / write access for some others, and so on. Also a directory of files may not be accessible to a group of users. For example, student users do not access to any other files except their own. Like files devices, databases, processes also need protection. All such items are grouped together as objects. Thus objects are to be protected from subjects who need access to these objects.

The operating system allows different access rights for different objects. For example, UNIX has read, write and execute (rwx) rights for owners, groups and others. Possible access rights are listed below:

- No access
- Execute only
- Read only
- Append only
- Update
- Modify protection rights
- Delete

A hierarchy of access rights is identified. For example, if update right is granted then it is implied that all rights above update in the hierarchy are granted. This scheme is simple but creation of a hierarchy of access rights is not easy. It is easy for a process to inherit access rights from the user who has created it. The system then need maintain a matrix of access rights for different files for different users.

The operating system defines the concept of a domain. A domain consists of objects and access rights of these objects. A subject then gets associated with the domains and access to objects in the domains. A domain is a set of access rights for associated objects and a system consists of many such domains. A user process always executes in any one of the domains. Domain switching is also possible. Domains in the form of a matrix are shown in table 12.1.

Table 12.1: Domains in matrix form

		OBJECTS							
		File 0	**File 1**	**File 2**	**File 3**	**File 4**	**File 5**	**Printer 0**	**Printer 1**
D O M A I N S	0	R W	R					W	
	1			R	R W X				W
	2					W	R W X		W
	3			W		R			

A variation of the above scheme is to organize domains in a hierarchy. Here also a domain is a set of access rights for associated objects. But the protection space is divided into ‘n’ domains from 0 to (n-1) in such a way that domain 0 has maximum access rights and domain (n-1) has the least. Domain switching is also possible. A domain switch to an outer domain is easy because it is less privileged whereas a domain switch to an inner domain requires permissions.

Domain is an abstract concept. In reality domain is a user with a specific id having different access rights for different objects such as files, directories and devices. Processes created by the user inherit all access rights for that user. An access control matrix showing users and objects (files) needs to be stored by the operating system in order to decide granting of access rights to users for files.

Since the matrix has many holes, storing the entire matrix is waste of space. Access control list is one way of storing the matrix. Only information in the columns is stored and that too only where information is present that is each file has information about users and their access rights. The best place to maintain this information is the directory entry for that file.

Capability list is another way of storing the access control matrix. Here information is stored row wise. The operating system maintains a list of files/ devices (objects) that a user can access along with access rights. A combination of both access control list and capability list is also possible.

Self Assessment Questions

7. Password is the most commonly used scheme for protection against illegal access. (True / False)
8. General design principles for protection were put forward by _______________ and _______________.
9. _______________ is a way of storing the access control matrix.

12.8 Encryption

Encryption is an important tool in protection, security and authentication. The process involves two steps (Refer figure 12.1):

- *Encryption:* the original message is changed to some other form
- *Decryption:* the encrypted message is restored back to the original

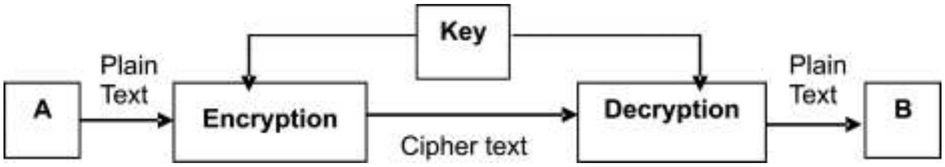

Fig. 12.1: Conventional Encryption

Data before encryption is called plain text and after encryption is called cipher text. Usually the above operations are performed by hardware.

Encryption could be by one of the following two basic methods:

- Transposition ciphers
- Substitution ciphers

In transposition ciphers the contents of the data are not changed but the order is changed. For example, a message could be sent in reverse order like:

I am fine → enif ma I

Rail fence cipher is a method that belongs to this class. The method is slow because the entire message is to be stored and then encrypted. It also requires more storage space when messages are long.

Substitution ciphers work by sending a set of characters different from the original like:

I am fine → r zn ormv

Caesar cipher is a popular method of this type. This method is fast and requires less memory because characters can be changed as they are read and no storage is required. Variations of this scheme are used for bit streams. Encryption in this case involves adding a key to every bit stream and decryption is removing the key from the cipher text. Thus every algorithm has a key. It must ensure restoration. Normally a single piece of hardware is responsible for both encryption and decryption.

In the conventional encryption scheme two parties A and B agree upon a key. Someone say A or B or a third party has to decide upon this common key; get concurrence from concerned parties and initiate communication. This is called key distribution. Each pair of nodes needs a unique key. If there are ‘n’ nodes then there will be n*(n-1)/2 keys. If ‘n’ is large then the number of keys will also be large. Deciding, conveying and storing these keys is a mammoth job. Tapping can take place. This is the key distribution problem.

An alternate is the public key encryption. Keys used for encryption and decryption are not the same. Key K1 is used for encryption and another key K2 is used for decryption. A message encrypted using K1 can be decrypted only using K2 and not K1. One of the keys is publicly known. Hence this type of encryption is called public key encryption. Decryption is done using a private key and hence information cannot leak out. Interchange of keys K1 and K2 is possible, that is, K2 to encrypt and K1 to decrypt.

Each user has two keys, one public and one private (See figure 12.2). The private key is a secret but the user publishes the public key to a central key database. The database maintains public keys of different users.

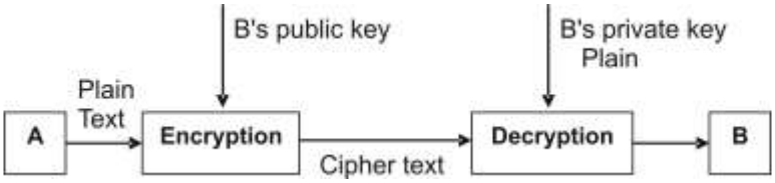

Fig. 12.2: Public key Encryption

Encryption and decryption are as follows:

- A wants to send a message to B.
- A searches the database of public keys for the public key of B.
- A encrypts the data using B’s public key.

- The cipher text is sent to B.
- B receives this cipher text.
- B decrypts the received cipher text using its private key and reads the message.

The problem here is that of authentication. B does not know who has sent the message to it because everybody knows B’s public key. In the conventional encryption method a single key is used between two parties and hence the receiver knows the sender. But it suffers from the problem of key distribution. In public key encryption method, for ‘n’ nodes in the network only 2*n keys (1 public and 1 private for each of the nodes) are required. There need be no agreement. Private key is chosen and a public key is made known. Key distribution is really not necessary. Key leakage and tapping are minimal. Protection is ensured but authentication is not provided.

12.9 Security in Distributed Environment

Security problems in a distributed environment are complex. Messages through a network can be tapped at multiple locations. For an active attack the intruder gets control over a link so that data modification / deletion is possible. For a passive attack the intruder just listens to a link and uses the passing information.

Encryption in a distributed environment can be of two forms:

- End-to-end encryption
- Link encryption

If end-to-end encryption is used, the encryption / decryption devices are needed only at the ends. Data from source to destination moves on the network in encrypted form. In packet switched networks, data is sent in the form of packets. Each packet has control information (source address, destination address, checksum, routing information, etc.) and data. Since routing address is needed for the packet to hop from the source till it reaches the destination, the control information cannot be encrypted as there is no facility to decrypt it anywhere in between. Only the data part in a packet can be encrypted. The system thus becomes vulnerable for tapping.

Link encryption needs more encryption / decryption devices, usually two for each link. This allows total encryption of a packet and prevents tapping. The method is expensive and slow. A combination of both is possible.

Message authentication allows users to verify that data received is authentic. Usually the following attributes of a user need to be authenticated:

- Actual message
- Time at which sent
- Sequence in which sent
- Source from which it has arrived

Common methods for message authentication are:

- Authentication code
- Encryption
- Digital signatures

In authentication code, a secret key is used to generate a check sum, which is sent along with the data. The receiver performs the same operation using the same secret key on the received data and regenerates the check sum. If both of them are same then the receiver knows the sender since the secret key is known to only both of them. Conventional encryption provides authentication but suffers from key distribution problems and public key encryption provides good protection but no authentication.

Digital signature is like a human signature on paper. If a signed letter is sent by A to B, A cannot deny having sent it to B (B has the signed copy) and B cannot refuse having got it (A has an acknowledgement for B having received it). This is what happens in a manual system and should happen in electronic messages as well.

As discussed earlier, public key encryption provides protection but not authentication. If we want to authentication without protection, reversal of the keys applied is a solution as shown in figure 12.3.

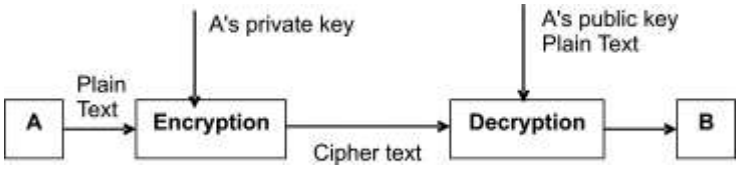

Fig. 12.3: Public key Encryption for authentication without protection

This is based on the concept that public key encryption algorithm works by using either of the keys to encrypt and the other for decryption. A encrypts the message to be sent to B using its private key. At the other end B decrypts the received message using A's public key which is known to everybody. Thus B knows that A has sent the message. Protection is not provided as anyone can decrypt the message sent by A.

If both authentication and protection are needed then a specific sequence of public and private keys is used as shown in below figure 12.4. The two keys are used as shown. At points 2 and 4 the cipher text is the same. Similarly at points 1 and 5 the text is the same. Authentication is possible because between 4 and 5 decryption is done by A's public key and is possible only because A has encrypted it with its private key. Protection is also guaranteed because from point 3 onwards only B can decrypt with its private key. This is how digital signatures work.

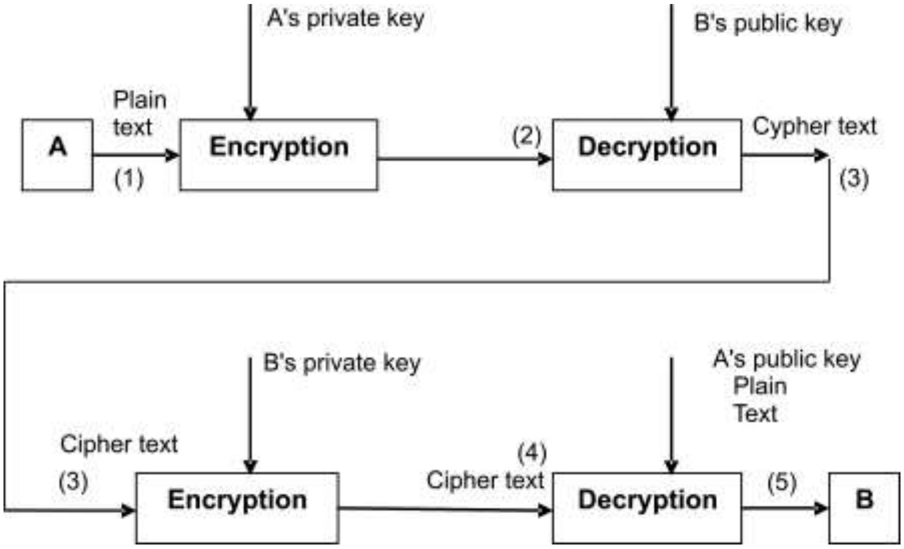

Fig. 12.4: Public key Encryption for both authentication and protection

Self Assessment Questions

10. Encryption is an important tool in protection, security and authentication. (True / False)
11. Data before encryption is called _______________ text and after encryption is called _______________ text.
12. __________ and _________ are the two common types of encryption.

12.10 Summary

Let’s recapitulate important points discussed in this unit:

- Security is concerned with the ability of the operating system to enforce control over storage and movement of data in and between the objects that the operating system supports.
- Unauthorized use of service (tapping) and unauthorized disclosure of information (disclosure) are passive threats whereas unauthorized alteration or deletion of information (amendment), unauthorized generation of information (fabrication) and denial of service to authorized users (denial) are active threats.
- A computer worm is a full program by itself. It spreads to other computers over a network and while doing so consumes network resources to a very large extent. It can potentially bring the entire network to a halt.
- A computer virus is written with an intention of infecting other programs. It is a part of a program that piggybacks on to a valid program.
- Encryption is an important tool in protection, security and authentication

12.11 Terminal Questions

1. Discuss the need for security and protection in computer systems.
2. Write a note on computer worm and computer virus.
3. Describe authentication by using passwords.
4. What is encryption? What are the different ways in which a message can be encrypted?

12.12 Answers

Self Assessment Questions

1. True
2. Integrity
3. a) Authentication
4. False
5. Virus
6. Anti-virus
7. True
8. Saltzer, Schroeder
9. Capability list

10. True
11. Plain, Cipher
12. Private key, Public key

Terminal Questions

1. Security is concerned with the ability of the operating system to enforce control over storage and movement of data in and between the objects that the operating system supports. (Refer section 12.2 for detail)
2. A computer worm is a full program by itself. It spreads to other computers over a network and while doing so consumes network resources to a very large extent. It can potentially bring the entire network to a halt. A computer virus is written with an intention of infecting other programs. It is a part of a program that piggybacks on to a valid program. (Refer sections 12.3 and 12.4)
3. Authentication is a process of verifying whether a person is a legal user or not. This can be by either verification of users logging into a centralized system or authentication of computers that are to work in a network or a distributed environment. (Refer section 12.6)
4. Encryption is an important tool in protection, security and authentication. Data before encryption is called plain text and after encryption is called cipher text. Usually the above operations are performed by hardware. (Refer section 12.8)