Multiple Independent Levels of Security

What

high-assurance security architecture based on the concepts of separation and controlled information flow

implemented by separation mechanisms that support both untrusted and trustworthy components

ensures security cannot be bypassed by alternate comm path
ensures a system is tamperproof
    Unauthorized changes to configuration & data is prevented
ensures a system can be *evaluated*
    Requires: modular components, well specified, compact, simple components, formally provable properties
is always invoked
    every access and message is checked by an appropriate security monitor

employs one or more separation mechanisms
  separation kernel
  partitioning communication system
  physical separation

Multiple Independent Levels of Security

What

supports enforcement of one or more application/system specific security policies by authorizing information flow only between components in the same security domain or through trustworthy security monitors

MILS architecture allows for execution of multiple applications at potentially multiple security levels or classifications

Each is protected from others and each may communicate with the others based on mechanisms that support policy enforcement

The old way to get separation was to have physically separate computers, networks, and displays – not practical

Multiple Independent Levels of Security

Importance

Military needs systems that are very highly secure

MILS architectures can be evaluated according to the Common Criteria

The US military requires evaluation to high security standards

COTS components that have a very high evaluation are desirable as they can save plenty of money in design and certification costs

Major application: military jets
Imagine: a squadron of planes is suddenly disabled in the air due to enemy intrusion
F-35 Joint Strike Fighter Communications, Navigation, Identification (CNI) system uses a MILS architecture

Major application: control of nuclear power generation

Major application: control of sewage treatment systems

Multiple Independent Levels of Security

Separation Kernels

Purpose: provide multi-level security on general purpose multi-user systems

Creates an environment which is indistinguishable from that of a of a distributed physical system

It must appear as if each regime is a separate, isolated machine and that information can only flow from one machine to another along known external communication lines

It must be proved that there are no channels for information flow between regimes other than those explicitly provided

- Data isolation ensures a partition can't access resources in other partitions
- Periods processing ensures applications within partitions execute for the specified duration in the system schedule
- Information flow defines permitted info flows between partitions
- Fault isolation ensures a failure in one partition does not impact any other partition within the system

Multiple Independent Levels of Security

Separation Kernels

Separation Kernel Protection Profile: provides a formal notion

Protection of all resources from unauthorized access

Separation of internal resources used by (target of evaluation) functions from exported resources made available to subjects

Isolation and partitioning of exported resources

Mediation of information flows between partitions and between exported resources

Auditing

Multiple Independent Levels of Security

Separation Kernels

Available from

Green Hills Software

Integrity 178B RTOS used in F-16, F-22, F-35, Airbus 380
Very tiny kernel – 4K lines
Kernel is evaluated to NSA EAL 6+ (semi-formally verified)

LynuxWorks

LynxSecure separation kernel and embedded hypervisor
LynxOS-178 RTOS

SYSGO

PikeOS – small set of privileged services
Used in product certified by the French NIS Agency

Wind River Systems

VxWorks MILS platform compliant with Separation Kernel Protection Profile (SKPP) from the NSA

OK Labs

OKL4 microkernel – in billions of mobile devices

Multiple Independent Levels of Security

Partitioning Communications System (PCS)

A communications security architecture compliant with an information flow separation policy

Extends the MILS architecture to network flows

Works with a separation kernel to ensure
- System security channels cannot be bypassed
- System can be evaluated
- System is tamperproof

Supports (a kind of) formal proof of correctness

Multiple Independent Levels of Security

Formal Proof of Correctness

Introduce and define States of a system in terms of security

Define transition rules from State to State based on various kinds of triggers (e.g. input or clock timer firing)

Check that the initial State is considered secure

For each transition from State A to State B, check that if A is considered secure then B can be considered secure

Then we have a proof that the system is secure

Multiple Independent Levels of Security

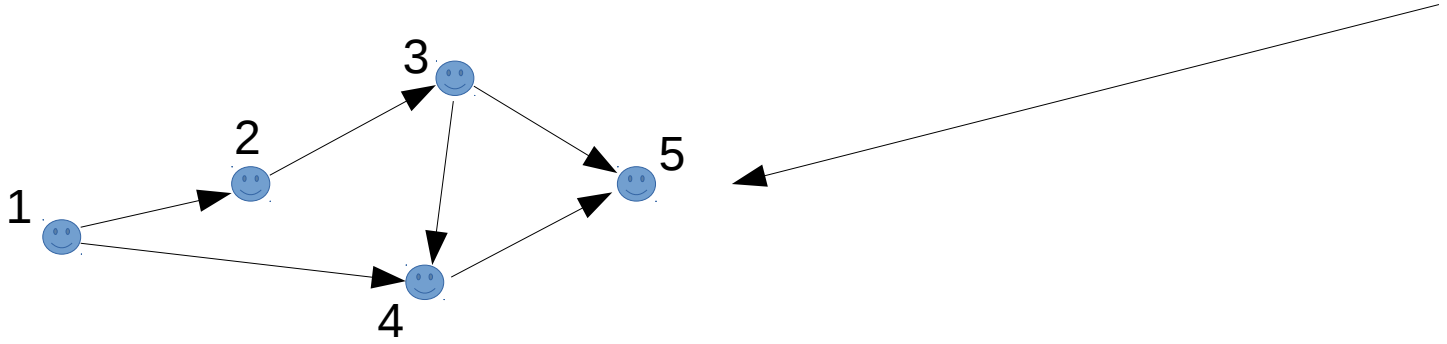
Formal Proof of Correctness

Operation:

Triple: (subject, object, operation)

Example: (franco, sshd, execute)

Subjects and Objects labeled with security levels in partial order



But each subject has a current security level and a maximum security level

Thus subjects can be 'downgraded' in security temporarily

Access control matrix (M): gives permissions for a given operation (o) on particular sets of security levels (l)

A **State:** (o,**M**,l)

Multiple Independent Levels of Security

Formal Proof of Correctness

Policy types (discretionary and mandatory):

Discretionary: access may be permitted (i.e. (s,o,op))
No read-up: subject may not read object at higher security level
No write-down: subject can't write to object at lower security level

Subjects are processes, memory is an object
Subjects have access to memory
Subjects can act as channels by reading one memory object and writing that information to another memory object

Trusted subjects are exempt from no write-down policy
Subjects can be 'downgraded' in security temporarily to losen the mandatory restrictions

A State is secure if all current access triples (s,o,op) are Permitted by the policies above
A State transition is secure if it is between two secure States

If the initial State is secure and all transitions are secure then the system is secure

Multiple Independent Levels of Security

Formal Proof of Correctness

Operations for a real OS:

Execute:

Read:

Write:

Read and write:

Get-read: requests read access to an object

Release-read: release an object

Give-read: grant read access to another process

Rescind-read: withdraw read permission given to another process

Create-object: OS has to check write access on the object directory is permitted and the security level of the object dominates the security level of the process

Change-subject-current-security-level:

Change-object-current-security-level: