St. Joseph's College of Engineering & Technology Palai

Department of Computer Science & Engineering

S8 CS

RT801 Security in Computing

Module 2

# Security in Computing - Module 2

## Syllabus

OS Security - Protection Mechanisms - Authentication & Access contro -l Discretionary and Mandatory access control

Authentication mechanisms - Official levels of computer security (DoD) - Security breaches

Concept of a hole - Types of a holes

Study of the security features for authentication, access control and remote execution in UNIX, WINDOWS 2000

## Contents

# Security in Operating Systems

Operating system manages all the resources in a computer system. Several users may present in the operating system. The various objects in operating system such as files, processes, hardware devices must be protected from unauthorised access, malicious destruction or inconsistency.

# Part I. Protection Mechanisms

In this section, we will learn some of the mechanisms used by the operating system to protect files and other resources.

## 1 Protection Domain

A computer system consists of a collection of processes and objects. Objects such as CPU, memory, printer, disk, files, programs etc.. Every object has a unique identifier.

A set of operations can be performed on the different objects. These operations depend on the object. For example, a CD drive can be read. A file can be created, opened, read, written and deleted.

A process is allowed to access only those objects it is authorised to access. T ofacilitate this, a domain structure is defined.

### Domain structure

A process operates within a protection domain. This domain specifies the resources the process is allowed to access.

A domain is a set of objects and the types of operations that may be done on the object.

Normally users are categorised in to different domains. A system can consist of many user accounts such as system administrator, user1, user2 etc..

#### Owner

Every object has an owner. Owner is a user account who owns the object.

#### Group

Different users can form a group.

For example, user1, user2, user5 can form a group G1.

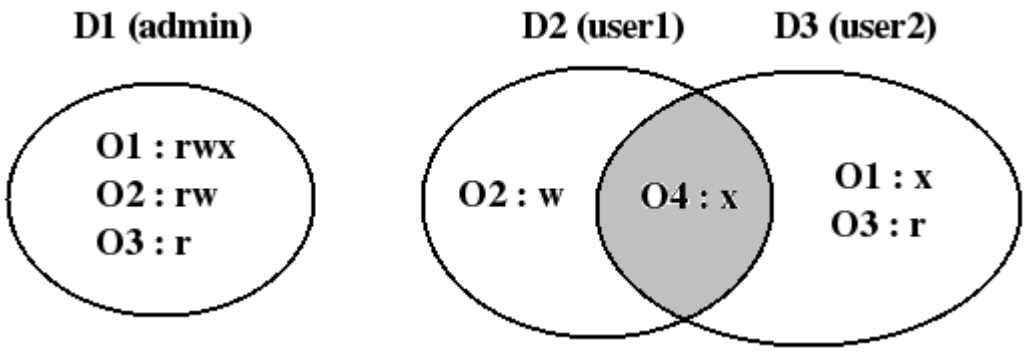
Normally the domains are categorised as,

owner,

group, and

others.

The following picture shows 3 domains.




In the above, we have 3 domains: D1, D2 and D3.

D1 is allowed to perform

read, write and execute operations on object O1.

read, write on object O2, and

only read on O3.

D2 is allowed to perform

write on object O2.

D2 and D3 can execute on object O4.

D3 is allowed to perform

execute operation on object O1.

only read on O3.

In the above scheme, user is defined as domain. Also a process can be a domain.

## 2 Protection Matrix (Access Control Matrix)

Our model of protection can be viewd as a matrix called protection matrix. Following shows a protection matrix:

| Object / Domain | F1 (k1.c) | F2 (c.pdf) | F3 (r.txt) | printer |
|---|---|---|---|---|
| D1 (admin) | r | | r | |
| D2 (user1) | | | | p |
| D3 (user2) | | r | x | |
| D4 (user3) | rw | | rw | |

The rows of the matrix represent domains. The columns represent objects. Each entry in the matrix consists of a set of access rights.

In the above matrix, there are 4 domains and 4 objects.

When a process executes in domain D1, it can read objects F1 and F3.

This means when a process executes on behalf of the admin user, it can read files k1.c and r.txt.

From the matrix, only user1 can access printer.

### Domain Switching

Using the protection matrix scheme, domain switching can also be done. We can switch a process from one domain to another. This can be done as follows:

| Object / Domain | F1 | F2 | F3 | printer | D1 | D2 | D3 | D4 |
|---|---|---|---|---|---|---|---|---|
| D1 | r | | r | | | switch | | |
| D2 | | | | p | | | switch | switch |
| D3 | | r | x | | | | | |
| D4 | rw | | rw | | switch | | | |

Here a process in domain D1 is allowed to switch to domain D2. Switching from D1 to D3 or D4 is not allowed.

### Implementation of Protection Matrix

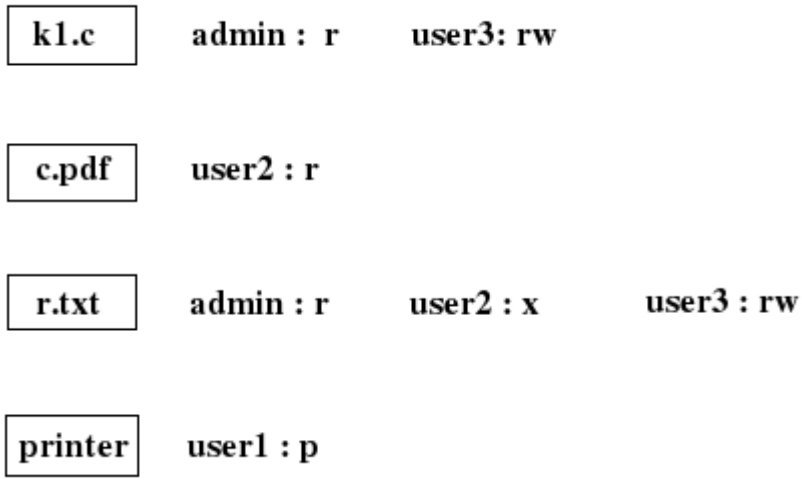
From the above, most of the positions in the protection matrix are empty. To implement this scheme efficiently, we will learn two mechanisms. They are,

Access control lists, and

Capability lists.

## 2.1 Access Control Lists

Here, each object is associated with a list. The list consists of all domains that may access the object. This list is called access control list or ACL. This is shown below:




Thus each file or object has an access control list associated with it.

File k1.c has two entries in its ACL. The first entry says that a process owned by admin may read k1.c. Second entry says that a process owned by user3 can read and write on k1.c. All other acccesses by these users and all accesses by other users are forbidden.

File c.pdf has one entry in its ACL.

### 2.2 Capability Lists

Another way to implement protection matrix is to arrange it by rows. Here associated with each user, there is a list of objects that may be accessed. With each object, the operations permitted are also shown.

| admin | user1 | user2 | user3 |
|---|---|---|---|
| ↓ | ↓ | ↓ | ↓ |
| k1.c : r | printer : p | c.pdf : r | k1.c : rw |
| r.txt : r | | r.txt : x | r.txt : rw |

This list is called capability list or C-List and the individual items are called capabilities.

In the above, a process owned by admin user can read on files c.pdf and execute on r.txt.

# Part II. Authentication

Whwn a user logs on to a computer, the operating system wishes to know who the user is. This is called user authentication.

Generally authentication is based on one of the following items:

1. User knowledge

   For example, a user name and a password, a personal identification number (PIN) or answer a prearranged set of questions.

2. User possession ( a key or card)

   For example, electronic key cards, smart cards or physical keys.

3. User attribute

   For example, recognition by finger print, retina, face, voice pattern, handwriting characteristics and typing rhythm.

### 3 Authentication using passwords

This is the most widely used form. User types a login name and password. Thsi is easy to implement.

The set of login names and passwords are stored in a file (/etc/passwd). When the user types his login name, it is looked up in the file and the typed password is compared to the stored password. If they match, login is allowed; otherwise login is rejected.

Normally, when the userr types the password, computer does not display typed characters. some systems display an asterisk (*) for each character and in others nothing is displayed when a password is typed.

### Problems with passwords

Passwords can often be guessed, accidently exposed, sniffed or illegally transferred between users.

#### Guessing

Intruders guess the password by using information about the user such as their spouse names, family details. The other way is to use enumeration, typing many combinations of letters.

#### Exposure

An intruder can get the password by watching the keyboard when the user types the password.

#### Sniffing

Also an intruder can use a network monitor which is a tool to watch all data transferred through a network including user ids and passwords.

#### Illegal transfer

If a user shares his account with others, security breach can occur. It may become impossible to determine who was using the ids at that time.

## Generation of Passwords

Passwords can be generated by system or by user selection.

## Improving Password Security

The use of password scheme for authentication can be improved by

1. Enforcing minimum number of characters in a password,
   For example, password should contain a minimum of 6 characters.
2. Passwords should contain uppercase and lowercase letters,
3. Passwords should contain at least one digit or special character,
4. Passwords should not be dictionary words, people's names,
5. Using one time passwords or use password aging.

Password aging forces users to change their passwords at regular intervals. For example every 3 months. At the extreme, password is to be changed for each session (one time passwords).

#### Encrypted Passwords

User names and passwords are stored in a file. To store theese passwords securely, many systems use encryption.

The operating system contains a function, f.

For the password, x, the system computes f(x). Thsi function value is stored in the password file.

When a user enters his password, x, f(x) is computed and compared with the value in the password file.

Normally the password file can be accessed only by the administrator user.

# 4 Challenge Response Authentication

**Scheme 1:**

Here the user selects a number of questions and store the answers for them.

When he logs in, system asks these questions and if the user gives the correct answer, he is allowed to log in.

Possible questions are,

1. what is your pet's name?
2. Where is your high school?
3. Who is your favourite singer?

**Scheme 2:**

In this scheme, user picks a function when he signs as a new user. When the user logs in, the server sends an argument, the user has to compute the value of the argument using his function.

For example, let the user selects the function, $f(x) = x^2$.

Then, when the server sends an argument, say, 7, user has to type 49 and login is allowed.

# 5 Authentication using Physical Objects

Physical objects such as memory cards, smart cards can be used for authentication.

**Memory cards**

Memory cards can store data. For example, a bank card contains a magnetic stripe on the back. It stores a simple security code which can be read using a card reader.

Here a card reader at a bank ATM center is attached to the bank computer. When a user wants to access the system, he needs to use the card and he needs to type a pin number.

The security code in the card is read by the card reader. This value and pin number is sent to the remote bank computer and user is authenticated.

This mechanism provides more security than a passsword.

**Smart cards**

Smart card contain an embedded microprocessor. An advantage of smart cards is that they do not need an online connection with the bank for withdrawing money.

Smart card is a very small computer that can communicate with a central computer to authenticate a user.

User authentication can be done in different ways using a smart card,

1. Static

   A user authenticates himself to the smart card and then the smart card authenticates the user to the computer.

2. Dynamic password generator

Here a smart card generates a unique password periodically (every month). The password is then entered in to the computer either manually or electronically by the smart card.

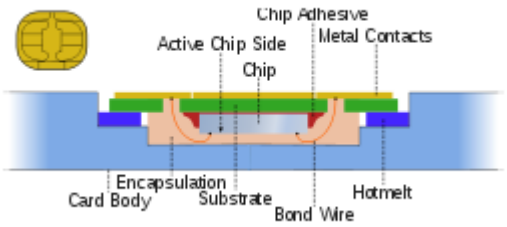
Here the computer system should know the password.

3. Challenge Response

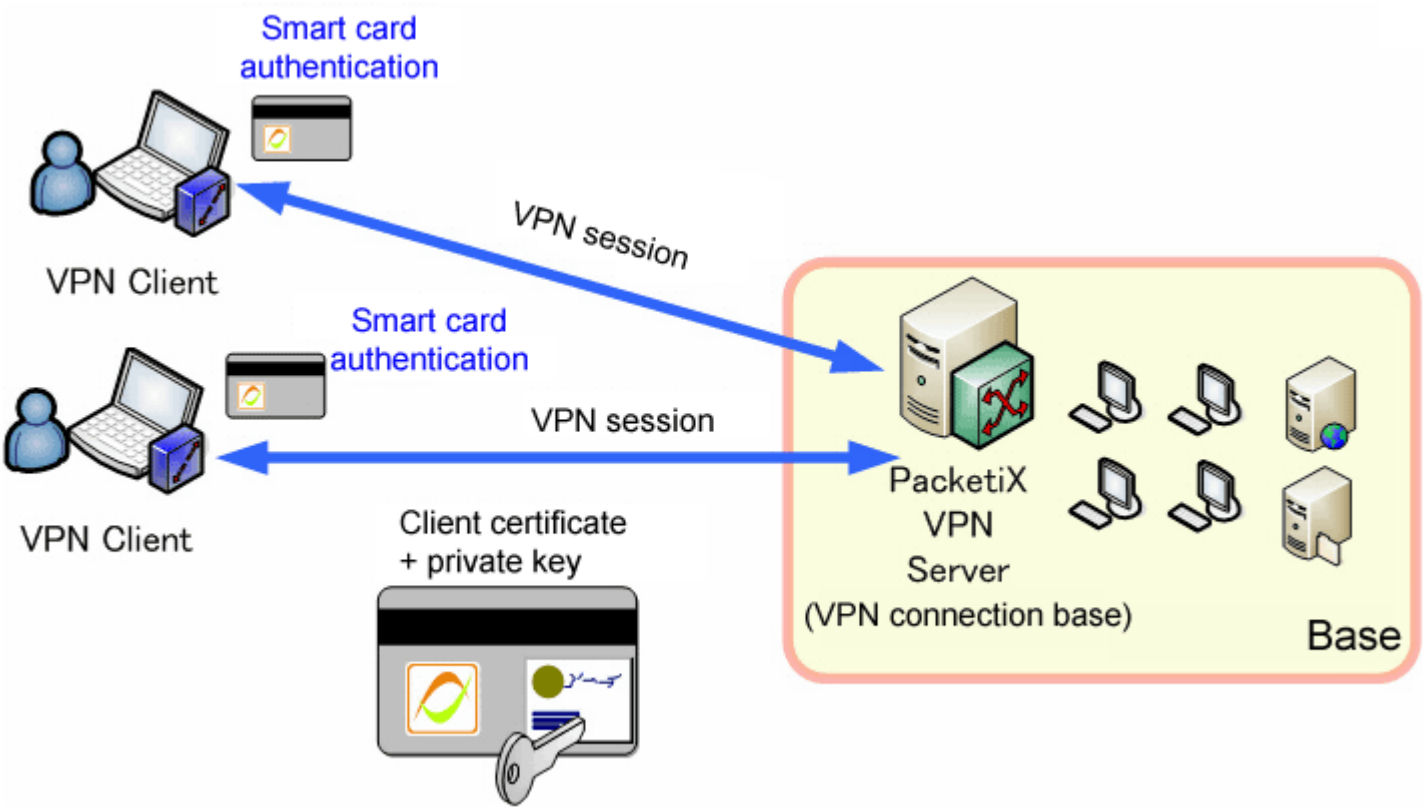
Here computer sends a random number to the smart card. The smart card combines this random number with user's password. Then it computes the value of a previously stored function from this combination. The result is then sent to the computer and user is authenticated.

A smart card has an electronic interface, normally a reader and can use one of the above schemes for authentication.

Thus a smart card contains a microprocessor, memory and I/O ports.








## 6 Biometric Authentication

This authentication method measures the physical characteristics of the user. Those physical characteristics are called biometrics.

For example, a finger print or voice print reader can authenticate a user.

facial characteristics, hand geometry, retinal pattern, iris, signature, voice etc..

A biometric system has two parts:

#### Enrollment

During enrollment, user's characteristics are measured and the results digitized. The significant features are stored in a central database.

#### Identification

The user shows up and provides a login name. The system makes the measurements again. If the new values match the stored values, login is allowed.

The most common physical characteristics are the following:

#### Facial characteristics

The common approach is to define characteristics based on relative location and shape of features such as eyes, eyebrows, nose, lips and chin shape.

Another approach is to produce a face thermogram using an infrared camera.

#### Finger prints

A finger print is a pattern of ridges and furrows on the surface of the finger tip. The yare believed to be unique across entire human race.

#### Hand geometry

It identifies features of the hand including shape, length and width of fingers.

#### Retinal pattern

The pattern formed by veins beneath the retinal surface is unique. A retinal biometric system obtains a digital image of the retinal pattern by projecting a beam of infrared light into the eye.

#### Iris

The detailed structure of the iris is unique and this can be used.

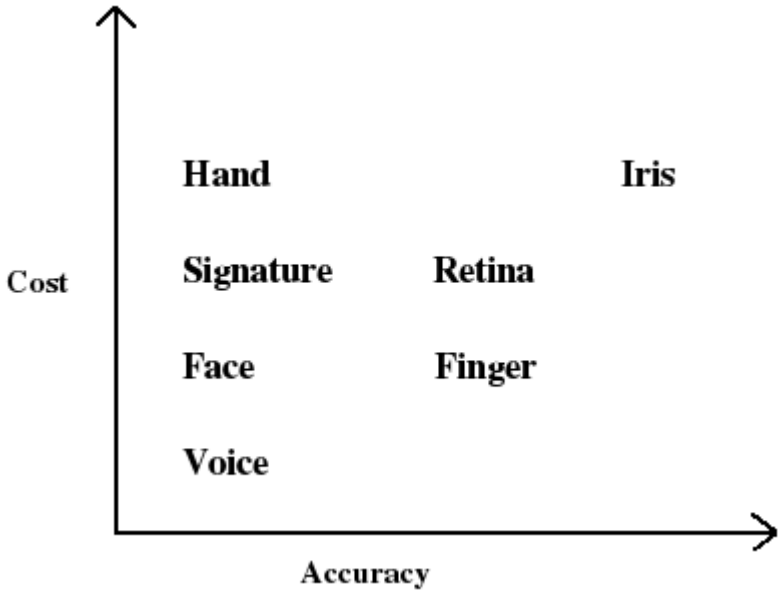
#### Signature

Each individual has a unique style of handwriting and this feature can be used for authentication purposes.

#### Voice

Voice patterns can be used for authentication.

The following figure shows the accuracy of these features:




# Part III. Access Control

An access control policy indicates what types of accesses are permitted, under what circumstances, and by whom.

Access control policies are grouped into,

Discretionary Access Control (DAC), and

Mandatory Access Control (MAC).

## 7 Discretionary Access Control (DAC)

In discretionary access control, individual users are allowed to determine who may read and write their files and other objects. This works fine in normal environments.

A model for implementing DAC is given below:

Consider the following access control matrix (protection matrix):

| | Users | | Files | | Processes | |
|---|---|---|---|---|---|---|
| | **u1** | **u2** | **F1** | **F2** | **P1** | **P2** |
| **u1** | Control | Owner | Read * | Read<br>Owner | Wake up | Wake Up |
| **u2** | | Control | Write * | **Execute** | | |

From the above, attrributes are control, owner, read, write, wakeup, read* etc..

Here read* operation of user u1 on file F1 means that user u1 can transfer this right to some other users.

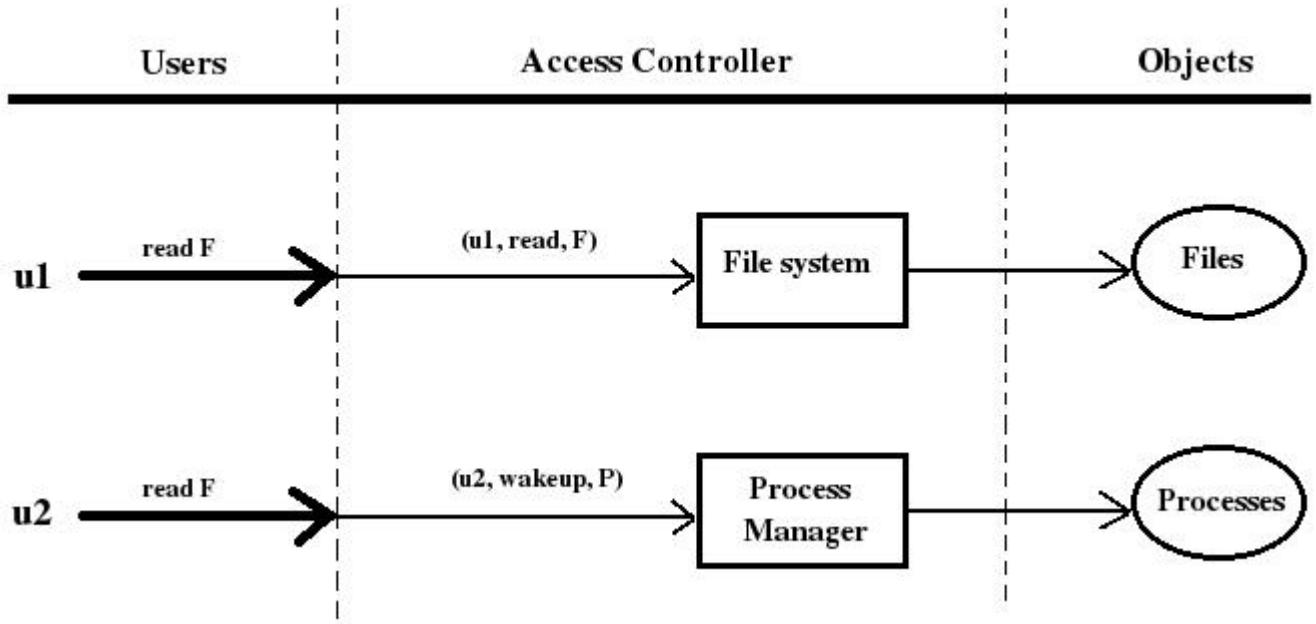
Here write* operation of user u2 on file F1 means that user u2 can transfer this right to some other users.

A separate access controller is associated with every object.

For all the files in the operating system, a file system controller is present.

For all processes, a process manager is present.

This is shown below:




When a user issues a request, the access controller checks the access rights of the object in the protection matrix.

Example 1:

For example,

1. User u1 issues a request read on file F1.

2. This causes the operating system to generate the message (u1, read, F1) to the file system controller.

3. The file system controller refers the protection matrix to determine whether a read is allowed on F1 for user u1. Here read is present. So access is allowed. If it was not present, read is not allowed.

Example 2:

1. User u1 issues the command to transfer 'read on F1' operation to user u2.

2. This causes the operating system to generate the message (u2, transfer, read) to the file system controller.

3. The file system controller checks the protection matrix. It finds an entry 'read*' in row u1 and column F1. So transfer of read right is allowed and read operation on F1 is added as an entry for user u2.

The new protection matrix is shown below:

| | Users | | Files | | Processes | |
|---|---|---|---|---|---|---|
| | **u1** | **u2** | **F1** | **F2** | **P1** | **P2** |
| **u1** | Control | Owner | Read * | Read<br>Owner | Wake up | Wake Up |
| **u2** | | Control | Write *<br>Read | **Execute** | | |

Note that a new entry is added for user u2 on file F1.

# 8 Mandatory Access Control

There are many environments where tighter security is needed such as military, hospitals etc.. Here the organisation has stated rules about who can see what, and these may not be modified by solidiers, lawyers or doctors, or at least without getting permission from their top authority. For such environments, mandatory access controls are needed.

Some of the models for mandatory access control are,

the Bell-La Padula model, and

the Biba model.

## 8.1 The Bell-La Padula Model

This model was designed for military security.

In the military world, all documents have a security level such as,

unclassified,

confidential,

secret, and

top secret.

People such as generals, lieutenants are assigned these levels depending on which documents they are allowed to see.

A general may be allowed to see all documents, whereas a lieutenant may be restricted to only few documents.

The rules in Bell-La Padula model are as follows:

**1. The simple security property**

A process running at security level, k can read only objects at its level or lower level.

For example, a general can read a lieutenant' documents, but a lieutenant cannot read a general's documents.

**2. The * property**

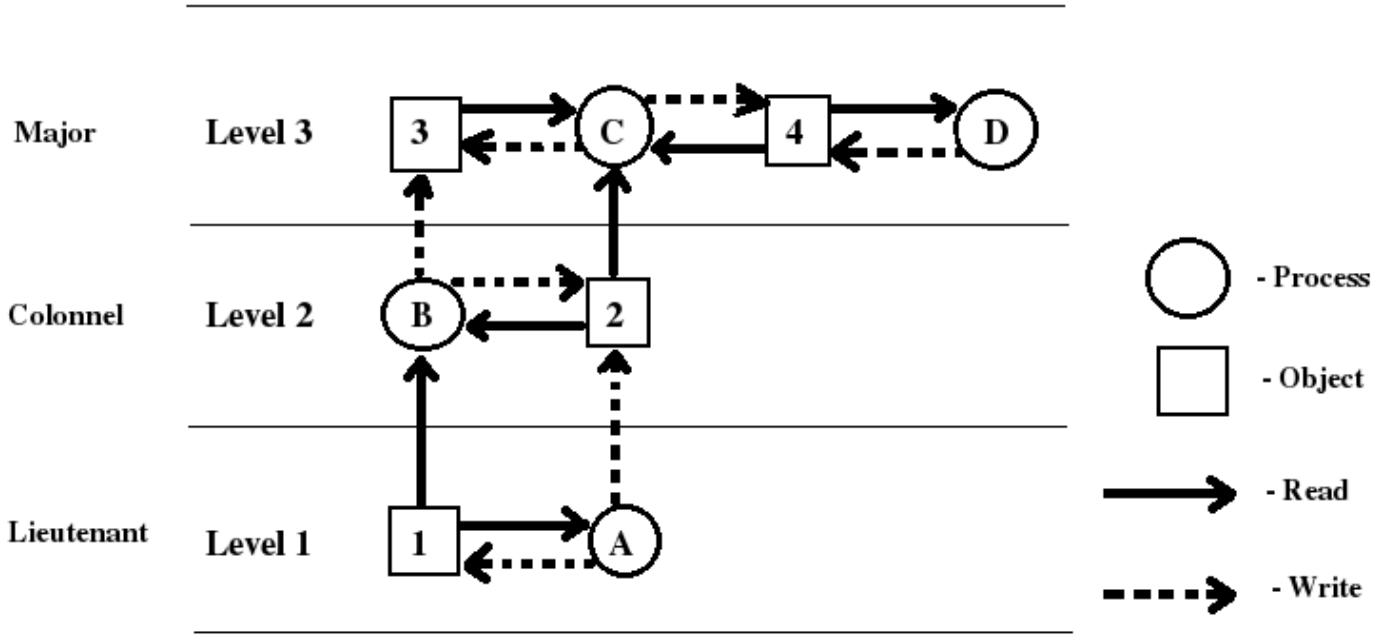
A process at security level,k can write only objects at its level or higher level.

For example, a lieutenant can write a message to a general's computer, but a general cannot write a message to a lieutenant' computer. Thsi is because the general may have seen top secret documents that should not be disclosed to a person in lower rank.

In short, processes can read down and write up, but not the reverse.

If the system uses these measures, no information will leak out from the higher security level.

This is shown below:




In the figure, solid arrow from object 1 to process B indicates that process B is reading the document 1.

The dashed arrow from process B to object 3 indicates that process B is writing on document 3.

From the above, the solid and dashed arrows always go sideways or up.

## 8.2 The Biba Model

For a commercial company or in civilian terms, the above military model may not work.

for example, suppose in a company, the programmers are at level 1, the project managers are at level 3, and president is at level 5.

Using the Bell-La padula model, a project manager can read from a programmer and he can overwrite the president's files. This is a serious problem. To compensate this, the Biba model was proposed.

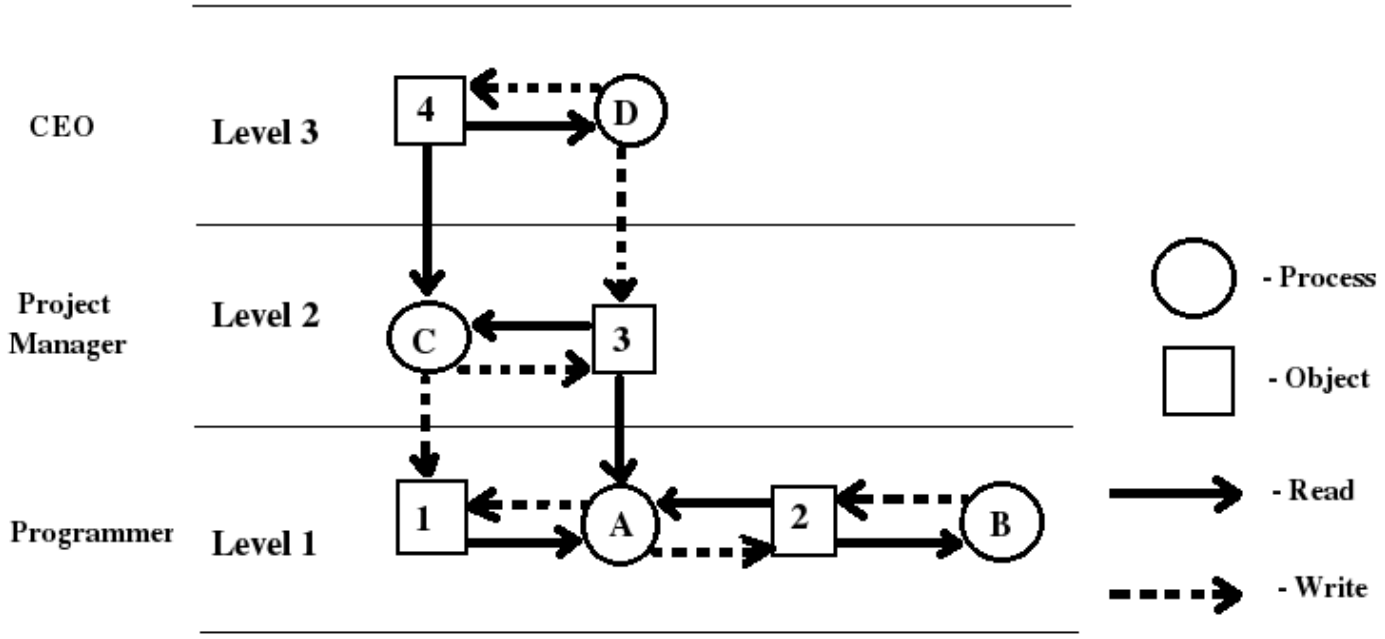
The rules in the Biba model are as follows:

1. The simple integrity principle

A process in security level, k can write only objects at its level or at lower level.

2. The integrity * property

A process running at security level, k can read only objects at its level or higher.




From this, a project manager can modify programmer's files, but not vice versa.

# Part IV. Official Levels of Computer Security

The US Department of Defence (DoD) published a document called Orange Book. This document divides the operating system in to 7 categories based on their security properties. A table of Orange Book requirements are shown below:

| Criterion | D | C1 | C2 | B1 | B2 | B3 | A1 |
|---|---|---|---|---|---|---|---|
| **Security Policy** | | | | | | | |
| Discretionary Access Control | | X | X | → | → | X | → |
| Object Reuse | | | X | → | → | → | → |
| Labels | | | | X | X | → | → |
| Label Integrity | | | | X | → | → | → |
| Exportation of labeled information | | | | X | → | → | → |
| Labeling human readable output | | | | X | → | → | → |
| Mandatory access control | | | | X | X | → | → |
| Subject sensitivity labels | | | | | X | → | → |
| Device labels | | | | | X | → | → |
| **Accountability** | | | | | | | |
| Identification and authentication | | X | X | X | → | → | → |
| Audit | | | X | X | X | X | → |
| Trusted path | | | | | X | X | → |
| **Assurance** | | | | | | | |
| System architecture | | X | X | X | X | X | → |
| System integrity | | X | → | → | → | → | → |
| Security testing | | X | X | X | X | X | X |
| Design specification and verification | | | | X | X | X | X |
| Covert channel analysis | | | | | X | X | X |
| Trusted facility management | | | | | X | X | → |
| Configuration management | | | | | X | → | X |
| Trusted recovery | | | | | | X | → |
| Trusted distribution | | | | | | | X |
| **Documentation** | | | | | | | |
| Security features user's guide | | X | → | → | → | → | → |
| Trusted facility manual | | X | X | X | X | X | → |
| Tested documentation | | X | → | → | X | → | X |
| Design documentation | | X | → | X | X | X | X |

Symbol X

means that there are new requirements here.

Symbol ⟶

means that requirements from the next lower category also apply here.

## Level D security

It has no security requirements at all. It is very easy to achieve.

MS-DOS, Windows 95/98 are at level D.

## Level C security

This is for environments in which there are cooperating users.

Level C allows individual users to control the access to their files. They need only discretionary access control (DAC).

Example: Windows NT, Modern Unix systems.

C1 level needs

a protected mode operating system,

authenticated user login system,

group names can be used for login,

discretionary access control.

Example operating system: Early versions of Unix

C2 level needs

it allows only individual login,

does not allow logging using a group name,

security audits,

authorised users are allowed to access certain files and programs,

discretionary access control available for users.

Unix access control scheme (rwx) obeys level C1, but not level C2.

## Level B and Level A security

These levels need a security label for all users and objects.

The security labels can be,

unclassified,

secret, or

top secret.

The system must be capable of using the Bell-La Padula model.

### Level B Security

It requires mandatory access control (MAC)

This means Operating systems needs a predefined central permission scheme to determine the permissions assigned to users.

The creator of a file cannot control permissions of that file.

Level B1

Level B1 needs a central permission scheme and operating system should be able to apply “confidentia labels” on users and files and other devices.

The access control mechanism uses these “confidential” labels to dertermine permissions.

Examples: HP-UX BLS, SEVMS, CS/SX

Level B2

In level B2, system must be designed in a top down way.

It includes all the requirements of B1.

Also the connection line for authentication between the user and operating system must be secure.

Examples: Muiltics, VSLAN

Level B3

It includes all the requirements of B2.

In level B3, there must be access control lists (ACLs),

security auditing must be there,

security crash recovery should be included.

**Level A security**

Level A1

It requires all the features of B3.

Examples: Boeing MLS LAN

Level A1 requires a formal model of the security system and a proof that the model is correct.

Level A2

A2 requirements are reserved for future use.

# Part V. Concept of Holes

A hole is a feature of an operating system or a software that allows unauthorised users to gain access or increase their level of access without authorisation.

Many peculiarities of software commonly known to all users can qualify as holes. One such peculiarity is that CMOS password on PCs are lost when the CMOS battery is shorted, disabled or removed.

A hole is some form of vulnerability. Every platform has holes, whether it is Windows or Unix. No computer system or network is completely safe.

## 9 Types of Holes

There are different types of holes. Some of them are,

holes that allow denial of service,

holes that allow local users with limited privileges to increase their privileges without authorization,

holes that allow outside parties (remote computers) unauthorised access to the network.

## Holes that allow denial of service

Holes of this type are of lower priority. These attacks are always operating system based. These holes exist within the networking portion of the operating system. For such holes, operating system authors correct them by adding patches.

In denial of service attack, an attacker exploits the features of Internet Protocol (IP).

One example, is TCP SYN attack, in which a connection requests are sent to a server in high volume and server is overwhelmed with requests. This results in a slow server.

In another attack, syn-flooder attack creates a large number of half open connections.

This is a problem inherent in the design of the TCP/IP protocol suite. Thsi hole exists within the hearts of the networking services of the Unix operating system.

## Holes that allow local users unauthorised access

Some holes allow local users increased and unauthorised access.

### sendmail problem

Sendmail is a tool to send and receive emails in Unix. Sendmail problem is well known. Sendmail is the most popular method of transmitting email and is the heart of the Internet's email system.

This sendmail program is normally always active in a Unix machine. When active, sendmail program listens (port 25) for email deliveries and other requests.

When sendmail gets started, it determines the identity of the user who started it. Only the administrator (root) user is allowed to start sendmail. Due to a coding error, a user can invoke sendmail in such a way that it bypasses this check.

As a result, any user can start sendmail. By manipulating sendmail's environment, a user can have sendmail execute an arbitrary program.

Thus here an ordinary user has got some form of administrator privileges.

Also some versions of sendmail has weaknesses in the buffer so that a user can overflow the buffer.

These types of holes are serious because administraor may never discover the unauthorised accesses by local users.

## Other Classes of Holes

Most holes occur from some defect within an application. Some common programming errors lead to such holes.

One error is the character buffer in programs written in C.

For example, consider the C program:

```
#include<string.h>
main()
{
    char p[5];
    strcpy(p,"welcome");
}
```

Here array p[] is of size only 5. We stored a string having 7 characters in this array. This is called buffer overflow problem in C.

When th buffer is too small, this can happen. when this overflow occurs, the rest of the characters are stored somewhere in memory.

Crackers by manipulating where these extra characters end up, can cause arbitrary commands to be executed by the operating system.

Commonly this mechanism is used by local users to gain access to root shell.

## Holes that allow remote users unauthorised access

It is the most dangerous hole.

Many webserver software contains a lot of documentation such as,

installation instructions,

binaries,

source code,

sample configuration files, and

sample CGI scripts.

These scripts can sometimes provide an intruder with root access.

A famous hole is the vulnerability in a file called 'test.cgi' came with early versions of Apache web server. This file contained a flaw that allowed intruders to read files from CGI directory.

The holes on platforms other than Unix take more time to solve.

Different holes have been detected in the following famous softwares,

ftp,

gopher,

telnet,

sendmail,

nfs,

ARP,

portmap,

finger.

Any hole that a hacker or cracker can exploit will lead to other flaws.

CERT (Computer emergency Response Team),

issues advisories to the Internet users whenever new security holes arise.

CIAC (Department of Energy Computer Incident Advisor Capability),

NISTCSRC (National Institute of Standards and Technology computer Security Resource Clearing House),

FIRST (Forum of Incident Response and Security Teams)

dela with preventing security holes.

# Part VI. Security Features in Unix Operating System

Unix is the dominant operating system in high end workstations and servers.

## Fundamental Concepts

### Owner, Group, Others

A Unix system can have a number of users. For example, root, arun, anil, sj07btcs01, sj07btcs03 etc.. Each user has a unique userid such as 100, 1200 etc..

Let the userid of sj07btcs03 is 250.

A file such as a.txt, p1.c, k.pdf, different processes and other resources have an owner associated with them. Owner is a user of the system.

For example, let the owner of a.txt in /opt is the user 'sj07btcs05'. Let the owner of k.pdf in /var is the user 'root'.

Every resource has an owner.

Users are organised in to groups. A group has a group id associated with it.

For example, sj07btcs03, sj07btcs05, arun can form a group with the name sjcet. Let this group sjcet has group id 750.

sj07btcs02, sj07btcs06, anil, sj08btcs04 can form another group with the name student. Let this group has group id 800.

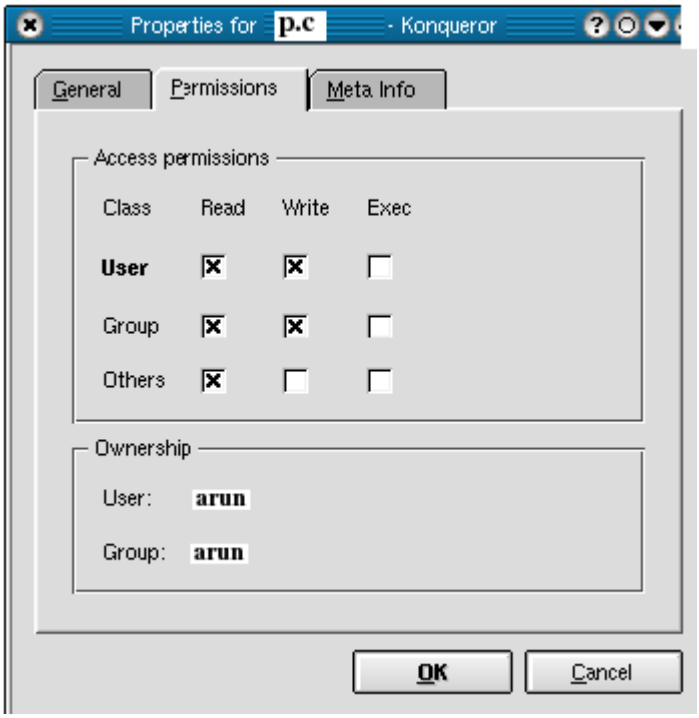
In current Unix systems, a user can be a member of many groups.

A process executing on behalf of a user carries the user id and group id of that user.

For example, a process firefox executing on behalf of the user sj07btcs03 carries the userid 250 and group id 750.

When a user 'arun' (userid=20, group id=85) creates a file p.c, arun will become the owner of the file p.c and the file carries the userid 20 and group id 85.

Files also get permissions. These permissions which specify what access the owner, owner's group and rest of the users have to the file. Potential access permissions are read, write and execute.




**Superuser (root)**

The user with userid 0 is called superuser (root). Root has the power to read and write all files in the system.

## 9.1 Security Features for Authentication in Unix

In a Unix system, a password file is present. (/etc/passwd). It is publicly readable. When a user logs in, the login program asks for a user name and password.

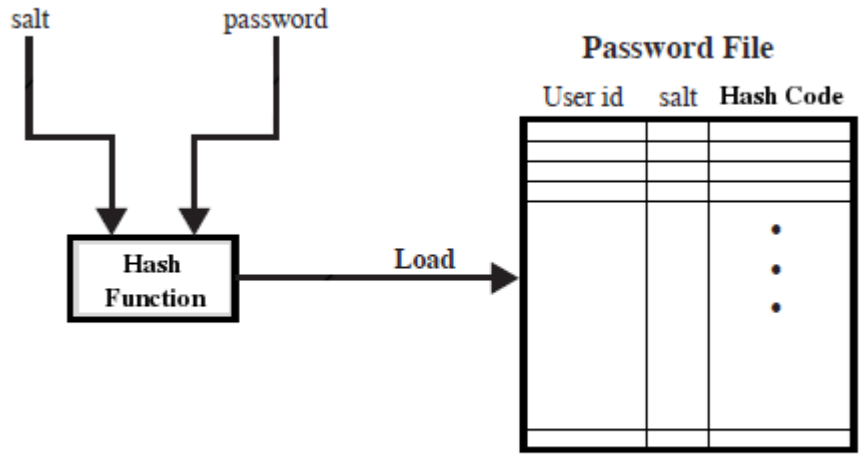
Unix system uses the technique of hashed passwords and salt values.

When a user is created, user types his password. This password is combined with a salt value. Salt value is a random number.

The password and salt value are given as input to a hashing algorithm and it produces a hash code. The hashed password is stored together with the salt in the password file.

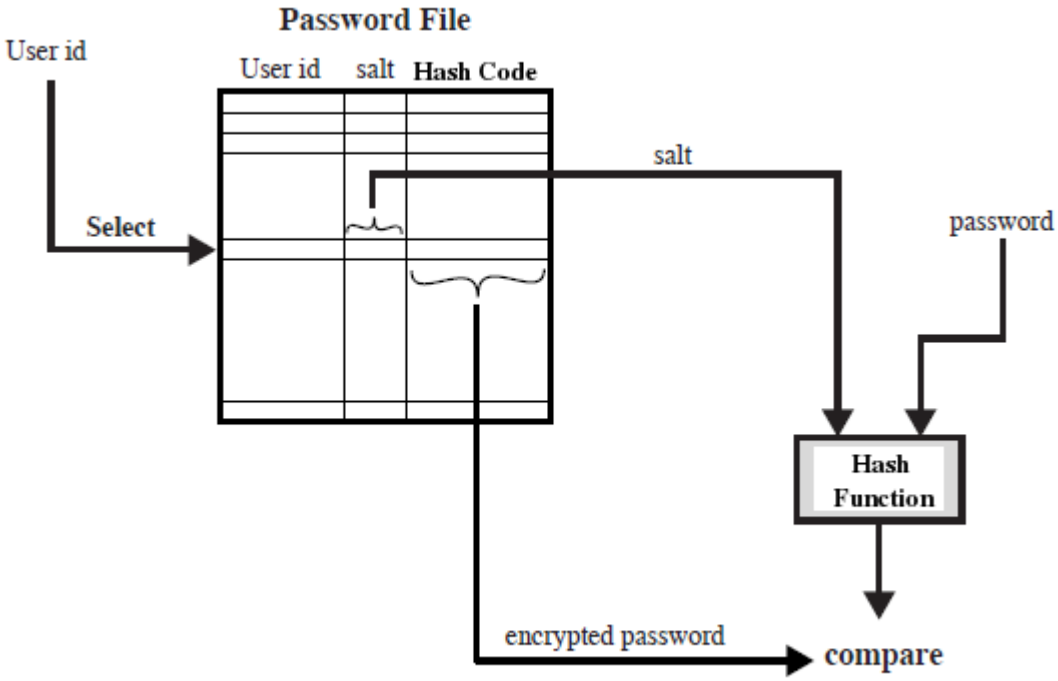
The hashing algorithm is either MD5 or DES.

Admin user can modify the choice of hashing algorithm.




Loading a new password

When a user types his password for authentication, the password is recombined with the salt value stored in the password file and passed through the same hash function. If the result matches with the entry in the password file, password is accepted.




Verifying a password

In current Unix systems, passwords are stored in a file which is not publicly readable (/etc/shadow). Longer passwords are allowed. There should be a minimum of 6 letters present in a password.

Pluggable Authentication Modules (PAM) system is used nowadays for authentication. It is based on a shared library that can be used to authenticate users. PAM modules can specify authentication methods, account restrictions, session-set up functions or password changing functions. PAM supports smart card, kerberos and voice authentication systems.

The salt is used to prevent duplicate passwords from being visible in the password file. If two users choose the same password, these passwords will be assigned different salt values.

The use of salt increases the difficulty of dictionary attacks.

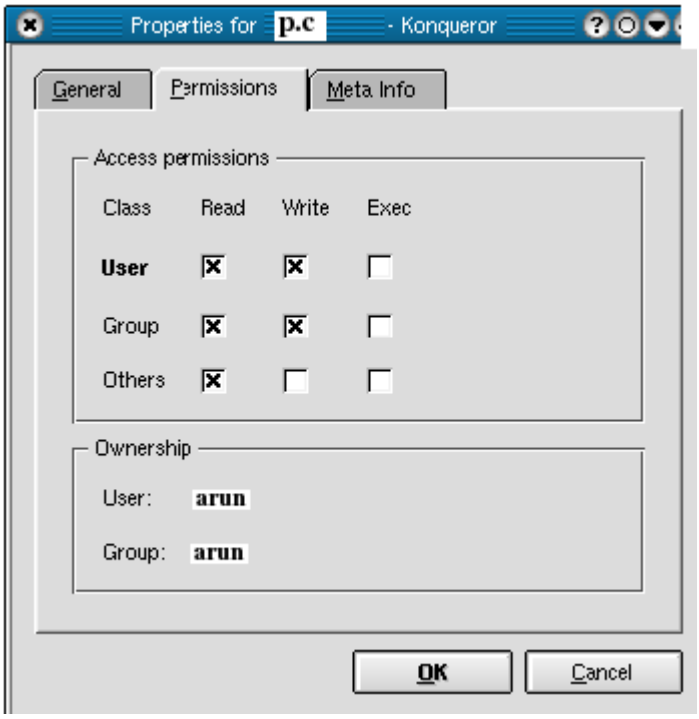
## 9.2 Access Control in Unix

As we learned, users are assigned unique identifier values (uid). A number of users can form a group and is assigned a groupid (gid).

Access control is applied to various objects in the system such as files. An object in the system has an owner. Owner is a user in the system such as root, sj08btcs01 etc..

Every object has a single uid and single gid associated with it.

Unix performs access control by assigning objects a protection mask that specifies access modes- read, write , execute. Users of an object are grouped into owner, owner's group, others.




The super user (root) has automatic access to any object in the system; bypassing normal access checks. They can perform reading any physical memory and opening network sockets. This allows kernel to prevent normal users from accessing these resources.

## 9.3 Remote Execution in Unix

Earlier, less secure programs such as 'rsh' and 'rexec' were used in Unix for remote execution. They sent information such as passwords in plain text.

SSH (Secure Shell) is a new protocol used in Unix for secure data connections. Using it, logging on to a remote machine and commands can be executed on that machine.

SSH program provides users with features such as TCP port and X Window System forwarding, facilities for copying files back and forth, cryptographic user authentication, integration with network file systems, transfer of user credentials across machines, pseudo-terminals and more.

SSH uses encryption for authentication. It uses public key cryptography to authenticate the remote computer and allow it to authenticate the user.

The list of authorised keys is stored in the user's home directory in the file

~/.ssh/autorized-keys.

An SSH process is present in the remote system accepting remote connections.

SSH is available in all the platforms such as Unix, MacOS, Linux and Windows.

# Part VII. Security Features in Windows Operating System

## 10 Authentication in Windows

Users have user names and passwords.

If a user wants to access a local system, operating system uses the Security Account Manager (SAM) database.

#### Domain Controller

A designated computer in the network handles remote authentication. The computer is called domain controller. It uses Kerberos protocol for network authentication.

The logon program passes the username and password to the Winlogon program, whic hpasses the credentials to the Local Security Authority (LSA). LSA is responsible for handling all authentication and authorization for users.

When the user logs onto a local machine, LSA verifies the user's credentials with the local SAM database.

If it is a network, LSA passes the credentials over the network to the domain controller via the security support provider interface (SSPI). The domain controller authenticates using Kerberos.

## 11 Access Control in Windows

#### Security Principal

A user, group, a process or a computer that performs an action in Windows is called a security principal. A number of users can form a security group. Security groups can be nested to form larger security groups.

A security principal is given a unique id calledd Security Identifier (SID). When a user logs on, the operating system uses its SID to assign an access token.

Access token stores security information about the security principal, its SID, SID of all groups in which it is a member of and the access control list.

When the security principal runs a process, operating system assigns that process a copy of security principal's access token.

These access tokens are used to implement fast user switching. In this, a new user can log on, without logging off the current user. As a result, first user's processes may be executing in the background. An access token stores a session id which is unique for each login session.

#### Security Descriptor

Every resource has a security descriptor. It contains its security information.

A security descriptor contains

the resource owner's SID, and

a discretionary access control list (DACL) which determines which security principal may access the resource. DACL is an ordered list of access control entities (ACEs). Each ACE contains the SID of the security principal and the type of access the security principal to that resource.

**Security Groups**

Windows defines three basic security groups:

Everyone (all authenticated security principals),

Anonymous (all unauthenticated security principals), and

guest ( a general pupose account).

**Questions**

MGU/May2012

1. What are the different authentication mechanisms adopted for OS security (4marks)?

2. Illustrate the different official levels of computer security (4marks).

3a. Discuss the need for OS security. Describe the protection mechanisms adopted for OS security.

OR

b. Discuss the security features for authentication, access control and remote execution in Windows 2000 (12marks).

MGU/May2010

1. Name the different protection mechanisms of OS (4marks).

2. What are the different types of holes (4marks)?

3a. Discuss in detail about discretionary and mandatory access control.

OR

b. Explain about the access control and remote execution in UNIX, Windows 2000 (12marks).

MGU/Nov2009

1. What are security breaches (4marks)?

2. What is a hole? Explain (4marks).

3a. Give in detail the security features of access control in Unix.

OR

b. Explain any two protection mechanisms available in OS security (12marks).

MGU/June 2009

1. Explain different official levels of computer security (4marks).

2. Explain types of holes (4marks).

3a. Differentiate between discretionary and mandatory access control.

OR

b. Explain security features for authentication (12marks).

MGU/May2008

1. Define OS security (4marks).

2. Explain access control in Unix (4marks).

3a. Explain the security features for access control.

OR

b. Explain authentication and accesss control in computer security (12marks).

MGU/Nov2008

1. Define OS security (4marks).

2. Explain access control in Unix (4marks).

3a. Explain authentication and access control in computer security.

OR

b. Discuss the security features for authentication (12marks).

MGU/July2007

1. Explain access control in Unix (4marks).

2. Define OS security (4marks).

3a. Explain authentication and access control in computer security.

OR

b. Explain the concept of the security features for authentication (12marks).

MGU/Jan2007

1. Explain access control (4marks).

2. What are the various types of holes in OS (4marks)?

3a. Explai nthe authentication and access control mechanisms in OS.

Or

b. Briefly explain Windows 2000 security features (12marks).

MGU/July2006

1. What is a role in operating system (4marks)?

2. Define basic principles behind the security mechanisms in OS (4marks).

3a. Explain the security features in Unix OS.

OR

b. Explain in detail the official levels of computer security (12marks).

**References**

.

Tanenbaum, A, S (2011 ). Modern Operating Systems. PHI.

Stallings, W (2009). Operating Systems. Pearson Education.

website: http://sites.google.com/site/sjcetcssz