ABDUL KALAM TECHNICAL UNIVERSITY

COMPUTER SYSTEM SECURITY
(KNC-401)

Instructor:

Neha Prakash

Assistant Professor

Computer Science & Engineering

MIET.

Subject and Code: Computer System Security (KNC-301) [L/T/P - 2/0/0]
Semester: IV
Faculty: Neha Prakash
Pre-requisite: Computer Fundamentals and some knowledge of network.

Course Outcomes (COs): Graduates would be able to

1. Identify the computer security fundamentals.
2. Identify the legal and ethical issues of information security
3. Apply the concepts of cryptography.
4. Demonstrate security control mechanism.
5. Identify network security mechanism.

Mapping of COs with POs

H=3, M=2, L=1

SEM	SUBJECT	CODE	L/T/P	CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
IV	Computer System Security	KNC-401	L	1.												
			L	2.												
			L	3.												
			L	4.												
			L	5.												

Syllabus

DETAILED SYLLABUS COMPUTER SYSTEM SECURITY		
Course Outcome (CO)		**Bloom's Knowledge Level (KL)**
At the end of course , the student will be able to understand		
CO 1	To discover software bugs that pose cyber security threats and to explain how to fix the bugs to mitigate such threats	**K1, K2**
CO 2	To discover cyber attack scenarios to web browsers and web servers and to explain how to mitigate such threats	**K2**
CO 3	To discover and explain mobile software bugs posing cyber security threats, explain and recreate exploits, and to explain mitigation techniques.	**K3**
CO 4	To articulate the urgent need for cyber security in critical computer systems, networks, and world wide web, and to explain various threat scenarios	**K4**
CO 5	To articulate the well known cyber attack incidents, explain the attack scenarios, and explain mitigation techniques.	**K5, K6**
DETAILED SYLLABUS	**3-1-0**	

Unit	Topic	Proposed Lecture
I	**Computer System Security Introduction**: Introduction, What is computer security and what to l earn? , Sample Attacks, The Marketplace for vulnerabilities, Error 404 Hacking digital India part 1 chase. **Hijacking & Defense:** Control Hijacking ,More Control Hijacking attacks integer overflow ,More Control Hijacking attacks format string vulnerabilities, Defense against Control Hijacking - Platform Defenses, Defense against Control Hijacking - Run-time Defenses, Advanced Control Hijacking attacks.	08
II	**Confidentiality Policies:** Confinement Principle ,Detour Unix user IDs process IDs and privileges , More on confinement techniques ,System call interposition ,Error 404 digital Hacking in India part 2 chase , VM based isolation ,Confinement principle ,Software fault isolation , Rootkits ,Intrusion Detection Systems	08
III	**Secure architecture principles isolation and leas:** Access Control Concepts, Unix and windows access control summary, Other issues in access control, Introduction to browser isolation. **Web security landscape :** Web security definitions goals and threat models , HTTP content rendering .Browser isolation .Security interface , Cookies frames and frame busting, Major web server threats ,Cross site request forgery ,Cross site scripting ,Defences and protections against XSS , Finding vulnerabilities ,Secure development.	08
IV	**Basic cryptography:** Public key cryptography ,RSA public key crypto ,Digital signature Hash functions ,Public key distribution ,Real world protocols ,Basic terminologies ,Email security certificates ,Transport Layer security TLS ,IP security , DNS security.	08
V	**Internet Infrastructure:** Basic security problems, Routing security, DNS revisited, Summary of weaknesses of internet security, Link layer connectivity and TCP IP connectivity, Packet filtering firewall, Intrusion detection.	08

Text books:

1. William Stallings, Network Security Essentials: Applications and Standards, Prentice Hall, 4th edition, 2010.
2. Michael T. Goodrich and Roberto Tamassia, Introduction to Computer Security, Addison Wesley, 2011.
3. Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, Handbook of Applied Cryptography, CRC Press, 2001.

Mapped With : https://ict.iitk.ac.in/product/computer-system-security/

UNIT 2
Computer System Security

Key points: Confidentiality, confinement principles, Detour (Unix User ID etc) , More on confinement techniques, system call interposition, IDS, Rootkit

Confidentiality Policies

A confidentiality policy is intended to protect secrets; specifically, it is intended to prevent unauthorized disclosure of information.
As we know the goals of security. (Confidentiality, Integrity and availability)
Like : Your personal information should not be breached as your medical information, financial information etc.

Goals of confidentiality Policies:

i)- Confidentiality Policies emphasize the protection of confidentiality.

ii)- Confidentiality policy also called information flow policy, prevents unauthorized disclosure of information.

iii)- Example: Privacy Act requires that certain personal data be kept confidential. E.g., income tax return info only available to IT department and legal authority with court order. It limits the distribution of documents/info.
Or HIPAA (Health Insurance Portability and Accountability Act) Health Insurance Portability and Accountability Act. abbreviation. (Insurance: Medical insurance) In the U.S., HIPAA is an act that protects people covered by health insurance and makes rules about storing personal medical data.

So this flow can be guided by two access control:

1)- Discretionary Access Control(DAC)
2)- Mandatory Access Control(MAC)

Discretionary Access Control (DAC)

•Mechanism where a user can set access control to allow or deny access to an object (System does not do anything)

•Also called Identity-based access control (IBAC) (Who is authorized to access the information using passwords, Pin number etc)

•It is a traditional access control techniques implemented by traditional operating system such as Unix, Windows etc.

A typical example of DAC is Unix file mode, which defines the read, write and execute permissions in each of the three bits for each user, group and others.

- –Based on user identity and ownership
- –Programs run by a user inherit all privileges granted to the user.
- –Program is free to change access to the user's objects
- –User may transfer object ownership to another user(s).
- –User may determine the access type of other users.
- –After several attempts, authorization failures restrict user access.
- –Support only two major categories of users:
  - •Completely trusted admins
  - •Completely untrusted ordinary users

Problems with DAC

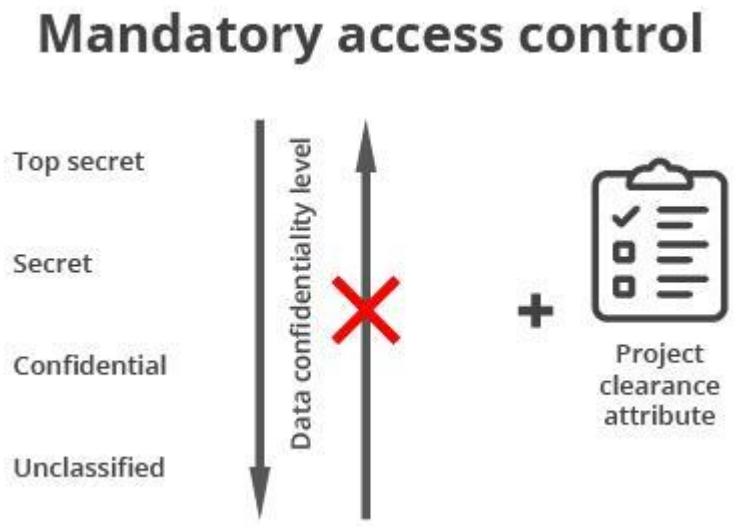
- ACL(Access control list) maintenance

- Grant and revoke permissions maintenance
- Only support coarse-grained privileges
  –Too simple classification of users (How about more than two categories of users?)

Mandatory Access Control:

- Mandatory access control (MAC) is a security strategy that restricts the ability individual resource owners have to grant or deny access to resource objects in a file system.
- MAC criteria are defined by the system administrator, strictly enforced by the operating system (OS) or security kernel, and are unable to be altered by end users.

Often employed in government and military facilities, mandatory access control works by assigning a classification label to each file system object. Classifications include confidential, secret and top secret.



With MAC, the process of gaining access looks like this:

- The administrator configures access policies and defines security attributes: confidentiality levels, clearances for accessing different projects and types of resources.
- The administrator assigns each subject (user or resource that accesses data) and object (file, database, port, etc.) a set of attributes.
- When a subject attempts to access an object, the operating system examines the subject's security attributes and decides whether access can be granted.

let's consider data that has the "top secret" confidentiality level and "engineering project" security label. It's available to a set of users that have "top secret" clearance and authorization to access engineering documents. Such users can also access information that requires a lower level of

clearance. But employees with lower levels of clearance will not have access to information that requires a higher level of clearance.

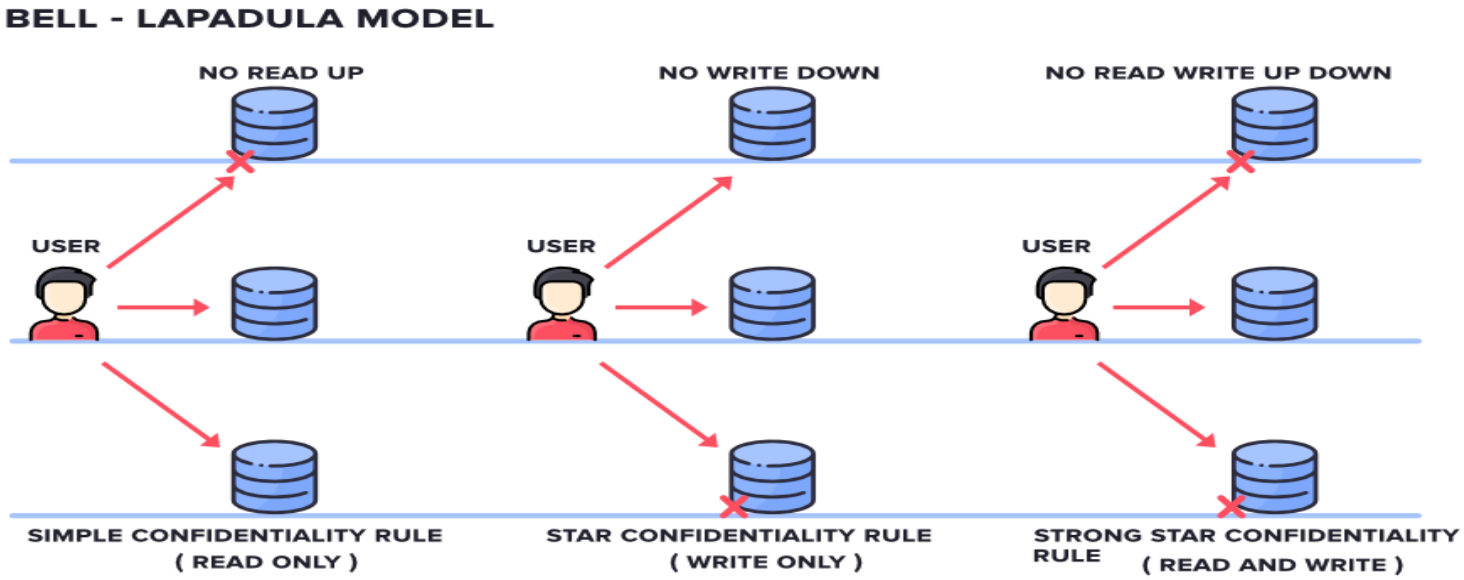
DAC vs MAC

DAC	MAC
DAC stands for Discretionary Access Control.	MAC stands for Mandatory Access Control.
DAC is easier to implement.	MAC is difficult to implement.
DAC is less secure to use.	MAC is more secure to use.
In DAC, the owner can determine the access and privileges and can restrict the resources based on the identity of the users.	In MAC, the system only determines the access and the resources will be restricted based on the clearance of the subjects.
DAC has extra labor-intensive properties.	MAC has no labor-intensive property.
Users will be provided access based on their identity and not using levels.	Users will be restricted based on their power and level of hierarchy.
DAC has high flexibility with no rules and regulations.	MAC is not flexible as it contains lots of strict rules and regulations.
DAC has complete trust in users.	MAC has trust only in administrators.
Decisions will be based only on user ID and ownership.	Decisions will be based on objects and tasks, and they can have their own ids.

Confidentiality Model:

Bell-LaPadula

This Model was invented by Scientists David Elliot Bell and Leonard .J. LaPadula.Thus this model is called the Bell-LaPadula Model. This is used to maintain the Confidentiality of Security. Here, the classification of Subjects(Users) and Objects(Files) are organized in a non-discretionary fashion, with respect to different layers of secrecy.



It has mainly 3 Rules:

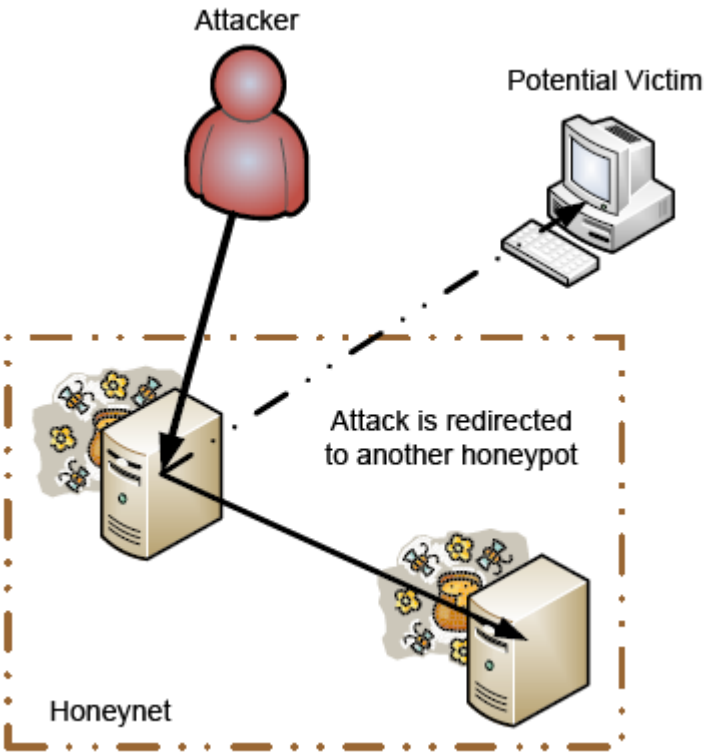
- SIMPLE CONFIDENTIALITY RULE: Simple Confidentiality Rule states that the Subject can only Read the files on the Same Layer of Secrecy and the Lower Layer of Secrecy but not the Upper Layer of Secrecy, due to which we call this rule as NO READ-UP
- STAR CONFIDENTIALITY RULE: Star Confidentiality Rule states that the Subject can only Write the files on the Same Layer of Secrecy and the Upper Layer of Secrecy but not the Lower Layer of Secrecy, due to which we call this rule as NO WRITE-DOWN
- STRONG STAR CONFIDENTIALITY RULE: Strong Star Confidentiality Rule is highly secured and strongest which states that the Subject can Read and Write the files on the Same Layer of Secrecy only and not the Upper Layer of Secrecy or the Lower Layer of Secrecy, due to which we call this rule as NO READ WRITE UP DOWN

Confinement Principles:

Running untrusted code
We often need to run buggy/untrusted code:
–programs from untrusted Internet sites:
•apps, extensions, plug-ins, codecs for media player

–exposed applications: pdf viewers, outlook

Honeypot is a network-attached system used as **a trap for cyber-attackers** to detect and study the tricks and types of attacks used by hackers. It acts as a potential target on the internet and informs the defenders about any unauthorized attempt to the information system.

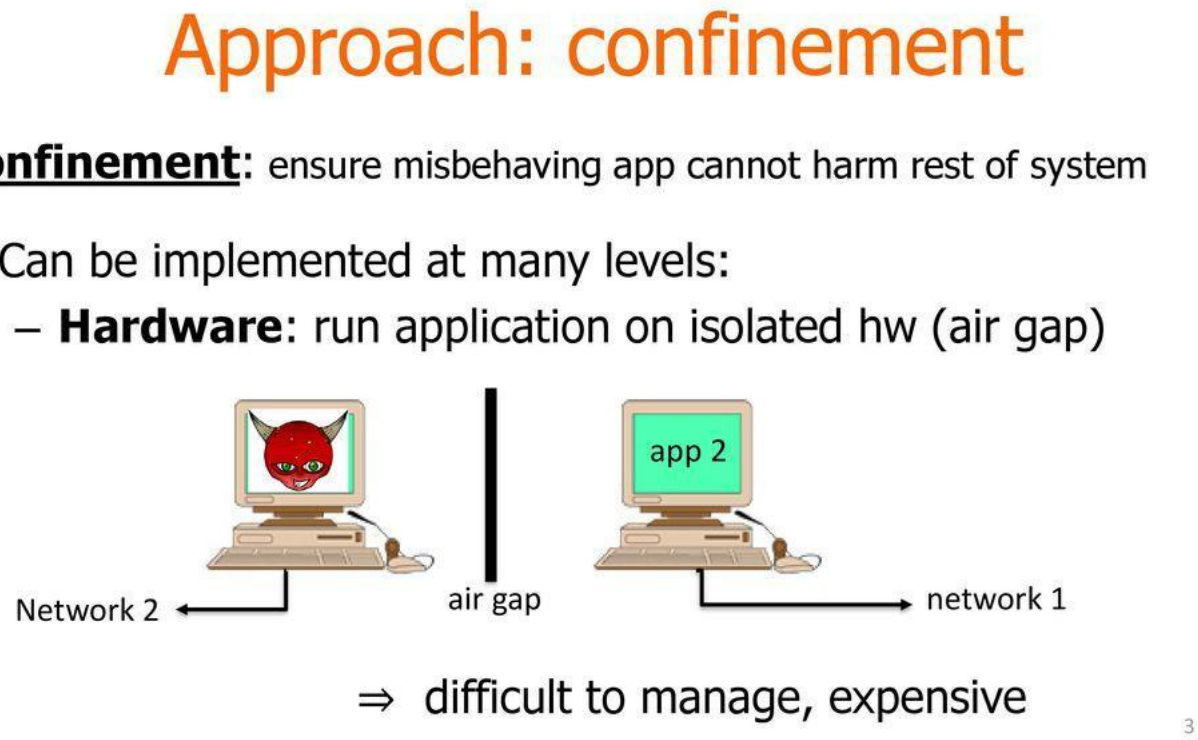


Confinement:
Ensure misbehaving application cannot harm the rest of the system.
If any application showing malicious activity kill it so that it cannot harm the rest of the system.

Confinement can be implemented at many levels:

1. HARDWARE

In hardware confinement, the hardware is isolated. Applications are run on isolated hardware. That is to run each program or application on separate systems.



The application running on network 1 and application 2 running on network 2 are separated by an air gap, so the chances of the intermingling of malware will be minimal.

NOTE: It is difficult to manage.

2.VIRTUAL MACHINES:

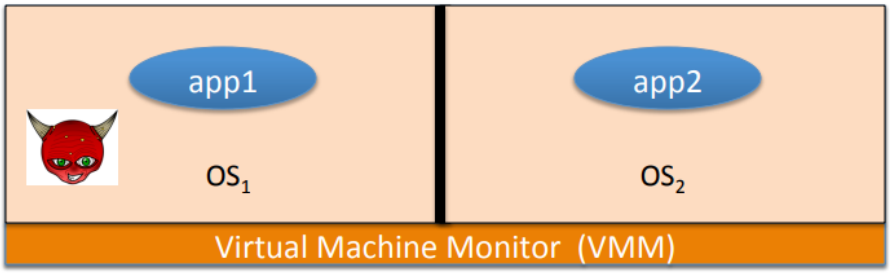
In virtual machine confinement, the operating system is isolated on a single machine.

Approach: confinement

Confinement: ensure misbehaving app cannot harm rest of system

- Can be implemented at many levels:
  - **Virtual machines**: isolate OS's on a single machine

What are some of the drawbacks of this approach?



4

Purpose of operating system isolation:
All running in different address spaces, to ensure correct operation, security and protection.
The word processor cannot access the memory of the browser or database.

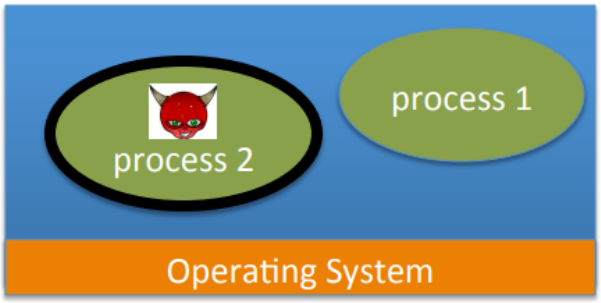
3. PROCESS:
In process confinement, a process is isolated in a single operating system by a system call interposition.

Approach: confinement

Confinement: ensure misbehaving app cannot harm rest of system

- Can be implemented at many levels:
  - **Process:** System Call Interposition

    Isolate a process in a single operating system



Each process has a wrapper around it.
Whenever the process tries for a system call, the wrapper is intercepted and only some portion is taken into for a system call to go through.

4. THREADS

In thread confinement, isolating threads sharing the same address space.
Each process has a thread, and threads share the same address space. If one thread is malicious and not to infect other threads in the same address space, the idea is to isolate the same address space shared by different threads.
This is done by software fault isolation(SFI).

Detour into Unix User IDs and IDs of Unix Processes:

1. Detour is defined as few words about Unix user IDs and IDs associated with Unix processes.
2. Everything in Unix like operating system is identified by different integer number, this unique number is called as User ID.

3. Each user account has a unique UID. The UID 0 means the super user (System admin). A user account belongs to multiple groups.

The root account is the special user in the /etc/passwd file with the user ID (UID) of 0 and is commonly given the user name, root. It is not the user name that makes the root account so special, but the UID value of 0. This means that any user that has a UID of 0 also has the same privileges as the root user.

4. There are three types of UID defined for a process, which can be dynamically changed as per the privilege of task.

The three different types of UIDS defined are :
À). Real User ID : It is account of owner of this process. It defines which files that this process has access to.
B). Effective User ID : It is normally same as real User ID, but sometimes it is changed to enable a non-privileged user to access files that can only be accessed by root.
C). Saved User ID: It is used when a process is running with elevated privileges (generally root) needs to do some under-privileged work; this can be achieved by temporarily switching to non-privileged account.

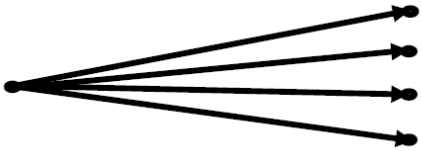
Principals and subjects/Objects:

A subject is a program (application or processes) executing on behalf of some principal(s). it is associated with uid/gid pairs. It sometimes it can object with operations:
Example: Kill, Suspend, Resume.

A principal may at any time be idle, or have one or more subjects executing on its behalf. (password, Pin, tokens, ID etc).

An object is anything on which a subject can perform operations usually objects are passive, for example:
a. File
b. Directory (or folder)
c. Memory segment.

USERS AND PRINCIPALS



USERS

Real World User

PRINCIPALS

Unit of Access Control and Authorization

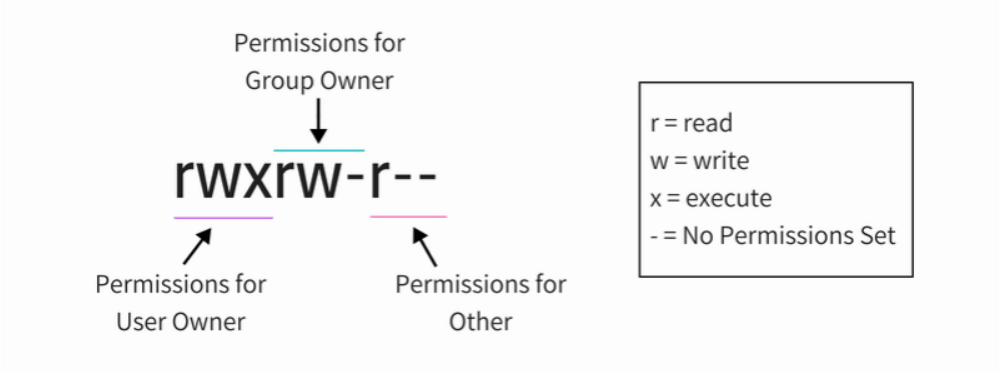
the system authenticates the human user to a particular principal

There should be a one-to-many mapping from users to principals
•a user may have many principals, but

•each principal is associated with a unique user

Organization of Objects

- •Almost all objects are modeled as files
  - –Files are arranged in a hierarchy
  - –Files exist in directories
  - –Directories are also one kind of files
- •Each object has
  - –owner
  - –group
  - –12 permission bits
    - •rwx for owner, rwx for group, and rwx for others



- •suid, sgid, sticky

Set User ID (setuid) :

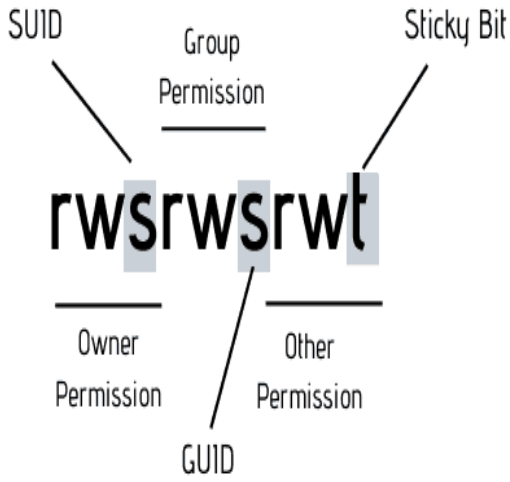
If a file is owned by the root user and has the setuid bit set, no matter who executed the file it would always run with root user privileges.

Set GroupID (setgid):

If the setgid bit is set on a directory, all files created within said directory inherit the group ownership of that directory.

Sticky Bit:

If the sticky bit is set on a directory then all files created within said directory can only be removed by its owner , or the root user.



Let's take a practical example. If you look at the binary executable file of the passwd command, it has the SUID bit set.

```
linuxhandbook:~$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 59640 Mar 22  2019 /usr/bin/passwd
```

This means that any user running the passwd command will be running it with the same permission as root.

System Call Interposition:

A system call is the programmatic way in which a computer program requests a service from the kernel of the operating system it is executed on. A system call is a way for programs to interact with the operating system.

Type of System Calls: There are 5 different categories of system calls –

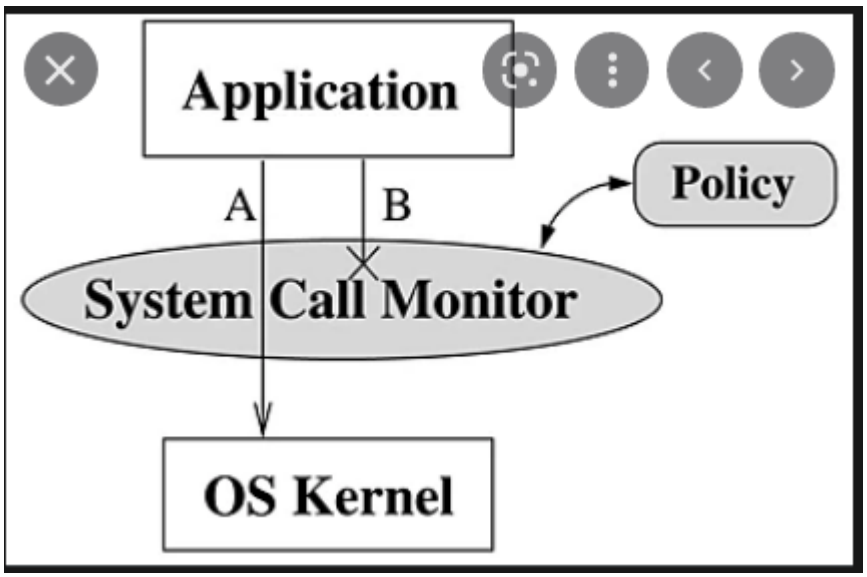
1. Process control: end, abort, create, terminate, allocate and free memory.
2. File management: create, open, close, delete, read file etc.
3. Device management
4. Information maintenance
5. Communication

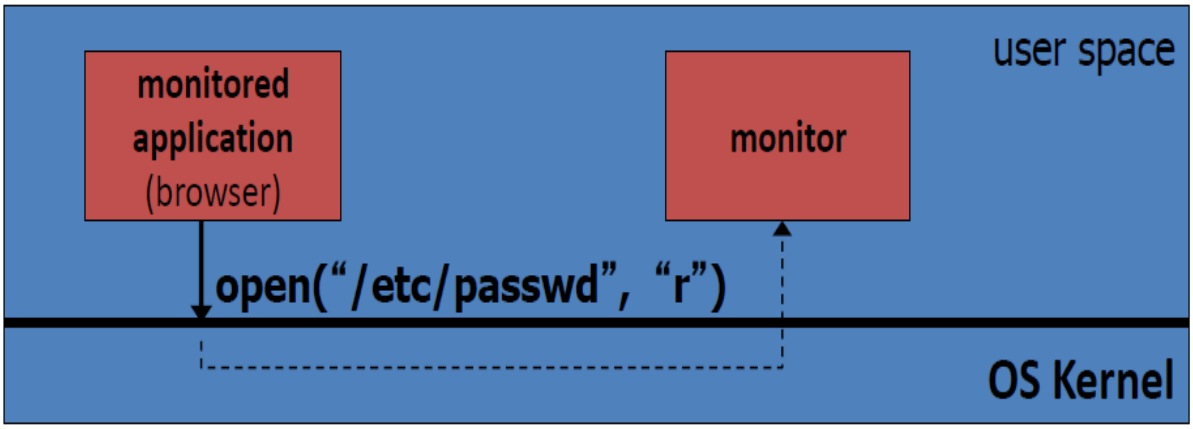
Examples of Windows and Unix System Calls –

	Windows	Unix
Process Control	CreateProcess()	fork()
	ExitProcess()	exit()
	WaitForSingleObject()	wait()
File Manipulation	CreateFile()	open()
	ReadFile()	read()
	WriteFile()	write()
	CloseHandle()	close()

System call interposition

System call interposition is a powerful method for regulating and monitoring program behavior. A wide variety of security tools have been developed which use this technique. A system call correlating method is proposed to identify the coherent system calls belonging to the same process from the system call sequence.





Monitor kills application if request is disallowed

System call interposition

Observation: to damage host system (e.g. persistent changes) app must make system calls:

- To delete/overwrite files: unlink, open, write
- To do network attacks: socket, bind, connect, send

Idea: monitor app's system calls and block unauthorized calls

Implementation options:

- Completely kernel space (e.g. GSWTK)
- Completely user space (e.g. program shepherding)
- Hybrid (e.g. Systrace)

Dan Boneh

I