CS 472 - PRINCIPLES OF INFORMATION SECURITY

- Overview of computer security, Security concepts, Need of Security, Access Control, Access control matrix
- Security policies
- Software vulnerabilities
- Security in current domains - Wireless LAN security, Cell phone security
- Secure Electronic transactions, Web Services security

MODULE 1

- Introduction: Overview of computer security, Security concepts, Need of Security- Threats- Deliberate software attacks, Deviation in quality of service, Attacks- malicious code, brute force, Timing attack, sniffers
- Access Control Mechanisms - Access Control, Access control matrix, Access control in OS- Discretionary and Mandatory access control, Role-based access control, case study SELinux

What is Security?

- "The quality or state of being secure—to be free from danger"
- A successful organization should have multiple layers of security in place:
  - Physical security
  - Personal security
  - Operations security
  - Communications security
  - Network security
  - Information security

What is Security? (continued)

- The protection of information and its critical elements, including systems and hardware that use, store, and transmit that information
- Necessary tools: policy, awareness, training, education, technology
- C.I.A. triangle was standard based on **confidentiality, integrity, and availability**
- C.I.A. triangle now expanded into list of critical characteristics of information

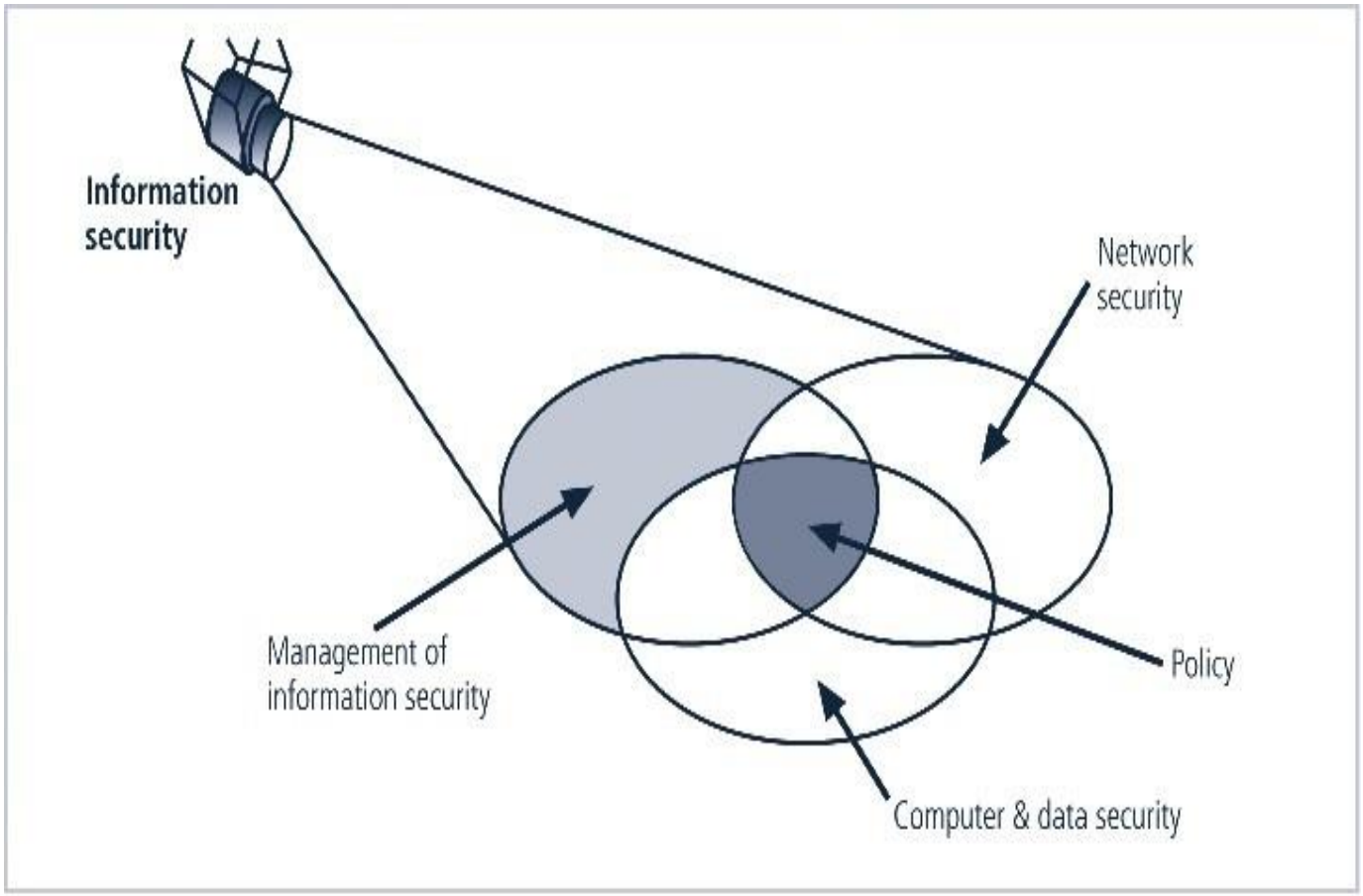


FIGURE 1-3 Components of Information Security

1.2 KEY SECURITY CONCEPTS

1. Confidentiality: Preserving authorized restrictions on information access and disclosure.

2. Integrity: Guarding against improper information modification or destruction.

3. Availability: Ensuring timely and reliable access to and use of information.

4. Authenticity: The property of being genuine and being able to be verified and trusted; confidence in the validity of a transmission, a message, or message originator.

5. Non-Repudiation: is a way to guarantee that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message.

1.3 SECURITY TERMINOLOGY

1. Adversary (threat agent) - An entity that attacks, or is a threat to, a system.

2. Attack -An assault on system security that derives from an intelligent threat; a deliberate attempt to evade security services and violate security policy of a system.

3. Countermeasure - An action, device, procedure, or technique that reduces a threat, a vulnerability, or an attack by eliminating or preventing it, by minimizing the harm it can cause.

4. Risk - An expectation of loss expressed that a particular threat will exploit a particular vulnerability with a particular harmful result.

5. **Security Policy** - A set of rules and practices that specify how a system or an organization provides security services to protect sensitive and critical system resources.

6. **Threat** - A potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm.

7. **Vulnerability** - Flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy.

Security concepts (C.I.A Triangle)

- 3 concepts
  - Confidentiality
  - Integrity
  - Availablity

Confidentiality

- Preserving authorized restrictions on information access and disclosure.
- Two related concepts
  - Data confidentiality – private or confidential information is not made available or disclose to unauthorized individuals
  - Privacy

Integrity

- Guarding against improper information modification or destruction.
- Loss of integrity is the unauthorized modification or destruction of information
- 2 related concepts
  - Data integrity
  - System integrity

Availability

- Ensuring timely and reliable access to and use of information
- A loss of availability is the destruction of access to or use of information

Need of security

Business Needs First

Information Security Important Functions

- Protect the organization's ability to function
- Enable the safe operation of applications
- Protect the data
- Safeguard technology assets : public key infrastructure (PKI),Email

Threats

- A threat is an object , person or other entity that presents an ongoing danger
- It is potential violation of security

Threats to Information Security

Categories of Threat	Examples
Compromises to intellectual property	Piracy, copyright
Software attacks	Viruses, worms, macros, DoS
Deviations in quality of service	ISP, power, WAN service issues from service providers
trespass	Unauthorized access and/or data collection
Forces of nature	Fire, flood, earthquake, lightning
Acts of human error or failure	Accidents, employee mistakes

Threats to Information Security

Categories of Threat	Examples
Information extortion	Blackmail or information disclosure
Deliberate acts of theft	Illegal confiscation of equipment or information
Missing, inadequate, or incomplete	Loss of access to information systems due to disk drive failure, without proper backup and recovery plan
Missing, inadequate, or incomplete controls	Network compromised because no firewall security controls
Sabotage or vandalism	Destruction of systems

Threats to Information Security

Categories of Threat	Examples
Theft	Illegal confiscation of equipment or information
Technical hardware failures or errors	Equipment failure
Technical software failures or errors	Bugs, code problems, unknown loopholes
Technological obsolescence	Antiquated or outdated technologies

Deliberate Software Attacks

- Malicious code
- Malicious software
- Malware : malicious software, is any program or file that is harmful to a computer user
- First business hacked out of existence
  - Denial-of-service attack
  - Cloudnine
    - British Internet service provider

Virus

- Segments of code
- Attaches itself to existing program
- Takes control of program access
- Replication

Worms

- Malicious program
- Replicates constantly
- Doesn’t require another program
- Can be initiated with or without the user download

Other Malware

- Trojan Horse
  - Hide their true nature
  - Reveal the designed behavior only when activated
- Back door or trap door
  - Allows access to system with special privileges
- Polymorphism
  - Changes it apparent shape over time
  - Makes it undetectable by techniques that look for preconfigured signatures
- Hoax
  - is a message warning ,the recipients of a non-existent computer virus threat. The message is usually a chain e-mail that tells the recipients to forward it to everyone they know.

Deviations in quality of service

- Deviations in quality of service includes situations where products or services not delivered as expected
- Internet service issues
- Communications and other service provider issues
- Power irregularities

Espionage or Trespass

- Access of protected information by unauthorized individuals
- Intelligence Gathering
  - Legal – competitive intelligence
  - Illegal – industrial espionage

Shoulder surfing

It is a type of social engineering technique used to obtain information such as personal identification numbers, passwords and other confidential data by looking over the victim's shoulder

Hackers

- Hackers are “people who use and create computer software [to] gain access to information illegally
- 2 levels
  - Experts
    - Develop software scripts
    - Develop program exploits
  - Novice
    - Script kiddie
      - Use previously written software
    - Packet monkeys
      - Use automated exploits
      - hackers sending large numbers of packets in order to disrupt system traffic.

System Rule Breakers

- Crackers
  - Individuals who crack or remove software protection designed to prevent unauthorized duplication
- Phreakers
  - Term used to describe the activity/culture of people who study, experiment with telecommunication systems, such as equipment and systems connected to public telephone networks

Attacks

An **attack vector** is a path or means by which hacker can gain access to a computer or network server in order to deliver a payload or malicious outcome..

Vector	Description
IP scan and attack	Infected system scans IP addresses and targets vulnerabilities
Web browsing	Infects web content files infectious
Virus	Infect other machines
Unprotected shares	Infects any device that is unprotected
Mass mail	e-mailing to all addresses in an address book
Simple Network Management Proto	Use common password employed in early versions of the protocol the attacking program can gain control of device

Methods of Attack

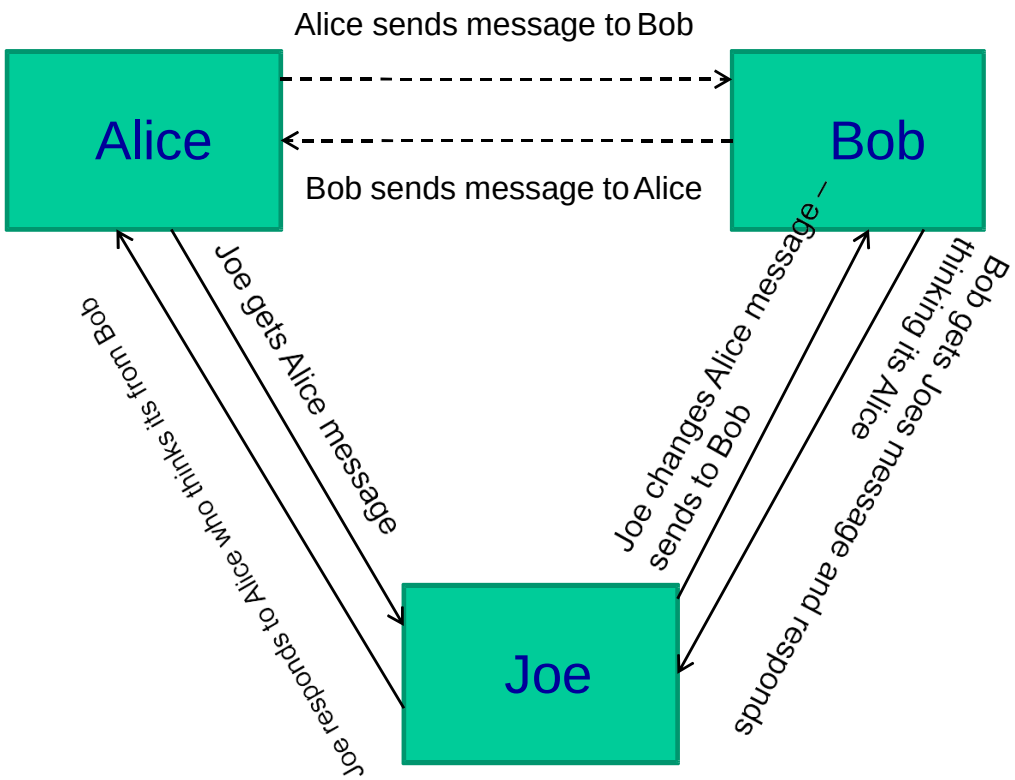
- Password Crack
- Brute force
- Dictionary

- The design of the network infrastructure and communication protocols are a major contributor

Methods of Attack

- Man-in-the-Middle
  - Monitors or sniffs packets from network
  - Modifies the packets
  - Inserts them back into the network
  - Allows attacker to eavesdrop, change, delete, reroute, add, or divert data.
  - Variant
    - Spoofing involves the interception of an encryption key exchange

Man-in -Middle



Methods of Attack

- Denial of service
  - Smurf send large amount of spoofed ping packets
  - Overwhelms the system
  - Can stop response
- Spam
- Mail bombing
- Sniffing
  - Monitors data traveling over a network
  - legitimated and non legitimate purposes
  - Packet sniffing

ACCESS CONTROL

- Access control seeks to limit the operations that an entity — user, process, etc. — Can consistent with a given security policy.
- Access control can be built into the application or it can be implemented at the system.
- Internet banking application may limit what facilities customers can avail of on the basis of customer profile.

- Access Control may be enforced at different levels of granularity.
- A database management system (DBMS), for example, may deny users access to an entire table containing sensitive information.
- The other hand, the DBMS may exercise fine-grained access control, permitting some users access to that table, but only to specific rows or columns in it.

- This topic deals primarily with access control at the Operating System (OS) level.
- Illegal accesses to memory need to be prevented. For example, a program should not be allowed to read/modify arbitrary locations in memory.
- If not, a program may get access to read and/or write the private variables of another user's program.
- At a higher level of granularity objects such as files, directories, sockets, etc. need to be protected from unauthorized access.
- Illegal accesses to memory are controlled by techniques such as paging and segmentation.

There are 3 key entities involved:

- Principal – a user or group of users (identified by user id (UID) or group id (GID))
- Subject – process or thread to whom access is granted
- Object – are resources that subjects need access to (ex . Files , printers , network sockets).
- An object may also be another process.

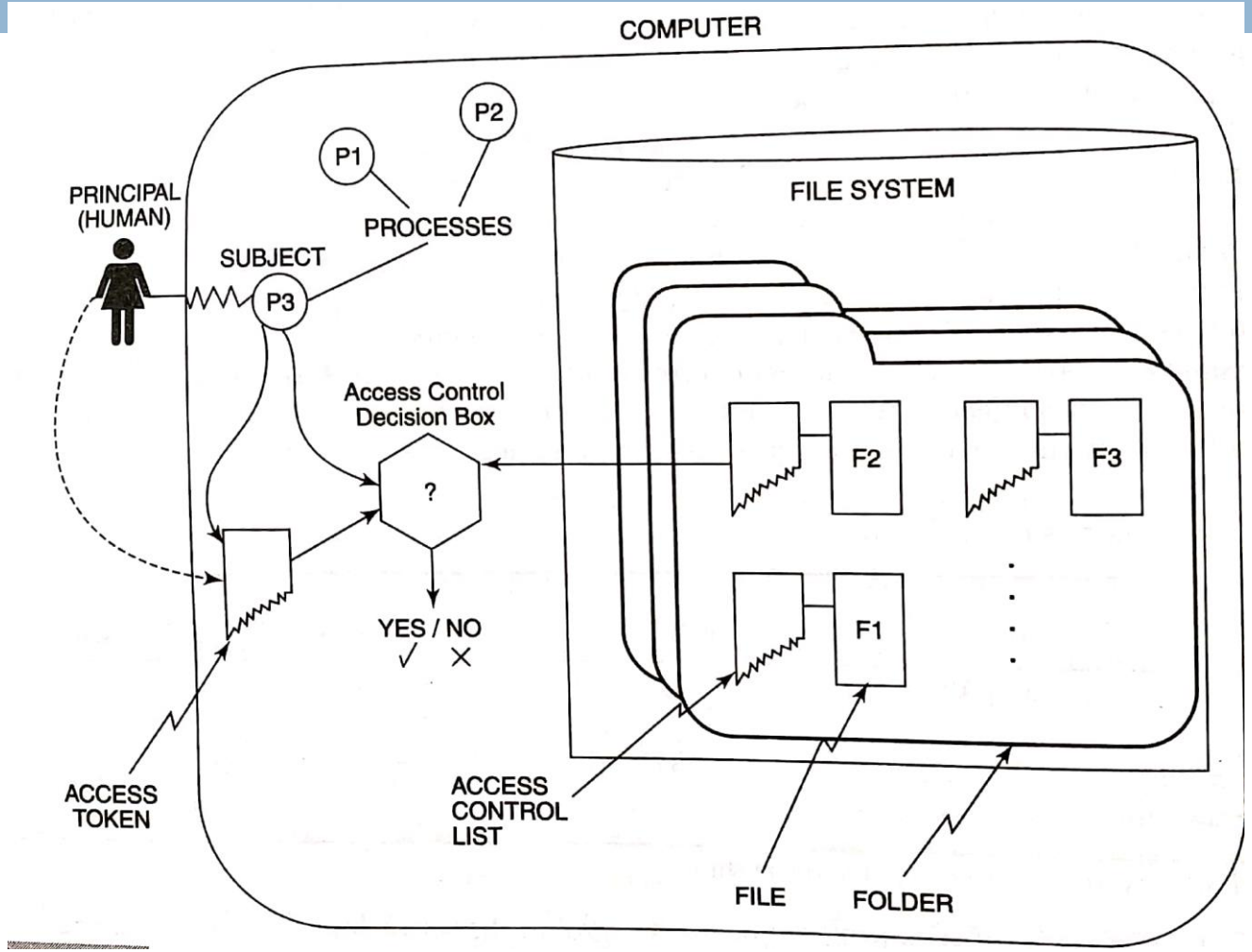
- Each object has a set of operations that may be performed on it.

  e.g. reading from a file ,listing a directory ,binding to a socket etc .

- Associated with each subject -object pair is a set of access rights or permissions to perform certain operations .
- In the context of files it includes:

  read , write , append , execute , ownership ,grant, revoke

Entities involved in access control



Access control matrix

- It is a simplest way to represent access control rights.
- Rows represents subjects and columns represents objects
- The entry in row i and column j is the set of access permissions that subject i has on object j .

User	Operating System	Accounts Program	Accounting Data	Audit Trail
Sam	rwx	rwx	r	r
Alice	rx	x	–	–
Accounts program	rx	r	rw	w
Bob	rx	r	r	r

Example access control matrix for bookkeeping.

Access Control Lists(ACL)

- Another way of simplifying access rights management is to store the access control matrix a column at a time, along with the resource to which the column refers.
- This is called an access control list, or ACL.

	OS	Accounting Program	Accounting Data	Insurance Data	Payroll Data
Bob	rx	rx	r	-	-
Alice	rx	rx	r	rw	rw
Sam	rwx	rwx	r	rw	rw
Acct. program	rx	rx	rw	rw	rw

Access Control Matrix

(Bob,-), (Alice,rw), (Sam, rw), (Accounting Program, rw).

List of permissions associated with each subject is called capability list or c-list

ACCESS CONTROL

- Access control can be handled by the system or application
- Three types of access controls
  - Discretionary Access Control
  - Mandatory Access Control
  - Role Based Access Control

DAC

- The access rights to an object is under the discretion of the owner of the object
- The owner specifies which subjects should be given access rights and preciously what those rights are.

MAC

- The access control is mandated by system wide security policy.
- The emergence of MAC was in part a stringent needs for information security demanded by military/intelligence community.
- The subjects and objects are assigned security labels.
- Access control decisions are based on the security labels of the subject and requested object.

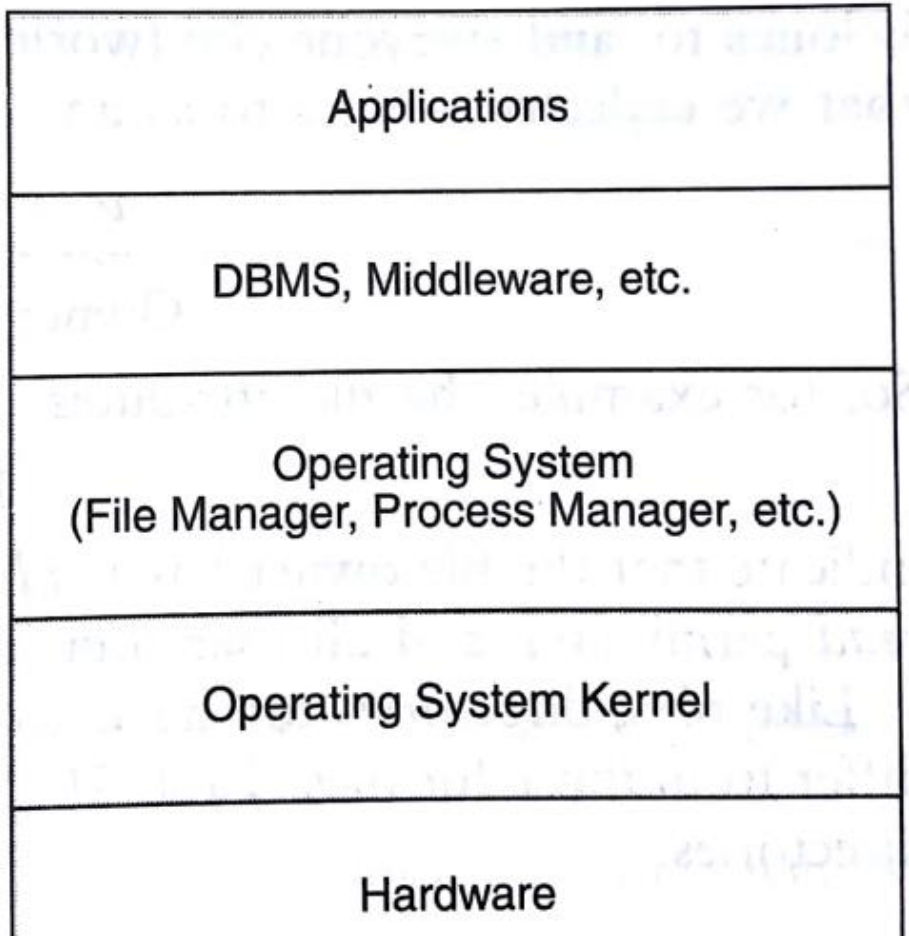
- Early forms of MAC provided multilevel security.
  - Clearance level – each principal was assigned a clearance level.
  - Sensitivity level – each object was classified based on its sensitivity level.

.

RBAC

- Each principal is assigned to one or more roles
- Principal may be in one role at a time.
- Access control is based on the current role of the principal

Layers between application and hardware



- In a computer there are several layers of software between application and hardware
- OS includes modules for file and memory management, process scheduling etc.
- Closest to the hardware is the OS kernel which handles the core OS functionality.
- Access control is one of the core functionality

- Access control is implemented in a critical component called **security kernel.**
- Another component is a **reference monitor**
  - which mediates the requests by subjects to various objects.

Attributes of security kernel

- All access requests to the system resources should pass through the security kernel. It should not be possible to bypass it.
- Being a critical component it is important that it should be easy to analyze and verify the working.
- Should be tamper proof

DAC - UNIX

- Principal – is a user who has account on the system.
- A user may belong to one or more groups.
- Both individual users and groups have 16 bit IDs
- UID – for individual user
- GID –for groups

- For each file the OS maintains an **inode**
- It stores the file size ,data of last modification, ID/UID of the owner.
- Owner is one who created the file.
- Inode also stores file access permissions – read , write ,execute etc.

- Access permissions are given to owner ,group and world
- rwx - 110 owner – has read write permissions on the file.
- rwx - 100 group- has only read permission.
- rwx -000 world – outside world has no permissions.

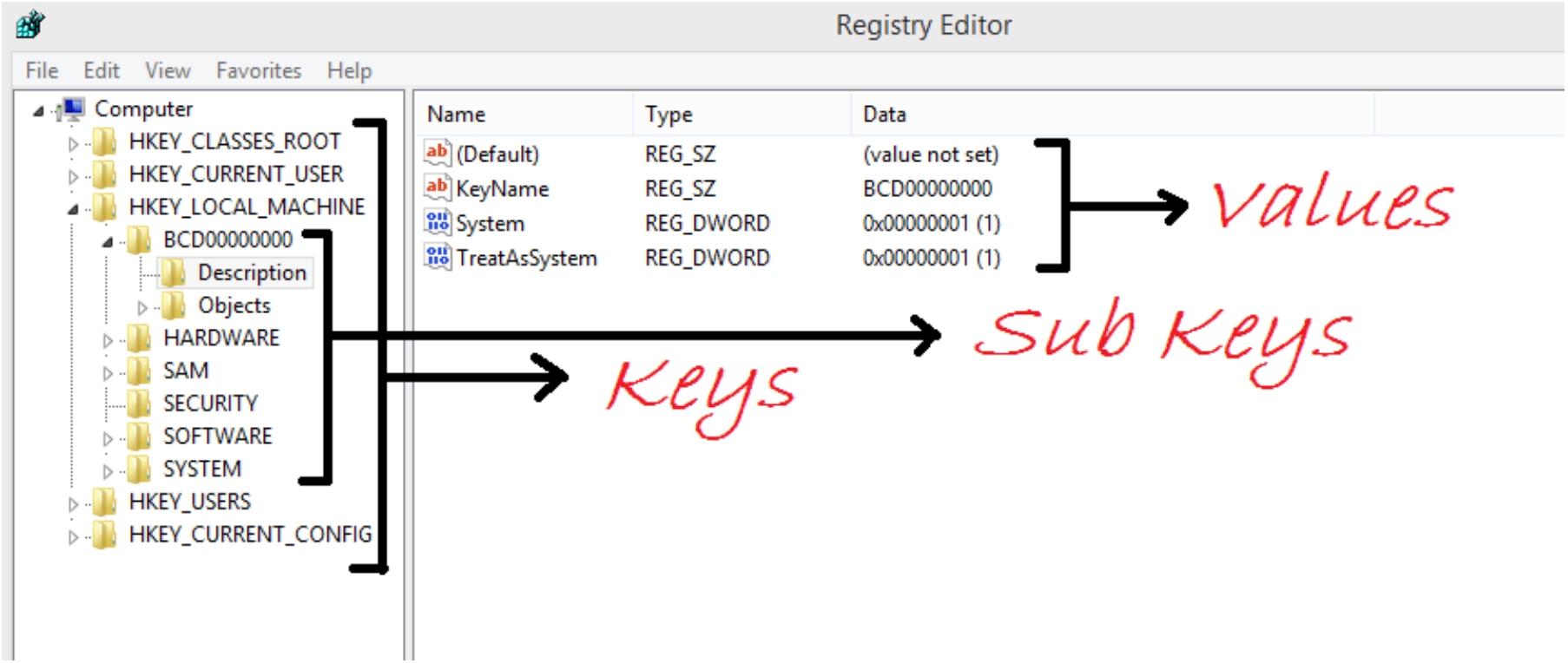
DAC - WINDOWS

- Access control on Unix objects are limited to three operations – read, write and execute.

- Designers of Microsoft windows assigned access control on process , threads , sockets , semaphores ,registry keys etc

Windows Registry

- Stores the configuration data
- Registry is organized as a database of key name – key value pairs


Registry Editor
File
Edit
View
Favorites
Help
Computer
HKEY_CLASSES_ROOT
HKEY_CURRENT_USER
HKEY_LOCAL_MACHINE
BCD00000000
Description
Objects
HARDWARE
SAM
SECURITY
SOFTWARE
SYSTEM
HKEY_USERS
HKEY_CURRENT_CONFIG
Name
Type
Data
(Default)
REG_SZ
(value not set)
KeyName
REG_SZ
BCD00000000
System
REG_DWORD
0x00000001 (1)
TreatAsSystem
REG_DWORD
0x00000001 (1)
values
Sub Keys
Keys

- The 5 Registry Root Keys of The Windows Registry:
  - HKEY_CLASSES_ROOT (HKCR) ...
  - HKEY_CURRENT_USER (HKCU) ...
  - HKEY_LOCAL_MACHINE (HKLM) ...
  - HKEY_USERS (HKU) ...
  - HKEY_CURRENT_CONFIGURATION (HKCC)

Security descriptor

- Each securable object has an associated Security descriptor, which includes :
  - the object owners' ID
  - the discretionary access control list (DACL)
  - the system access control list (SACL)
  - flags related to inheritance

Security Identifier (SID)

- The SID (Security Identifier) is a unique ID number that a computer or domain controller uses to identify you.
- It is a string of alphanumeric characters assigned to each user on a Windows computer, or to each user, group, and computer on a domain-controlled network

- An SID looks like this:
  - S-1-5-32-1045337234-12924708993-5683276719-19000
- Microsoft usually breaks this down into this pattern:
  - (SID)-(revision level)-(identifier-authority)-(subauthority1)-(subauthority2)-(etc)

- Format of security identifier is S-R-I-SA-N

  S – id begins with S

  R- revision number

  I- identifier authority

  SA- sub authority

  N- authority's namespace

- **SID**: The initial S merely identifies the following string as being an SID.
- **Revision level**: To date, this has never changed and has always been 1.
- **Identifier-authority**: This is a 48-bit string that identifies the authority (the computer or network) that created the SID.
- **Subauthority**: This is a variable number that identifies the relation of the user or group described by the SID to the authority that created it. The number tells you:
  - Which computer (or network) created the number
  - Whether this user is a normal user, a guest, an administrator, or part of some other group
  - In what order the user's account was created by this authority (i.e., "This was the first user" or "This is the 231st machine account created".)

- *S - 1 - 1 - 0* (everyone)
- *S - 1 - 5 - 11* (authenticated users)
- *S - 1 - 5 - 18* (local system)
- *S - 1 - 5 - 32 - 544* (built-in group with administrator privileges)

DACL AND ACE

- The most important component in the security descriptor is the DACL.

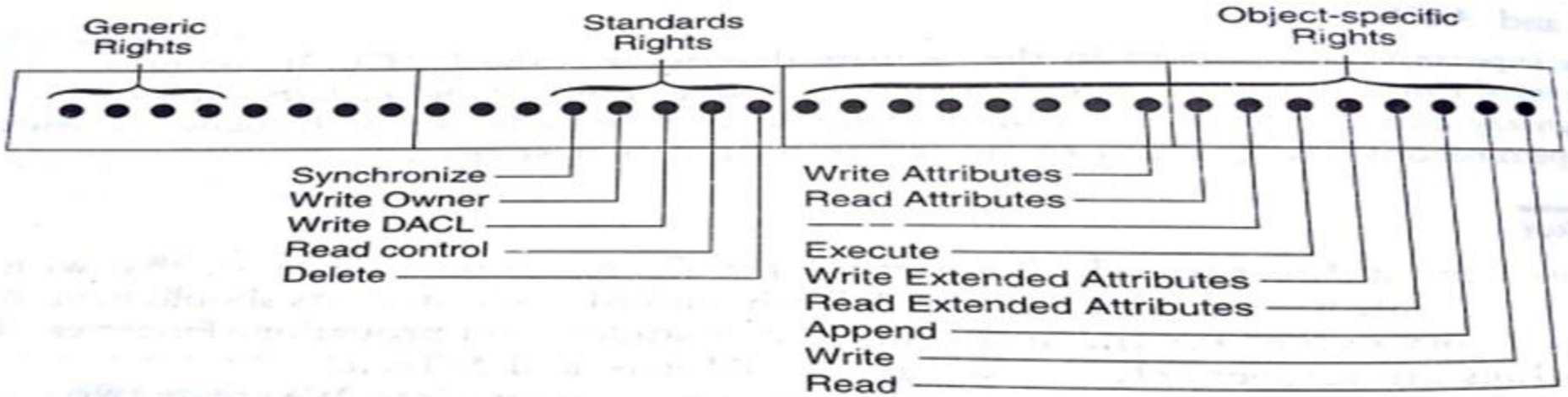
- DACL consists of one or more access control entries(ACEs) .
- Windows support both authorizations.
- These are respectively contained in allow and deny ACEs.
- Basically ACE identifies which specific access rights or permissions are granted or denied to a particular subject.

Access Mask

- Defines all possible actions for a particular type of object (file, folder, and so on) for each access control entry (ACE) in a discretionary access control list (DACL) or a system access control list (SACL). The system chooses the access rights that it can grant to a thread from the possible actions listed in the access mask.

Access Mask

- Windows uses a 32 bit access mask
- Least significant 16 bits are used to represent object specific rights.



Access control mask in windows (object–specific rights are for a file object)

- **Object Specific access rights:**
- These include access rights such as FILE_READ_DATA and FILE_APPEND_DATA, which provide permission to read and write data in a file.
- Objects can have up to 16 different specific access rights, depending on the object type.

Bit # in Access Mask	File Right	DirectoryRight
0	Read	List
1	Write	Add File
2	Append	Add Subdirectory
3	Read EA	Read EA
4	Write EA	Write EA
5	Execute	Traverse
6	—	—
7	Read Attributes	Read Attributes
8	Write Attributes	Write Attributes

File specific and directory specific rights

- **Standard access rights:**

- In addition to object specific type, mask specifies rights to all objects (Standard rights) which includes
  - Delete
  - Read control
  - Write DACL
  - Write owner
  - synchronize
- An object with no DACL is accessible to everyone. Such an object is said to be NULL DACL

- **Generic access rights:**
- These map to specific access rights and standard access rights.
- Each type of securable object maps generic access rights to its own specific and standard access rights.
- Generic access rights for file objects include
  - FILE_GENERIC_READ
  - FILE_GENERIC_WRITE
  - FILE_GENERIC_ EXECUTE.
- These three types are listed in Windows Explorer in Windows 2000 and in Windows NT Explorer in Windows NT as the special permissions read (R), write (W), and execute (X).

ACESS CONTROL ALGORITHM

- A subject makes a request to the OS for access to an object
- The OS makes a Yes /No decision based on three inputs
  - the access token of the subject(requester)
  - The security descriptor of the requested object
  - The permissions requested by the subject on that object

- Both the security descriptor and access token are created by the OS
- Security descriptor is associated with an object , an access token is associated with a subject

Access token

- Access token carries the credentials of a user. It includes -
  - SID of user
  - SIDs of the groups the user belongs to
  - a list of user privileges

USER SID	VIVEK
GROUP SIDs	MTECH2 RAs SysAdmin
PRIVILEGES	SHUTDOWN SYSTEM Generate Security Audits Backup Files and Directories

E.g.

Access token for the user 'VIVEK'

It shows VIVEK is a member of 3 groups and he has some set of privileges (System wide activity)

- When a user logs in , the OS creates an access token for that user
- This token is attached to the initial process.
- It is inherited by the child processes of the initial process.
- In addition to the security descriptor of an object the access token of the subject is used in the access control decision.

Let **P** be the set of desired permissions on an object.
Let **G** be the set of permissions granted on the object.
Initially **G** = Φ, i = 1.
Let a = Number of ACEs in the object's DACL.

```
for ( i = 1 to a ) { // for each ACE in the DACL
if ( SID of ACE_i matches a user/group SID in the Access Token )
   & ( AccessMask(ACE_i ) ∩ ( P - G ) ≠ Φ ) {
   if ACE_i is of type "deny",
            then quit with
            Access Control Decision = "ACCESS DENIED"
   if ACE_i is of type "allow",
            then G = G ∪ ( P ∩ AccessMask(ACE_i ) )
   if ( G = P ),
            quit with
            Access Control Decision = "ACCESS ALLOWED"
   }
}
```

- Access control algorithm makes an all-or-nothing decision.
- Either the request is granted in totality or it is denied.
- Default access decision is ACCESS DENIED

ACE inheritance

- How is a DACL assigned to a newly created object ?
  - The process that creates an object could provide a security descriptor for that including DACL
  - A newly created file inherits the ACEs of its parent directory by default

MANDATORY ACCESS CONTROL

- One of the earliest mechanisms to implement MAC was to assign system-wide security levels to both objects and subjects .
- Access permissions are granted based on the security levels of the subject and object.

Multi level security (MLS)

- All information carrying objects (files , documents) are classified based on their sensitivity levels
- Principals(users) are also assigned a clearance level
- Commonly used sensitivity/clearance levels are
  - TOP SECRET
  - SECRET
  - CONFIDENITAL
  - UNCLASSIFIED

- From the perspective of confidentiality alone
- If the subject has been cleared at level SECRET then he/she should be able to read all documents labelled SECRET or lower.
- However he/she should not be permitted to read a document labelled TOP SECRET

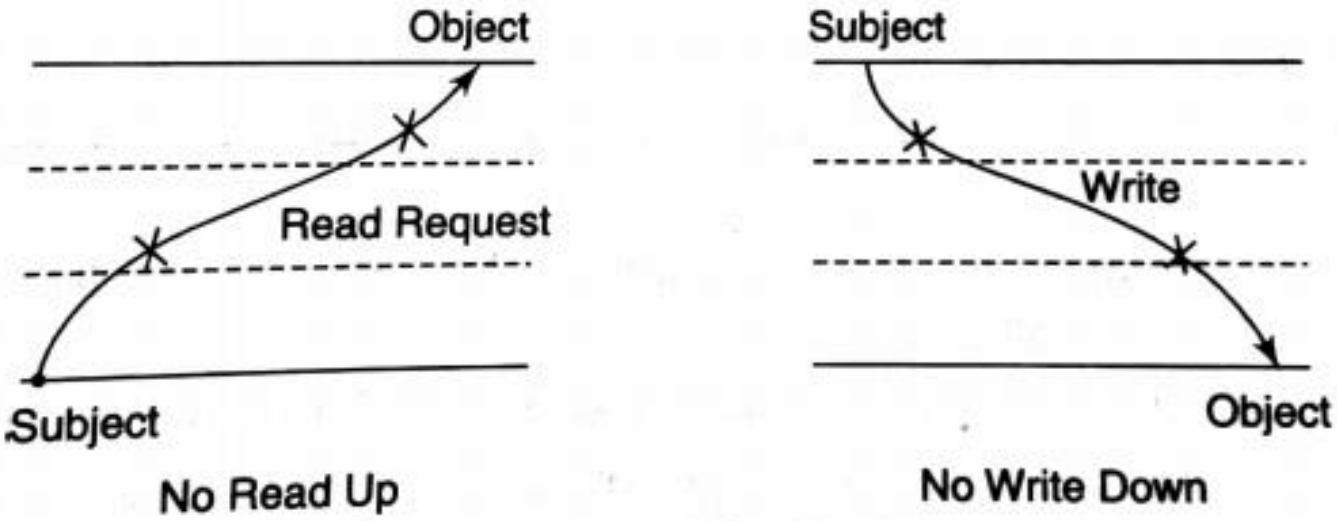
Need-to-know basis

- The term "need to know", when used by government and other organizations describes the restriction of data which is considered very sensitive.
- Even if one has all the necessary official approvals (such as a security clearance) to access certain information, one would not be given access to such information, unless one has a specific *need to know*.
  - i.e. access to the information must be necessary for one to conduct one's official duties
  - Need-to-know also aims to discourage "browsing" of sensitive material by limiting access to the smallest possible number of people.

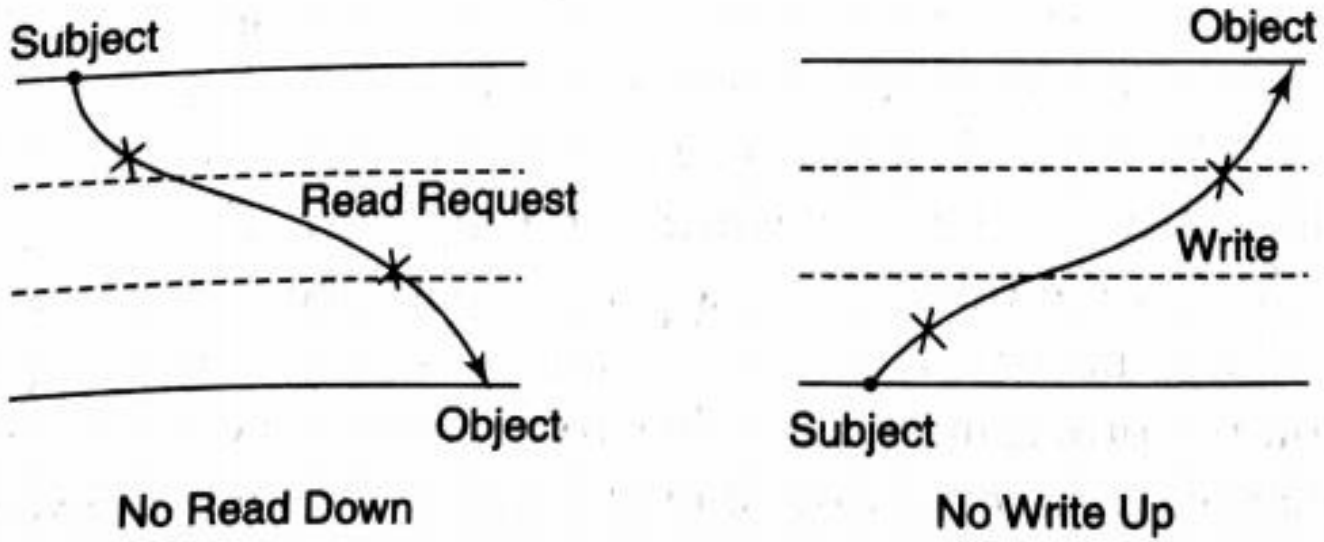
- Every documents in an organization can be placed under different subject category.
  - Like sales, New products & Business partners
- For a user sales manager is concerned ,he need to get access only with first 2 subjects categories.
- A compartment (subset of subject category) can be created and can assign clearance to sales manager.

Security policy

- Security policy is used to prevent the flow of information from an object at a high security level to a subject at lower clearance level.
- Such policy is relevant for applications in national defense/military.
- Confidentiality requirement is captured by BellLaPadula Model (BLP)
- Integrity requirement is captured by Biba model



(a) Confidentiality requirement (Bell-LaPadula)



(b) Integrity requirement (Biba)

RBAC

- Its an access control scheme which is used widely in real world systems.
- Roles determines the functions or tasks
- Primary concern in implementing RBAC is the mapping of roles to permissions (assignment of permissions to roles)

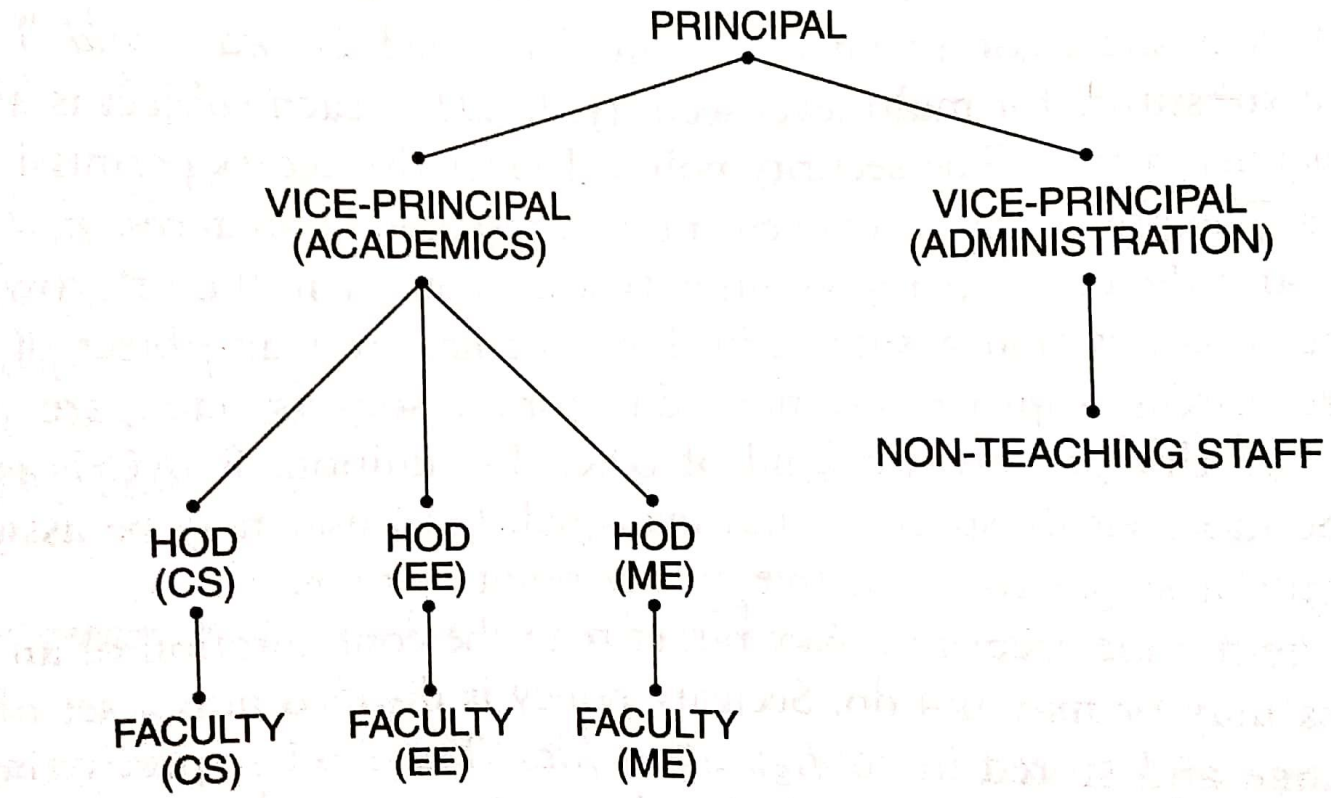
- Another mapping is assignment of peoples to roles.
- Multiple persons may be assigned to same role.

Working

- At login time user is mapped to one or more roles
- But at login session user may access(invokes) different programs .
- Depending on the program invoked user's role may change.

- In RBAC primary concern is the mapping of roles to permissions.
- Assignment of users to roles is secondary and it change more frequently.
- The roles-to-permission mapping is more stable while the mapping of users to roles is more dynamic.
- Roles reflect duties and authority within an organizations so role hierarchies may be identified.

Role heirarchy



Staff roles in a small engineering college

- Each role in the hierarchy has permissions of all of the roles below it.
- Constraints may be imposed on the combination of roles that a person may have.
- For example :
  - A Student cannot be registered for a course for which he is also the Teaching Assistant(TA) offering of that course.
  - Software developer and tester for a given module in a project should not be the same individual.

  By specifying and enforcing constraints of this nature ,RBAC realize the principle of 'separation of duty'

Security-Enhanced Linux

Background

Security-Enhanced Linux is a NSA (National Security Agency) backed research project.

Goals:

- Promote Security Research
- Address Operating System Security
- Demonstrate MAC (Mandatory Access Controls) through Type Enforcement® technology in a mainstream operating system

Note: SELinux project is **not** intended as a complete security solution for Linux

Background

In Conjunction with Secure Computing Corporation (SCC)

Previous projects:

LOCK system – Secure Ada project through Honeywell

DTMach – Mach-based prototype

DTOS (Distributed Trusted Operating System)

Fluke* - University of Utah's research operating system

Flask architecture*

*** - Fluke was a pre-existing operating system used by the Flux Research group at Utah. During the integration/transfer of technology into the system, enhanced dynamic security policies were produced. The resulting architecture is named Flask.**

Background

Why choose Linux?

As hinted in the goals, Linux is an open source project with many developers; therefore:

- Provides an opportunity for more research.
- Allows application/testing in a mainstream operating system.
- Improves security in an existing operating system.

SELinux

Security-Enhanced Linux (SELinux)

- Uses the Linux Security Modules (LSM) framework to implement flexible Mandatory Access Control (MAC) in the Linux kernal.
- Restricts privileges of user programs and system servers using security labels and an administratively-defined policy.

SELinux

MAC versus DAC

Discretionary Access Control (DAC) is the standard security model for Linux. In this model, access privileges are based on the user identity and object ownership.

Mandatory Access Control (MAC) limits privileges for subjects (processes) and objects (file, socket, device, etc).

SELinux

Security Policies are implemented using:

- **Type Enforcement® (TE)**
- **Role-based access control (RBAC)**

Type Enforcement

(introduced in 1985 by Boebert and Kain)

- Traditional TE model uses a domain attribute for each process and a type attribute for each object. User operation is limited to certain domains.
- In SELinux, a single attribute is used for both subject and object (“A domain is simply a type that can be associated with a process“).
- Security classes can distinguish objects of the same type. Uses the RBAC model instead of associating users with domains.
- An access matrix defines the privileges of each type for a given domain.

Role-Based Access Control

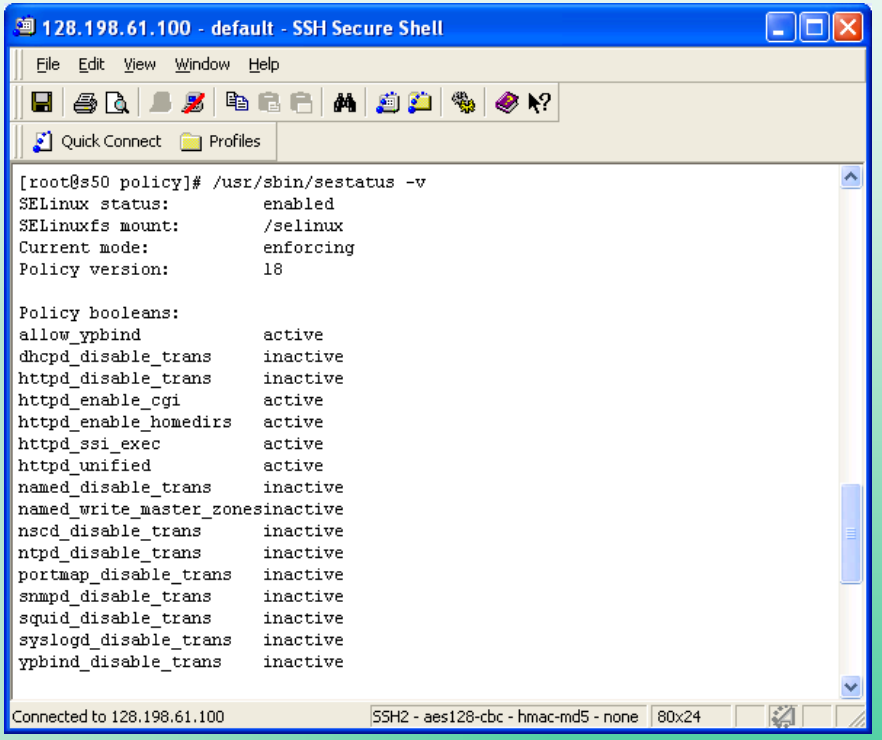
- Each user gets a set of roles
- Each role is assigned a set of TE domains.

Note: users are not identified by Linux uids; instead a user identity attribute is used in the security context.

Policies

Configuration consists of:

- Flask definitions
- TE and RBAC declarations and rules
- User declarations
- Constraint definitions
- Security context specifications.



Policies

TE Statements

- Attribute Declarations
- Type Declarations
- TE Transition Rules
- TE Change Rules
- TE Access Vector Rules
- TE Access Vector Assertions
- Type Member Rules

RBAC Statements

- Role Declarations and Dominance
- Role Allow Rules
- Role Transition Rules

Policies

```
te_rbac -> te_rbac_statement | te_rbac te_rbac_statement
te_rbac_statement -> te_statement | rbac_statement
te_statement -> attrib_decl |
                type_decl |
                type_transition_rule |
                type_change_rule |
                te_av_rule |
                te_assertion
rbac_statement -> role_decl |
                  role_dominance |
                  role_allow_rule
```

Syntax for TE and RBAC declarations*

```
type_decl -> TYPE identifier opt_alias_def opt_attr_list ';'
opt_alias_def -> ALIAS aliases | empty
aliases -> identifier | '{' identifier_list '}'
identifier_list -> identifier | identifier_list identifier
opt_attr_list -> ',' attr_list | empty
attr_list -> identifier | attr_list ',' identifier
```

Syntax for type declarations*

Policies

File: domains/program/**apache.te** (patch) Size: 4823 Mime type: text/plain

```
--- tbleher@gmx.de--selinux/policy--suse--0--patch-11/domains/program/apache.te
+++ tbleher@gmx.de--selinux/policy--suse--0--patch-12/domains/program/apache.te
@@ -21,6 +21,8 @@
 ########################################################################
 type http_port_t, port_type, reserved_port_type;

+bool httpd_unified false;
+
 # Allow httpd cgi support
 bool httpd_enable_cgi false;

@@ -30,6 +32,9 @@
 # Run SSI execs in system CGI script domain.
 bool httpd_ssi_exec false;

+# Allow http daemon to communicate with the TTY
+bool httpd_tty_comm false;
+
 ########################################################################
 # Apache types
 ########################################################################
@@ -40,7 +45,7 @@

 append_logdir_domain(httpd)
 #can read /etc/httpd/logs
-allow httpd_t httpd_log_t:lnk_file { read };
+allow httpd_t httpd_log_t:lnk_file read;

 # For /etc/init.d/apache2 reload
 can_tcp_connect(httpd_t, httpd_t)
@@ -130,7 +135,8 @@

 # execute perl
 allow httpd_t { bin_t sbin_t }:dir r_dir_perms;
-can_exec(httpd_t, bin_t)
```

Bleher, Thomas November 30, 2004

Limitations

Performance Overhead – calculated at approximately 7%.

Complexity – requires in-depth knowledge of the operating system, the security policies, and the operating environment.

Maintenance – policy fine-tuning, changes required for changes to the system.